

Charu C. Aggarwal

Probability and Statistics for Machine Learning

A Textbook



Probability and Statistics for Machine Learning

Charu C. Aggarwal

Probability and Statistics for Machine Learning

A Textbook



Charu C. Aggarwal
IBM T. J. Watson Research Center
Yorktown, NY, USA

Supplementary Information A Solution Manual to this book can be downloaded from <https://link.springer.com/book/978-3-031-53281-8>.

ISBN 978-3-031-53281-8 ISBN 978-3-031-53282-5 (eBook)
<https://doi.org/10.1007/978-3-031-53282-5>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Paper in this product is recyclable.

To my wife Lata, my daughter Sayani,
and all my mathematics teachers

Contents

1	Probability and Statistics: An Introduction	1
1.1	Introduction	1
1.1.1	The Interplay Between Probability, Statistics, and Machine Learning	2
1.1.2	Chapter Organization	2
1.2	Representing Data	3
1.2.1	Numeric Multidimensional Data	4
1.2.2	Categorical and Mixed Attribute Data	4
1.3	Summarizing and Visualizing Data	6
1.4	The Basics of Probability and Probability Distributions	8
1.4.1	Populations versus Samples	12
1.4.2	Modeling Populations with Samples	13
1.4.3	Handling Dependence in Data Samples	15
1.5	Hypothesis Testing	15
1.6	Basic Problems in Machine Learning	16
1.6.1	Clustering	16
1.6.2	Classification and Regression Modeling	17
1.6.3	Outlier Detection	20
1.7	Summary	21
1.8	Further Reading	22
1.9	Exercises	22
2	Summarizing and Visualizing Data	25
2.1	Introduction	25
2.1.1	Chapter Organization	26
2.2	Summarizing Data	26
2.2.1	Univariate Summarization	26
2.2.2	Multivariate Summarization	34
2.3	Data Visualization	46
2.3.1	Univariate Visualization	46
2.3.2	Multivariate Visualization	52
2.4	Applications to Data Preprocessing	57
2.4.1	Univariate Preprocessing Methods	57
2.4.2	Whitening: A Multivariate Preprocessing Method	59

2.5	Summary	61
2.6	Further Reading	62
2.7	Exercises	62
3	Probability Basics and Random Variables	65
3.1	Introduction	65
3.1.1	Chapter Organization	66
3.2	Sample Spaces and Events	66
3.3	The Counting Approach to Probabilities	73
3.4	Set-Wise View of Events	74
3.5	Conditional Probabilities and Independence	77
3.6	The Bayes Rule	80
3.6.1	The Observability Perspective: Posteriors versus Likelihoods	82
3.7	The Basics of Probability Distributions	83
3.7.1	Closed-Form View of Probability Distributions	84
3.7.2	Continuous Distributions	86
3.7.3	Multivariate Probability Distributions	89
3.8	Distribution Independence and Conditionals	92
3.8.1	Independence of Distributions	92
3.8.2	Conditional Distributions	93
3.8.3	Example: A Simple 1-Dimensional Knowledge-Based Bayes Classifier	95
3.9	Summarizing Distributions	97
3.9.1	Expectation and Variance	97
3.9.2	Distribution Covariance	104
3.9.3	Useful Multivariate Properties Under Independence	106
3.10	Compound Distributions	108
3.10.1	Total Probability Rule in Continuous Hypothesis Spaces	109
3.10.2	Bayes Rule in Continuous Hypothesis Spaces	111
3.11	Functions of Random Variables (*)	114
3.11.1	Distribution of the Function of a Single Random Variable	114
3.11.2	Distribution of the Sum of Random Variables	117
3.11.3	Geometric Derivation of Distributions of Functions	119
3.12	Summary	121
3.13	Further Reading	122
3.14	Exercises	122
4	Probability Distributions	127
4.1	Introduction	127
4.1.1	Chapter Organization	128
4.2	The Uniform Distribution	128
4.3	The Bernoulli Distribution	131
4.4	The Categorical Distribution	132
4.5	The Geometric Distribution	135
4.6	The Binomial Distribution	138
4.7	The Multinomial Distribution	141
4.8	The Exponential Distribution	145
4.9	The Poisson Distribution	149
4.10	The Normal Distribution	152
4.10.1	Multivariate Normal Distribution: Independent Attributes	159

4.10.2	Multivariate Normal Distribution: Dependent Attributes	161
4.11	The Student's <i>t</i> -Distribution	165
4.12	The χ^2 -Distribution	170
4.12.1	Application: Mahalanobis Method for Outlier Detection	174
4.13	Mixture Distributions: The Realistic View	175
4.13.1	Why Mixtures are Ubiquitous: A Motivating Example	176
4.13.2	The Basic Generative Process of a Mixture Model	176
4.13.3	Some Useful Results for Prediction	178
4.13.4	The Conditional Independence Assumption	179
4.14	Moments of Random Variables (*)	180
4.14.1	Central and Standardized Moments	180
4.14.2	Moment Generating Functions	182
4.15	Summary	186
4.16	Further Reading	186
4.17	Exercises	186
5	Hypothesis Testing and Confidence Intervals	191
5.1	Introduction	191
5.1.1	Chapter Organization	193
5.2	The Central Limit Theorem	193
5.3	Sampling Distribution and Standard Error	194
5.4	The Basics of Hypothesis Testing	196
5.4.1	Confidence Intervals	200
5.4.2	When Population Standard Deviations Are Not Available	203
5.4.3	The One-Tailed Hypothesis Test	205
5.5	Hypothesis Tests For Differences in Means	208
5.5.1	Unequal Variance <i>t</i> -Test	208
5.5.2	Equal Variance <i>t</i> -Test	213
5.5.3	Paired <i>t</i> -Test	214
5.6	χ^2 -Hypothesis Tests	217
5.6.1	Standard Deviation Hypothesis Test	218
5.6.2	χ^2 -Goodness-of-Fit Test	220
5.6.3	Independence Tests	222
5.7	Analysis of Variance (ANOVA)	224
5.8	Machine Learning Applications of Hypothesis Testing	230
5.8.1	Evaluating the Performance of a Single Classifier	230
5.8.2	Comparing Two Classifiers	231
5.8.3	χ^2 -Statistic for Feature Selection in Text	232
5.8.4	Fisher Discriminant Index for Feature Selection	233
5.8.5	Fisher Discriminant Index for Classification (*)	235
5.9	Summary	239
5.10	Further Reading	240
5.11	Exercises	240
6	Reconstructing Probability Distributions from Data	245
6.1	Introduction	245
6.1.1	Chapter Organization	247
6.2	Maximum Likelihood Estimation	247
6.2.1	Comparing Likelihoods with Posteriors	252

6.3	Reconstructing Common Distributions from Data	252
6.3.1	The Uniform Distribution	252
6.3.2	The Bernoulli Distribution	253
6.3.3	The Geometric Distribution	255
6.3.4	The Binomial Distribution	256
6.3.5	The Multinomial Distribution	257
6.3.6	The Exponential Distribution	258
6.3.7	The Poisson Distribution	259
6.3.8	The Normal Distribution	260
6.3.9	Multivariate Distributions with Dimension Independence	262
6.3.10	Gaussian Distribution with Dimension Dependence	263
6.4	Mixture of Distributions: The EM Algorithm	265
6.5	Kernel Density Estimation	272
6.6	Reducing Reconstruction Variance	274
6.6.1	Variance in Maximum Likelihood Estimation	276
6.6.2	Prior Beliefs with Maximum A Posteriori (MAP) Estimation	278
6.6.3	Kernel Density Estimation: Role of Bandwidth	283
6.7	The Bias-Variance Trade-Off	285
6.8	Popular Distributions Used as Conjugate Priors (*)	289
6.8.1	Gamma Distribution	290
6.8.2	Beta Distribution	292
6.8.3	Dirichlet Distribution	294
6.9	Summary	298
6.10	Further Reading	298
6.11	Exercises	298
7	Regression	303
7.1	Introduction	303
7.1.1	Chapter Organization	304
7.2	The Basics of Regression	304
7.2.1	Interpreting the Coefficients	305
7.2.2	Feature Engineering Trick for Dropping Bias	305
7.2.3	Regression: A Central Problem in Statistics and Linear Algebra	307
7.3	Two Perspectives on Linear Regression	308
7.3.1	The Linear Algebra Perspective	308
7.3.2	The Probabilistic Perspective	310
7.4	Solutions to Linear Regression	314
7.4.1	Closed-Form Solution to Squared-Loss Regression	314
7.4.2	The Case of One Non-Trivial Predictor Variable	318
7.4.3	Solution with Gradient Descent for Squared Loss	321
7.4.4	Gradient Descent For L_1 -Loss Regression	324
7.5	Handling Categorical Predictors	324
7.6	Overfitting and Regularization	326
7.6.1	Closed-Form Solution for Regularized Formulation	329
7.6.2	Solution Based on Gradient Descent	330
7.6.3	LASSO Regularization	331
7.7	A Probabilistic View of Regularization	331
7.8	Evaluating Linear Regression	334
7.8.1	Evaluating In-Sample Properties of Regression	334

7.8.2	Out-of-Sample Evaluation	338
7.9	Nonlinear Regression	339
7.9.1	Interpretable Feature Engineering	341
7.9.2	Explicit Feature Engineering with Similarity Matrices	343
7.9.3	Implicit Feature Engineering with Similarity Matrices	346
7.10	Summary	349
7.11	Further Reading	349
7.12	Exercises	349
8	Classification: A Probabilistic View	353
8.1	Introduction	353
8.1.1	Chapter Organization	354
8.2	Generative Probabilistic Models	354
8.2.1	Continuous Numeric Data: The Gaussian Distribution	357
8.2.2	Binary Data: The Bernoulli Distribution	362
8.2.3	Sparse Numeric Data: The Multinomial Distribution	365
8.2.4	Plate Diagrams for Generative Processes	368
8.3	Loss-Based Formulations: A Probabilistic View	370
8.3.1	Least-Squares Classification	372
8.3.2	Logistic Regression	377
8.3.3	Multinomial Logistic Regression	382
8.4	Beyond Classification: Ordered Logit Model	385
8.4.1	Maximum Likelihood Estimation for Ordered Logit	386
8.5	Summary	388
8.6	Further Reading	388
8.7	Exercises	388
9	Unsupervised Learning: A Probabilistic View	393
9.1	Introduction	393
9.1.1	Chapter Organization	394
9.2	Mixture Models for Clustering	394
9.2.1	Continuous Numeric Data: The Gaussian Distribution	398
9.2.2	Binary Data: The Bernoulli Distribution	403
9.2.3	Sparse Numeric Data: The Multinomial Distribution	404
9.3	Matrix Factorization	407
9.3.1	The Squared Loss Model	408
9.3.2	Probabilistic Latent Semantic Analysis	413
9.3.3	Logistic Matrix Factorization	419
9.3.4	Gradient Descent Steps for Logistic Matrix Factorization	421
9.4	Outlier Detection	421
9.4.1	The Mahalanobis Method: A Probabilistic View of Whitening	422
9.4.2	Mixture Models in Outlier Detection	426
9.4.3	Matrix Factorization for Outlier Detection	427
9.5	Summary	430
9.6	Further Reading	430
9.7	Exercises	430
10	Discrete State Markov Processes	435
10.1	Introduction	435

10.1.1 Chapter Organization	436
10.2 Markov Chains	437
10.2.1 Steady-State Behavior of Markov Chains	439
10.2.2 Transient Behavior of Markov Chains	442
10.2.3 Periodic Markov Chains	446
10.2.4 Ergodicity	447
10.2.5 Different Cases of Ergodicity and Non-Ergodicity	450
10.2.6 Properties and Applications of Non-Ergodic Markov Chains	451
10.2.7 Probabilities of Absorbing Outcomes	458
10.2.8 The View from Matrix Algebra (*)	460
10.3 Machine Learning Applications of Markov Chains	463
10.3.1 PageRank	463
10.3.2 Application to Vertex Classification	466
10.4 Markov Chains to Generative Models	470
10.5 Hidden Markov Models	471
10.5.1 Formal Definition and Techniques for HMMs	474
10.5.2 Evaluation: Computing the Fit Probability for Observed Sequence	475
10.5.3 Explanation: Determining the Most Likely State Sequence for Observed Sequence	476
10.5.4 Training: Baum-Welch Algorithm	477
10.6 Applications of Hidden Markov Models	479
10.6.1 Mixture of HMMs for Clustering	479
10.6.2 Outlier Detection	480
10.6.3 Classification	481
10.7 Summary	481
10.8 Further Reading	482
10.9 Exercises	482
11 Probabilistic Inequalities and Approximations	485
11.1 Introduction	485
11.1.1 Chapter Organization	486
11.2 Jensen's Inequality	486
11.3 Markov and Chebychev Inequalities	490
11.4 Approximations for Sums of Random Variables	494
11.4.1 The Chernoff Bound	496
11.4.2 The Normal Approximation to the Binomial Distribution	500
11.4.3 The Poisson Approximation to the Binomial Distribution	502
11.4.4 The Hoeffding Inequality	504
11.5 Tail Inequalities Versus Approximation Estimates	507
11.6 Summary	510
11.7 Further Reading	511
11.8 Exercises	511
References	515
Index	519

Preface

“Lies, damned lies, and statistics.” — Mark Twain

Most of machine learning is directly or indirectly related to probability and statistics. After all, machine learning is all about making predictions based on data, which inevitably leads to statistical methods. These statistical methods are often couched as *models*, which use *probabilities* to quantify the likelihoods of events. Therefore, having a strong background in probability and statistics is critical.

The familiarity required with probability and statistics often goes well beyond what is taught in undergraduate curricula. As a result, this presents a challenge to beginners in the field. In many cases, the type of techniques required from probability and statistics are specific to machine learning, which is not covered by generic courses on probability and statistics. This book therefore develops a treatment of probability and statistics from the specific perspective of machine learning.

This book teaches probability and statistics with a specific focus on machine learning applications. As a natural consequence of this approach, many key concepts in machine learning are covered in detail. Therefore, it is possible to learn a significant amount of machine learning while learning probability and statistics from this book. The chapters are organized as follows:

1. *The basics of probability and statistics:* These chapters focus on the basics of probability and statistics, and cover the key principles of these topics. Chapter 1 provides an overview of the area of probability and statistics as well as its relationship to machine learning. The fundamentals of probability and statistics are covered in Chapters 2 through 5.
2. *From probability to machine learning:* Many machine learning applications are addressed using probabilistic models, whose parameters are then learned in a data-driven manner. Chapters 6 through 9 explore how different models from probability and statistics are applied to machine learning. Perhaps the most important tool that bridges the gap from data to probability is maximum-likelihood estimation, which is explored extensively in these chapters.
3. *Advanced topics:* Chapter 10 is devoted to discrete-state Markov processes. It explores the application of probability and statistics to a temporal and sequential setting,

although the applications extended to more complex settings such as graphical data. Chapter 11 covers a number of probabilistic inequalities and approximations.

About 215 worked examples are provided in the book in order to elucidate different concepts. Furthermore, the book contains about 150 unsolved exercises within chapters and 285 unsolved exercises at the end of chapters. The worked examples should be solved first without looking at the solutions. This will lead to slower progress through chapters but a better understanding. In-chapter exercises are often similar to worked examples — hints for solving the more difficult of these exercises are often given immediately after the exercise in order to help the reader along. Exercises at the end of the chapter are intended to be solved as refreshers after completing the chapter. An instructor’s solution manual is available containing solutions to end-of-chapter exercises. There are a total of about 650 (solved, in-chapter, and end-of-chapter) exercises in the book. Therefore, the book provides ample opportunity for practice.

Prerequisites for the Book

Probability and statistics are grounded in a solid understanding of basic mathematics. Therefore, a strong *high-school-level* background in calculus is assumed (at the AP Calculus BC level in the US) but an advanced college background in mathematics is not assumed. A basic understanding of vectors and matrices is needed (which are often taught in high-school pre-calculus and physics classes). The only advanced concept from linear algebra that is used repeatedly in the book is that of principal component analysis, which is described from first principles in Chapter 2. Except for a single (clearly demarcated) section in Chapter 10, advanced concepts in linear algebra are not needed for understanding the material. The book makes an effort to point out advanced sections that can be skipped over without loss in continuity. The corresponding sections have been marked by an asterisk (*) in the section header. These sections are typically used only occasionally in machine learning.

Notations

Throughout this book, a vector or a multidimensional data point is annotated with a vector right arrow, such as \vec{X} or \vec{y} . A vector or multidimensional point may be denoted by either small letters or capital letters, as long as it has a bar. Vector dot products are denoted by centered dots, such as $\vec{x} \cdot \vec{y}$. A matrix is denoted in capital letters without a vector symbol, such as R . Random variables are also denoted by capital letters, and the difference between a random variable and a matrix is usually obvious from the underlying context. Samples of random variables are denoted by small letters. Throughout the book, the $n \times d$ matrix corresponding to the entire training data set is denoted by D , with n data points and d dimensions. The individual data points in D are therefore d -dimensional row vectors, and are often denoted by $\vec{x}_1 \dots \vec{x}_n$. Note that these vectors use small letters rather than capital letters, because they are assumed to be samples from some underlying data distribution. Vectors with one component for each data point (observation) are usually n -dimensional column vectors. An example is the n -dimensional column vector \vec{y} of class variables of n data points. An observed value y_i is distinguished from a predicted value \hat{y}_i by a circumflex at the top of the variable.

Acknowledgments

I would like to thank my family for their love and support during the busy time spent in writing this book. I would particularly like to thank my daughter for the encouragement that she invariably provides me during the process of book writing. It is easy to get fatigued with writing books, and she never lets that happen.

I learned much of my probability and statistics during high school and college, and therefore I am indebted to numerous teachers at various educational institutions who introduced me to these concepts. I would like to thank Mr. P. C. Pathrose, Mr. S. Adhikari, Dr. S. Gupta, Dr. S. Sadagopan, Dr. A. Drake, and Dr. A. Barnett as some of the teachers from whom I have learned a lot over the years. I would particularly like to thank the late Professor Alvin Drake, whose legendary teaching style in his probability class at MIT motivated me to prioritize intuition over algebra in probabilistic thinking. This approach helped me greatly in internalizing many subtle points, which I have tried to emphasize in this book. Over the years, I have come to the realization that the secret to appreciating probability and statistics is to enjoy its subtly deceptive nature. The choice of Mark Twain's quote at the beginning of the preface reflects this perspective, and it was not meant to disparage the field in any way — an interesting anecdote is that a variant of this quote was written on the blackboard during my first statistics class in high school.

I also learned a lot from my collaborators in machine learning over the years, which exposed me to several applied aspects of probability and statistics. In particular, I would like to thank Tarek F. Abdelzaher, Jinghui Chen, Jing Gao, Quanquan Gu, Manish Gupta, Jiawei Han, Alexander Hinneburg, Thomas Huang, Nan Li, Huan Liu, Ruoming Jin, Daniel Keim, Arijit Khan, Latifur Khan, Mohammad M. Masud, Jian Pei, Magda Procopiuc, Guojun Qi, Chandan Reddy, Saket Sathe, Jaideep Srivastava, Karthik Subbian, Yizhou Sun, Jiliang Tang, Min-Hsuan Tsai, Haixun Wang, Jianyong Wang, Min Wang, Suhang Wang, Wei Wang, Joel Wolf, Xifeng Yan, Wenchao Yu, Mohammed Zaki, ChengXiang Zhai, and Peixiang Zhao. Last but not the least, I would like to thank Horst Samulowitz for being a supportive manager at IBM.

Author Biography

Charu C. Aggarwal is a Distinguished Research Staff Member (DRSM) at the IBM T. J. Watson Research Center in Yorktown Heights, New York. He completed his undergraduate degree in Computer Science from the Indian Institute of Technology at Kanpur in 1993 and his Ph.D. from the Massachusetts Institute of Technology in 1996.



He has worked extensively in the field of data mining. He has published more than 400 papers in refereed conferences and journals and authored over 80 patents. He is the author or editor of 20 books, including textbooks on data mining, recommender systems, linear algebra, neural networks, and outlier analysis. Because of the commercial value of his patents, he has thrice been designated a Master Inventor at IBM. He is a recipient of an IBM Corporate Award (2003) for his work on bio-terrorist threat detection in data streams, a recipient of the IBM Outstanding Innovation Award (2008) for his scientific contributions to privacy technology, and a recipient of two IBM Outstanding Technical Achievement Awards (2009, 2015) for his work on data streams/high-dimensional data. He received the EDBT 2014 Test of Time Award for his work on condensation-based privacy-preserving data mining.

He is also a recipient of the IEEE ICDM Research Contributions Award (2015) and the ACM SIGKDD Innovation Award (2019), which are the two highest awards for influential research contributions in data mining. He received the IIT Kanpur Distinguished Alumnus Award in 2023.

He has served as the general co-chair of the IEEE Big Data Conference (2014) and as the program co-chair of the ACM CIKM Conference (2015), the IEEE ICDM Conference (2015), and the ACM KDD Conference (2016). He served as an associate editor of the IEEE Transactions on Knowledge and Data Engineering from 2004 to 2008. He is an associate editor of the IEEE Transactions on Big Data, an action editor of the Data Mining and Knowledge Discovery Journal, and an associate editor of the Knowledge and Information Systems Journal. He has served as an editor-in-chief of ACM SIGKDD Explorations as well as ACM Books. He currently serves as the editor-in-chief of the ACM Transactions on Knowledge Discovery from Data. He has served as the vice-president of the SIAM Activity Group on Data Mining and as a member of the SIAM industry committee. He is a fellow of the SIAM, ACM, and the IEEE, for “contributions to knowledge discovery and data mining algorithms.”



Chapter 1

Probability and Statistics: An Introduction

“Chance is the only source of true novelty.”— Francis Crick

1.1 Introduction

Machine learning builds mathematical models from which the predictions are made by learning from data samples. The predictions are naturally probabilistic because the samples only provide an incomplete view of the entire data. There are several ways in which probability and statistics are used in machine learning:

- Probability and statistics are used to build probabilistic models in machine learning. For example, a data set may contain multiple properties of individuals, such as their age, salary, credit score, and so on. Can one model how the individual’s credit score is predicted from other attributes? Probabilistic techniques are used to model these types of relationships and use them for prediction.
- Various methods from probability and statistics are used to quantify the confidence level in the predictions of machine learning models. If the same model is run on different subsets of the data, one might arrive at different conclusions. Are the core conclusions artifacts of specific statistical quirks of a particular data sample, or can they be trusted to provide a consistent view across different samples? This is important because there are inherent limitations to data collection in the real world, and one must often make predictions using small data samples.
- In many cases, the data is best understood with the use of statistical summaries and visualization techniques. This type of exploration is often a precursor to more detailed analysis.

This book will introduce probability and statistics, while providing numerous examples from machine learning in order to illustrate various concepts.

1.1.1 The Interplay Between Probability, Statistics, and Machine Learning

The difference between probability and statistics is that between modeling the likelihood of future events and analyzing the frequency of past events. For example, if one throws a fair die, probability theory asserts that the chance of a particular face appearing is one out of six. However, if the die is actually thrown a number of times, the precise fraction will be different from the expected value of one out of six because of the natural variability associated with actual outcomes. In other words, the *sample outcomes* from a probabilistic model are only approximations of the *probabilistically expected outcomes*. Probability theory is useful for modeling expected outcomes, whereas statistics is useful for analyzing sample outcomes.

Probabilistic models are especially important when the probability of an event is not trivially obvious (like the case of the fair die). For example, consider a die with an irregular weight distribution so that the probabilities of the different faces are unknown. In such cases, a practitioner can make a *hypothesis* or *model* in which the probabilities of different faces are treated as *parameters* of the model. These parameters are then estimated from data that are outcomes of experiments (e.g., repeated die throws). This final model with “filled in” parameters can be used to predict the most likely outcomes of subsequent throws.

The basic principles in machine learning are very similar, wherein the observed data are assumed to be outcomes of experiments over a probabilistic model with incompletely specified parameters. The design of the probabilistic model is the most important job of the data analyst. Most probabilistic models in machine learning make several simplifying assumptions, because they are only approximations of a highly complex real world. One can view such a simplifying assumption in machine learning as a “hypothesis”—the accuracy of any conclusion from the resulting model is highly dependent on the reasonableness of the underlying assumption. For example, the probability of a die face might depend not only on the die but also on surface on which the die is thrown. However, in the real world, this effect is minimal and therefore one may not need to account for the surface on which the die is thrown while modeling face probabilities. Most models of real-world phenomena also use simplifying assumptions and are approximations of their true behavior. An important skill learned by data analysts over time is the ability to make reasonable assumptions that work well in practice.

1.1.2 Chapter Organization

This chapter is organized as follows. The next section introduces how data is represented in machine learning applications. General statistical methods that are used to summarize and visualize data are discussed in section 1.3. The concept of probability distributions is introduced in section 1.4. Hypothesis testing is discussed in section 1.5. Section 1.6 introduces key problems in machine learning and their connections to probability and statistics. A summary is given in section 1.7.

Table 1.1: An example of a multidimensional data set

Name	Age	Gender	Race	ZIP Code	Education
John S.	45	M	African American	05139	Doctorate
Ahanu C.	31	M	Native American	10598	Bachelors
Sayani A.	22	F	South Asian	10547	Bachelors
Jack M.	56	M	Caucasian	10562	Masters
Wei L.	63	F	East Asian	90210	Masters

1.2 Representing Data

The simplest form of data used in statistics and machine learning is *tabular data*, which is also referred to as *multidimensional data*. This data typically contains a set of *observations*, which are represented by rows in the data table. An observation is also referred to as a *data point*, *database record*, *instance*, *example*, *transaction*, *entity*, *tuple*, *object*, *sample*, or *feature-vector*. Each observation contains a set of *fields*, which are also referred to as *attributes*, *dimensions*, *variables*, or *features*. These terms will be used interchangeably throughout this book. These fields are represented by columns of the table, and they describe the different properties of the specific observation (row) at hand. For example, consider the demographic data set illustrated in Table 1.1, which contains a total of five observations corresponding to the demographic properties of five individuals. Here, the demographic properties of an individual, such as age, gender, and ZIP code, are illustrated. A multidimensional data set is a collection of row vectors organized in matrix form, which is defined as follows:

Definition 1.1 (Multidimensional Data) *A multidimensional data set is an $n \times d$ data matrix $D = [x_{ij}]_{n \times d}$, which may also be represented by a table containing n rows and d columns. The n rows of the data matrix are denoted by the n row vectors, $\vec{x}_1 \dots \vec{x}_n$, such that each vector \vec{x}_i contains a set of d features or variables denoted by $[x_{i1} \dots x_{id}]$. Each vector is also referred to as an **observation**.*

Most of this book will work with multidimensional data because it is the simplest form of data and establishes the broader principles on which the more complex data types can be processed. A data set with a single dimension ($d = 1$) is referred to as *univariate data*, since it has a single variable. A multidimensional data set with $d > 1$ is correspondingly referred to as *multivariate data*. When a data set has multiple variables, one might sometimes restrict the analysis to a single dimension — this process is referred to as *univariate analysis*. For example, finding the average value of each dimension is a form of univariate analysis, even when it is performed on all the dimensions (as long as the results from different dimensions are not combined together in some way). Similarly, analyzing multiple dimensions together is referred to as *multivariate analysis*. For example, determining whether two variables (such as age and education level) are related in some way is multivariate analysis.

The n data points $\vec{x}_1 \dots \vec{x}_n$ in d dimensions are represented by an $n \times d$ matrix denoted by D . The i th row of this matrix is \vec{x}_i , which is always assumed to be a row vector in this book. The (i, j) th entry of the matrix D is therefore the j th dimension of \vec{x}_i , which is x_{ij} . The relationship between the $n \times d$ matrix D and its individual entries x_{ij} is denoted by $D = [x_{ij}]_{n \times d}$.

1.2.1 Numeric Multidimensional Data

The attributes in Table 1.1 are of two different types. The age field has values that are numerical in the sense that they have a natural ordering, and the distances between the different values can be quantified. Such attributes are referred to as *numeric* or *quantitative*. Thus, when each value of x_{ij} in Definition 1.1 is numeric, the corresponding data set is referred to as quantitative or numeric multidimensional data. In the machine learning and statistics literature, this particular subtype of data is the most common one, and many algorithms discussed in this book work with this subtype of data. An important point is that most other data types can often be converted to numeric data through a variety of transformation methods. This is done very frequently in machine learning because the vast majority of statistical analysis methods are designed for numeric data.

Numeric data can be either *continuous* or *discrete*. Discrete numeric data are drawn from a set of (typically) equidistant numeric values. For example, the birth year of a person is a discrete numeric value because it is always an integer in a particular range. Continuous numeric data can take on any value over the real number line, such as π or $\sqrt{2}$. It is noteworthy that the distinction between continuous and discrete data is a semantic one from the computational perspective, because finite-precision computers represent all data in discrete form as a multiple of the lowest level of precision. A computer does not represent $\sqrt{2}$ exactly, but only up to a particular precision depending on the computer architecture. In spite of this fact, the term discrete data is used only for integer data or data that has significantly rough granularity within the application at hand. For example, the attribute height is treated as continuous because the measurement is considered sufficiently fine-grained.

1.2.2 Categorical and Mixed Attribute Data

Many data sets in real applications may contain categorical attributes that take on *discrete unordered* values. For example, in Table 1.1, the attributes such as gender, race, and ZIP code are represented by discrete values without a natural ordering among them. If each value of x_{ij} in Definition 1.1 is categorical, then such data is referred to as *unordered discrete-valued* or *categorical*. Categorical attributes can be represented by strings (sequences of characters). For example, the race attribute in Table 1.1 contain attributes that are words describing the race of the person at hand, and there is no ordering among them. In the case of ZIP code, the attribute values *seem* numerical at first glance, but they are really categorical (i.e., sequences of numerical characters) because there is no inherent ordering among them. On the other hand, certain attributes (like education in Table 1.1) *seem* categorical at first glance because they are represented by words, but there is an ordering among the values. The education level of a person, such as Bachelors, Masters, and doctorate, are clearly ordered, but there is no numerical value associated with them. Since these attributes are not directly embedded on a numerical range, their analysis is quite different from numerical attributes in the context of statistical applications. Such attributes are referred to as *ordinal*. The final column of Table 1.1 is an ordinal data type. Data sets that contain multiple types of attributes (such as categorical and numeric) are considered *mixed-attribute data types*. Table 1.1 contains multiple types of attributes and is therefore a mixed-attribute data type.

The categorical attribute corresponding to gender has only two possible values. In such cases, it is possible to impose an artificial ordering between these values and use algorithms designed for numeric data for this type. This is referred to as *binary* data, and it can be considered a special case of either numeric or categorical data. Binary data is considered a

“bridge” to transform numeric or categorical attributes into a common format that enables convenient statistical analysis in many scenarios. For example, a categorical data attribute can be transformed to numeric data using the process of *one-hot encoding*, wherein each value of the categorical attribute can be transformed into a binary attribute. For example, for the ZIP code attribute, one can create as many binary attributes as the number of ZIP codes. The binary attribute for a ZIP code takes on the value of 1, when the underlying attribute corresponds to that specific ZIP code; the binary attributes for all other ZIP codes take on the value of 0. Therefore, one-hot encoding ensures that exactly one binary attribute derived from an underlying categorical attribute takes on the value of 1 and others take on the value of 0.

Numeric data can also be converted into binary data through *discretization*, wherein a binary attribute is defined for a range of values of the numeric attribute. Therefore, multiple binary attributes corresponding to all ranges of the numeric attribute can represent the numeric attribute approximately. All multidimensional data sets, whether they are numeric or categorical, can be converted to numeric multidimensional data. Numeric multidimensional data is particularly convenient from a statistical perspective, because many forms of statistical analysis assume that the underlying data is numerical. Simply speaking, it is easier to design mathematical and statistical functions, when working with numeric data. As a result, there is a significant focus in the machine learning community to convert arbitrarily complex data types (such as *graphs* or *discrete sequences*) into numeric multidimensional data. This process is referred to as *feature engineering*.

The one-hot encoding approach is inherently redundant, because exactly one of these attribute values can be 1, and other attribute values take on the value of 0. Therefore, in a data set with m ZIP codes, knowing the value of the first $(m - 1)$ ZIP code columns can be used to predict the value of the m th column — if the first $(m - 1)$ columns are 0s, then the m th column takes on the value of 1, and 0, otherwise. Therefore, an alternative way of encoding categorical attributes is to use any $(m - 1)$ ZIP codes as the generated attributes, and treat the “missing” ZIP code as a *reference attribute value*. In this representation, *at most* one of the attributes takes on the value of 1, and the reference value of the ZIP code is taken on when all $(m - 1)$ binary attributes take on the value of 0. The one-hot encoding approach is common in the machine learning community, whereas the reference encoding approach is common in the statistics community. The reference encoding approach has the advantage that it retains better interpretability in many real-world applications, such as *regression*. Some of the uses of the reference encoding approach will be explored in Chapter 7.

Example 1.1 Consider an employee database with multiple attributes. Indicate the types of attributes for (i) years of employment, (ii) salary, (iii) department identifier, (iv) seniority in level number, and (v) manager flag indicating whether person has managerial responsibilities.

Solution: The solutions to the different parts are as follows:

- (i) The years of employment should be treated as a discrete numeric attribute if it is stored at a low level of granularity (such as the integer number of years). Otherwise, it should be treated as a continuous numeric attribute.
- (ii) The salary attribute typically has sufficient granularity to be considered a continuous-valued numeric attribute.
- (iii) The department identifier is a categorical attribute because there is typically no

ordering across departments or distance between them.

(iv) The seniority level is an ordinal attribute. There is an ordering among levels but it is difficult to define a distance between a pair of levels.

(v) The manager flag is a binary attribute with two possible values. It is a special case of both categorical and discrete numeric data. ■

Example 1.2 Suppose that you have a machine learning algorithm that works only on binary multidimensional data. How would you use it on a mixed-attribute data set containing both categorical and numeric attributes?

Solution: The numeric attributes should be transformed to binary attributes with the use of discretization. The categorical attributes should be transformed to binary attributes with the use of one-hot encoding. The resulting data set is a binary multidimensional data set, and it can be used with the aforementioned algorithm. ■

1.3 Summarizing and Visualizing Data

In real-world applications, one is often faced with large volumes of data, and it is useful to have an overview of the data in terms of summary statistics. Two common forms of summaries of the data distribution include *measures of central tendency* and *measures of dispersion*, which are as follows:

- *Measures of central tendency:* The measures of central tendency identify representative points corresponding to central regions of the data. For example, the average of a set of points can be considered a measure of central tendency. However, other measures of central tendency also exist, such as a point so that half of the points are greater than it and the other half are less than it. Such a point is referred to as the *median*. Note that ranking-based measures can be used not just to compute the central points but other placements within the data distribution. For example, one can compute specific *percentile* values within the data distribution. This type of measure is often used in all types of standardized exam reporting such as the *Scholastic Aptitude Test (SAT)*.
- *Measures of dispersion:* The measures of dispersion model the degree of spread of the distribution from the center of the data. The simplest approach is to compute the average of the absolute distance from the mean of the data. This measure is referred to as the *mean absolute deviation*. Since computing absolute values is inconvenient for many mathematical operations (such as taking the derivative in calculus), a more common approach is to average the squared deviations, which is referred to as the *variance*. Other measures of dispersion include the distance between specific percentile points, such as the 25 percentile point and the 75 percentile point of the ranked data values. This distance is referred to as the *inter-quartile range*. The inter-quartile range is essentially the width of the range of values containing the middle 50% of the data.

The measures of central tendency and dispersion are univariate summary statistics because they are based on only one dimension. Several forms of summary statistics, such as *covariance* and *correlation* provide an idea of how different attributes are related to one another.

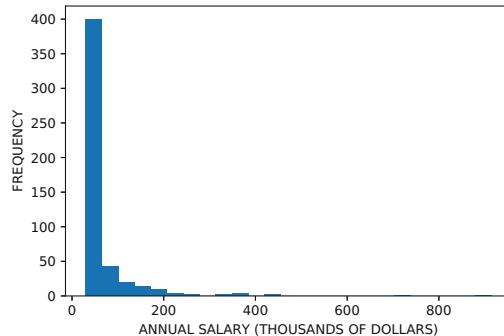


Figure 1.1: An example of a visualization of the distribution of salaries in a firm with 500 employees

For example, age and salary are two attributes that are often positively associated with one another, and this will result in the correlation between the two attributes being positive. These types of summary statistics are forms of multivariate analyses.

While summary statistics can provide a rough idea of the location and extent of the data distribution, the resulting understanding is often quite incomplete because it provides very little idea of the frequencies in different parts of the data. Which ranges of values in the data have the highest frequency? Are these ranges contiguous? Such questions can be answered by using graphical descriptions of the data. An important phase of machine learning and statistical analysis is to explore and visualize the underlying data, because the choice of technique used may sometimes be data-sensitive. Note that exploring a data set visually helps one obtain a more detailed understanding the true distribution of the underlying data than a raw statistical summary, such as an average. For example, consider a firm with 500 employees in which the mean annual employee salary is claimed to be 64,300 dollars. While this may seem to be a reasonable value, a different picture emerges when one plots the frequency distribution of the number of employees over different ranges of salaries (cf. Figure 1.1). This representation is called a *histogram*, which partitions the variable of interest (salary in this case) into bins, and then plots a frequency measure (e.g., raw frequency or relative frequency) for each bin. The X-axis in Figure 1.1 illustrates the different ranges of salaries, whereas the Y-axis illustrates the number of employees. It is evident from the histogram that more than 70% of the employees make between 20,000 and 50,000, and the high mean is an artifact of a few employees making very large salaries. These types of insights can only be obtained from detailed visualizations rather than summary statistics of the data. A key point is that this visual representation provides an understanding of the *shape* of the underlying frequency distribution, which is not possible using only measures of central tendency and dispersion.

Visualizations such as histograms are particularly useful for settings corresponding to a single attribute of the data. Therefore, histograms represent univariate visualizations of the data. How does one handle cases in which there are multiple attributes? For example, consider a data set in which the age and salary of a set of people is available. How can one visualize the nature of the association between the age and the salary? Do older people tend to earn more? These types of questions can be answered with the help of a visualization referred to as a *scatter plot*. In a scatter plot, the two attributes are represented along the two axes, and each point is represented by a single marker on the figure. An example of a scatter plot between age and salary is shown in Figure 1.2. It becomes immediately evident

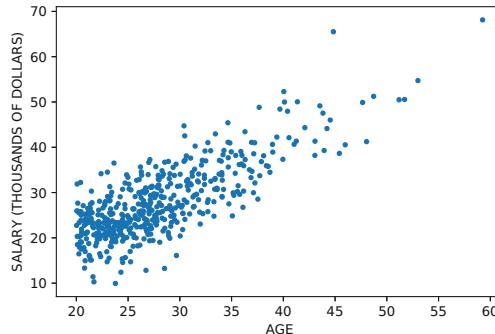


Figure 1.2: A scatter plot of salary and age. Note that older people tend to earn more than younger people.

from the scatter plot that age is positively associated with the salary. Scatter plots form an example of multivariate visualization of the data.

Many other forms of visualizations are used in order to provide intuitive views of the data. Two of the most common forms of visualization include *bar charts* and *box plots*. These different forms of visualizations provide different types of understanding of the data set at hand. A detailed discussion of data summarization and visualization methods is given in Chapter 2. The chapter also discusses several preprocessing methods that are used to prepare the data with the use of various measures of central tendency, dispersion, and inter-attribute correlation.

1.4 The Basics of Probability and Probability Distributions

Probabilities are used to quantify the fraction of the time that specific outcomes are expected to occur in experiments, which are referred to as *trials*. Therefore, probabilities are defined over a *sample space* of *outcomes*, and each outcome is associated with a probability value in $[0, 1]$. The sum of the probabilities over all the outcomes of a trial is 1. For example, when a six-sided die is thrown, each throw is a trial; the sample space of outcomes corresponds to all possible faces of the die denoted by $\Omega = \{1, 2, 3, 4, 5, 6\}$, and the probability value associated with each of the six outcomes is $1/6$. One can represent the probability values in a sample space in *tabular form*, wherein the outcomes and their corresponding probability values are contained in columns. Therefore, the resulting table has six rows corresponding to the different outcomes and their probabilities. The set of outcomes and their probabilities for the case of the six-sided die throw is illustrated in Table 1.2. Table 1.2 represents the simplest form of a probability distribution, which is also referred to as a *probability mass function* for the case of discrete data.

Another example of the probabilities associated with a set of outcomes is that of the number of times a die shows up as the face-value of 2 in ten throws of the die. In such a case, the outcome is the number of times that this face shows up in ten throws. In this case, the face 2 might show up any number of times between 0 and 10, and therefore the table will contain 11 rows. Let us refer to each show of this face as a “success.” The corresponding outcomes and their probabilities are illustrated in Table 1.3. How does one compute the

Table 1.2: Outcomes for a die throw and their probabilities

Face	Probability
1	1/6
2	1/6
3	1/6
4	1/6
5	1/6
6	1/6

Table 1.3: Outcomes for number of successes in 10 die throws and their probabilities

Number of Successes	Probability
0	0.1615
1	0.3230
2	0.2907
3	0.1550
4	0.0543
5	0.0130
6	0.0022
7	0.0002
8	< 0.0001
9	$\ll 0.0001$
10	$\ll 0.0001$

probabilities in this table? It turns out that a formula exists to compute the probability of each possible numeric outcome in terms of the total number n of throws, the “success” probability p of the face 2 appearing in a single throw, and the (discrete) numeric value k of the outcome:

$$\text{Probability of } k \text{ successes} = \frac{n!}{(n - k)!k!} p^k (1 - p)^{(n-k)} \quad (1.1)$$

In cases where the outcome is computed as a function of a set of parameters (e.g., n and p) and the discrete numeric outcome (e.g., number k of successes), an explicit table of probabilities is no longer needed to represent the probabilities of various outcomes in the sample space. This type of scenario is particularly convenient for algebraic analysis of probabilities in closed form. Such representations are referred to as closed-form probability distributions. There are many types of closed-form probability distributions, and the particular one shown above is the *binomial distribution*. Such representations are more convenient than cumbersome tabular representations in which each outcome is paired with a probability value; for example, the tabular representation for a billion throws of the die would require a billion and one rows in the table, whereas the distribution-centric approach can calculate the probability in a single formula. This type of concise representation is more useful for modeling purposes because the closed-form representation allows various types of analyses like calculus-based optimization.

In cases where the outcomes of the probability distribution are discrete numeric values, the resulting distribution is referred to as a discrete probability distribution, and the resulting table (or formula) of probability values is referred to as a probability mass function. The probability mass function for the binomial distribution of Table 1.3 is illustrated visually in terms of a histogram of relative frequencies (probabilities) Figure 1.3. One can already begin to see the relationship between probability distributions and visual statistical methods

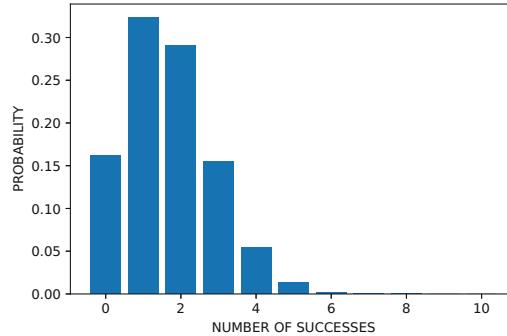


Figure 1.3: The probability mass function for a binomial distribution created by 10 die rolls

such as histograms. Several probability distributions arise frequently in machine learning, such as the Bernoulli and the binomial distributions. As we will see in later chapters, these distributions are useful for various types of machine learning models.

Since the sample space represents all possible outcomes, the sum of the probabilities over these values is 1. For example, consider the case where the probability of outcome x_i is denoted by $p(x_i)$. Examples of the pairs $(x_i, p(x_i))$ are shown in Tables 1.2 and 1.3. Then, the sum of the probabilities over all possible values of x_i is 1:

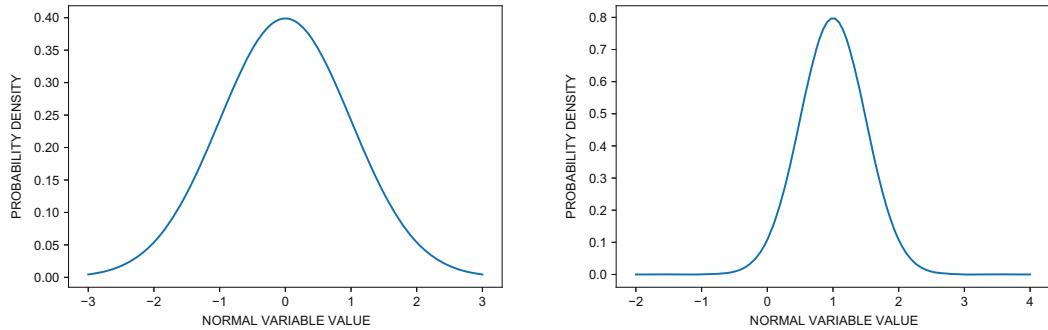
$$\sum_{x_i \in \Omega} p(x_i) = 1$$

What happens in cases where the outcomes are continuous? In this case, the sample space Ω is infinitely large. For example, if one throws a dart that is equally likely to hit anywhere on the number line from 0 to 10, the outcome would be a real-valued quantity in $(0, 10)$. It would be impossible to create a table of outcomes because there are infinitely many outcomes in Ω , the probability of each of which is infinitesimally small. In such cases, probability values can be defined only over *ranges of outcomes*. For example, the probability that the dart hits exactly at 1.25 is essentially 0, but the probability that the dart hits somewhere between 1.2 and 1.3 is $0.1/10$, which is 0.01. In such cases, it is meaningful to talk of *probability density functions* in which the area enclosed by the probability density function $f(x)$ and the X-axis between two values (a, b) of x yields the probability that the random variable takes on a value in the range (a, b) . In the dart example above, the density function is uniform and takes on the constant value of $f(x) = 0.1$ across the entire range of possible values of x in $(0, 10)$. Therefore, the density function can be expressed as follows:

$$f(x) = \begin{cases} 0.1 & \text{if } 0 \leq x \leq 10 \\ 0 & \text{otherwise} \end{cases}$$

Note that the area under the entire curve is 1, whereas the area under the probability density curve between 1.2 and 1.3 is $0.1 * (0.3 - 0.2)$, which is 0.01. This is also equal to the probability that the dart lands between 1.2 and 1.3. In general, the probability that the random variable x takes on a value in the range $[a, b]$ is given by the following:

$$\text{Probability that } x \text{ lies in } (a, b) = \int_{x=a}^b f(x)dx$$



(a) Mean of 0 and variance of 1

(b) Mean of 1 and variance of 0.25

Figure 1.4: The probability density function for two examples from the family of normal distributions

The example of the distribution induced by the modeling of the dart throw is referred to as the *uniform distribution*. A more common distribution that arises in statistics is one in which the data is clustered close to the mean and the probability density function is shaped like a bell curve. This distribution is referred to as the *normal distribution*, and is illustrated in Figure 1.4. The probability density function of a normal distribution is parameterized by its mean μ and variance σ^2 . The mean refers to the central point in the distribution, which is roughly equal to the average of a large number of samples from the distribution. The variance is a measure of the *spread* of the data distribution in terms of how “tightly” it is distributed around the mean. It is equal to average squared distance (from the mean) of a large number of sampled points from the distribution. The particular normal distribution with mean 0 and variance 1 has the following probability density function:

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2) \quad (1.2)$$

The exponential nature of the underlying probability density function ensures the clustering of the data around the mean value of 0, which is reflected in a corresponding bell-shaped curve. A more general formula for the probability density of the normal distribution with mean μ and variance σ^2 is as follows:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (1.3)$$

As in the case of all probability density functions, the total probability over all possible outcomes is always 1 unit:

$$\int_{x=-\infty}^{+\infty} f(x)dx = 1$$

Many classical distributions that are represented in closed form, such as the normal distribution, represent an *infinite family* of distributions with a characteristic shape — this shape varies with the parameters of the distribution such as μ and σ . Examples of two members of the family of normal distributions are shown in Figure 1.4. It is evident from the figure that increasing σ causes the distribution to be more widely spread out around the mean. The study of important families of distributions is useful from the perspective of machine learning, because many real-world phenomena are often modeled by these types

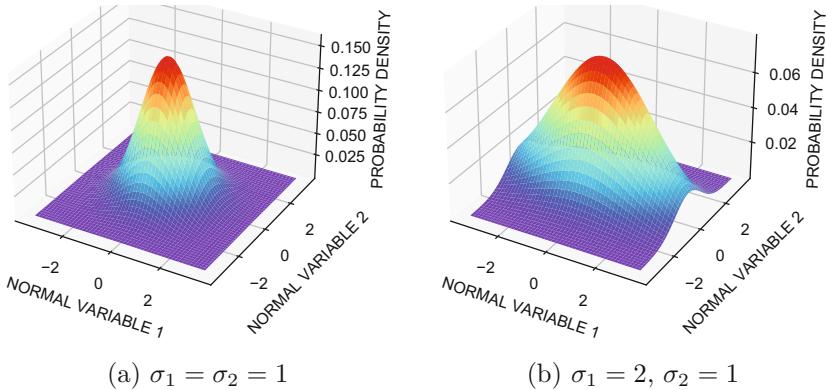


Figure 1.5: The probability density functions for two multivariate normal distributions with zero mean. The first has equal variance along two directions and the second has four times the variance in the first direction as compared to the second direction.

of distributions. The basics of probability are introduced in Chapter 3, whereas probability distributions are introduced in Chapter 4.

The aforementioned discussion assumes that the underlying probability distributions are based on a single numeric outcome. Such probability distributions are referred to as *univariate*. In practice, many probability distributions are based on multiple numeric outcomes, and such probability distributions are referred to as *multivariate*. In such cases, the underlying probability mass and density functions are multivariate as well. An example of the density function of a d -dimensional multivariate normal distribution with mean μ_i along the i th dimension and corresponding standard deviation σ_i is given below:

$$f(x_1, \dots, x_d) = \frac{1}{(2\pi)^{d/2} \prod_{i=1}^d \sigma_i} \exp \left(-\sum_{i=1}^d \frac{(x_i - \mu_i)^2}{2\sigma_i^2} \right) \quad (1.4)$$

Note that the input is a set of d arguments and the output is a density value. Two examples of the multivariate normal distribution are shown in Figure 1.5. Since the total probability of all possible outcomes is one unit, the following holds:

$$\int_{x_1=-\infty}^{+\infty} \int_{x_2=-\infty}^{+\infty} \dots \int_{x_d=-\infty}^{+\infty} f(x_1, \dots, x_d) dx_1 dx_2 \dots dx_d = 1$$

Although multivariate probability distributions are more powerful at representing relationships among data attributes, they are inherently more complex to use in probabilistic models. This difficulty is because the underlying variables may be related to one another, and such relationships may be hard to infer with a limited amount of data. Therefore, numerous simplifying assumptions are used in such cases. Many of these assumptions are introduced in various chapters of the book.

1.4.1 Populations versus Samples

Most real-world questions arise in the context of a potentially large collection of objects that is often difficult to collect in a comprehensive way. For example, suppose that a biological statistician is trying to quantify the mean length of all one-year-old sharks over the

last decade. In order to achieve this goal, one must have measured all sharks of age one over the last decade. This is obviously a lost cause from a practical perspective. This set of “potentially uncollectable” data is referred to as the *population*. In the real-world, the population is sometimes hard to identify in a comprehensive way; nevertheless, making the assumption that it is some fixed, large, and clearly defined collection of items is necessary to develop the theoretical machinery needed in statistics. In some applications, the population may even be infinitely large. The field of statistics attempts to *estimate* different properties of the population of sharks by using a smaller sample of sharks of age one. The sample is created by *probabilistically selecting* elements from the population. This approach is referred to as *probability sampling*. It is noteworthy that the estimated length of the shark will vary with the sample that is collected. *Any data set used in machine learning can be viewed as a sample of some fixed and definite population*, and the process of data collection¹ is some implicit form of sampling from the population. Although the resulting estimate of a property from a sample may not be a true measure of the corresponding value over the entire population, it is often close enough to be useful in practice. Almost all of machine learning works with samples of an underlying data population of potentially infinite size, and therefore the underlying inferences are only estimates of what one might obtain if one had access to an unlimited amount of data. Here, it is important to understand that even though an analyst typically has access to only one data set, it should still be viewed as a sample from some population that cannot be collected in totality (in most real-world applications). In general, if two analysts collect different data samples using exactly the same methodology, they will typically obtain slightly different estimates of the quantity being predicted. For example, each sample would have different measures of dispersion and central tendency simply based on random variations in the collection process. Clearly, this variation in predictions is indicative of an underlying error that most analysts are willing to live with.

1.4.2 Modeling Populations with Samples

Data-driven analysis is one way of estimating the properties of populations by making the assumption that the available data is a probability sample of some unknown population. In some cases, it may be desirable to reconstruct the closed-form probability distribution of the population in order to make predictions about data with incompletely specified attributes. Unfortunately, the probability distribution of the population may be arbitrarily complex and cannot be exactly reconstructed with only a sample from the population. A different way of dealing with this practical problem is to *model* the population probabilistically by making *simplifying assumptions*. For example, one may assume that the data in the aforementioned population of shark lengths is generated from a particular family of well-known probability distributions, and use this assumption to estimate the parameters of the distribution with the use of *statistical estimation methods* such as *maximum likelihood estimation*. Such assumptions are often simplifications (other than in toy settings like coin tosses), because most real-world populations do not follow standard families of probability

¹The process of data collection in the real world has all types of quirks, which may affect how the underlying population is defined. Most forms of data collection have sufficient flaws that it is practically impossible to draw a theoretically representative sample from a pre-defined population. Nevertheless, one can always assume that a data sample is a representative of some population, even if it does not perfectly represent the originally intended one before collection.

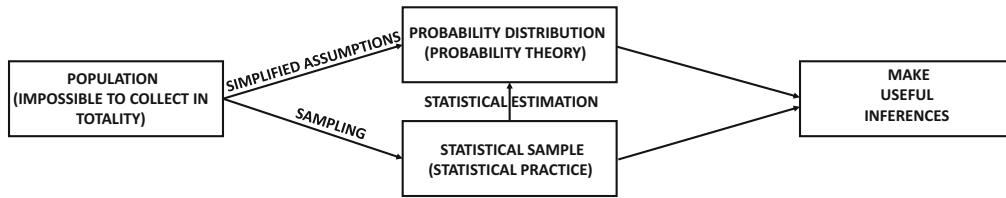


Figure 1.6: How populations and samples interact with probability and statistics in machine learning

distributions that can be expressed in closed form. For example, if one assumes that the length of sharks follows a normal distribution, one can estimate the mean and variance of this normal distribution using our sample of sharks. The modeled and estimated normal distribution is then used to make various types of complex inferences in machine learning. This type of approach is particularly useful when the underlying inferences are complex and require a lot of data for accurate predictions. Many models in machine learning, such as *classification and regression*, make assumptions about the distribution of the underlying population to make inferences. The overall understanding of how the machine learning process fits in with probability and statistics is shown in Figure 1.6. The figure shows that a sample is drawn from the population and used to learn the parameters of a (simplified) probability distribution. This simplified probability distribution is often used to make predictions on new data. Figure 1.6 is important because it will be repeated over the course of the book while illustrating the inner workings of various parts of this figure.

One important issue with sampling from populations (i.e., collecting data) is that it may sometimes not be reflective of the population that was intended to be analyzed, which can cause *sample selection bias* in the predictions derived from such samples. In the aforementioned shark example, if the intention is to model the length of all sharks, it is important not to choose to sample sharks only from a particular geographical locality (for convenience or other reasons). Such an approach would lead to inaccurate modeling of the probability distribution and subsequent errors in downstream applications. The importance of collecting the data in a way that does not lead to sample selection bias is important, and is often under-appreciated by data analysts.

Example 1.3 Suppose that you are analyzing the heights of chimpanzees with age at least one year anywhere on the planet on a specific date in the near future. Is it possible to ever collect the population of heights? If you did manage to collect the heights, do you think that any known family of probability distributions (such as the normal distribution) could exactly define this population?

Solution: Since it is not possible to measure all chimpanzees anywhere on the planet on a specific date (or even over an extended period), it is impossible to collect the population of chimpanzee heights (even with allowances on date of measurement). The heights will not follow any neat distribution, and there will be numerous variations at different points in the distribution owing to varying heights of different subspecies. The most that one could hope for is to collect a large number of samples of heights from various places on the planet and model this sample with a grossly

simplified distribution. All of probability, statistics, and machine learning works on this assumption. The population is almost always a theoretical construct for statistical modeling and not a practical one. ■

Example 1.4 You are trying to estimate the interest in video games in the US teenage population. Therefore, you create a survey with questions about interest level in several popular video games. You visit different homes in your neighborhood and hand out this survey and tabulate the interest level in different video games. Can you imagine what can go wrong with this type of survey?

Solution: The problem with this type of survey is that it is not representative of the US teenage population. A neighborhood is often a relatively homogeneous group of people with demographic and socioeconomic characteristics that are unrepresentative of the broader US population. These differences may also be reflected in differences in video game tastes. Therefore, the results from the survey are likely to be inaccurate (as far as representing the US population is concerned). It is noteworthy that data collection processes in machine learning are often biased as well. These biases are reflected in the accuracy of the underlying results. ■

1.4.3 Handing Dependence in Data Samples

Multidimensional data sets correspond to scenarios in which the different data samples are assumed to be drawn from populations (and their corresponding probability distributions) via the process of independent sampling. In the example introduced earlier on estimating the length of sharks, each length is assumed to be an independent sample from the same population. What happens in cases when there is a natural dependence among the data samples? For example, if the words in a sentence can be assumed to be samples from a probability distribution, then successive words in a sentence are always related to one another in some way and cannot be assumed to be independent samples drawn over a distribution of words. Such probabilistic dependence can be formally modeled with the use of discrete-state Markov processes, which are discussed in Chapter 10. Such processes are useful in a variety of machine learning applications, including in text, sequences, and graphs. Markov processes are the most common models used to address complex data types with dependencies among attributes.

1.5 Hypothesis Testing

Hypothesis testing is a useful application of statistics in machine learning, particularly during model evaluation. Hypothesis testing is necessitated by the fact that the true accuracy of models is defined over the entire population, which is often inaccessible in practice. For example, imagine a credit scoring application that uses past credit activity to compute scores for individuals. In order to create a truly accurate model, one might need the entire every possible set of credit scores and their associated features that has ever existed (which defines the population). However, since the population cannot be exhaustively collected, one is forced to work with data samples. Since machine learning works with samples of the

data, the accuracy of a particular model may vary with the sample used. As a result, the relative performances of different models may also vary with the sample at hand. This type of variability is particularly common with small data samples, which are common in many data-centric settings.

In hypothesis testing, one tries to show that one model is better than the other by attempting to statistically disprove a “null hypothesis,” which states that the two models are similar. This is achieved by computing the probability that the difference in their performance is a result of natural statistical variability (which could occur in the presence of the null hypothesis). When this probability is very low, it is concluded that one model is statistically better than the other. In other words, the null hypothesis is rejected. One can even create *confidence intervals* containing a range of values in which the true (i.e., population-centric) difference in accuracy between the two models must lie with high probability (with the typical definition of “high probability” being either 95% or 99%).

The design of confidence intervals is a useful framework in machine learning, because one can use it for applications beyond comparing pairs of models. The most basic application is to bound the performance measure of a particular machine learning model within a range in which the true value would lie if the entire population were available. In principle, one can use it for a variety of performance measures. There is a school of thought in machine learning evaluation that believes that all results in machine learning should be presented with confidence intervals in order to bound the uncertainty in the underlying results. Hypothesis testing methods can also be used for testing the mutual independence of features and for selecting them by discriminative power in machine learning applications. Hypothesis testing and its applications are discussed in Chapter 5.

1.6 Basic Problems in Machine Learning

Machine learning is about constructing models on observed examples and using these models to make predictions about missing entries or properties of previously unseen examples. Many of these models are probabilistic in nature. This process is also referred to as *learning*, which is where “machine learning” derives its name. Throughout this book, we assume that we have an $n \times d$ data matrix D , which contains n examples of d -dimensional data points in its rows. Therefore, the rows of the data matrix D are $\vec{x}_1 \dots \vec{x}_n$, and the (i, j) th entry of D is x_{ij} . For example, in a medical application, each row of the data matrix D might correspond to a patient, and the d dimensions might represent the different attributes garnered from the patient, such as their height, weight, test results, and so on. This set of n observations can be viewed as n samples from a (possibly infinite) population of observations, which may be unavailable in practice. Machine learning uses these examples for various applications, such as that of predicting the value of a particular dimension in the data, finding anomalous patients, or grouping similar patients. These correspond to classical problems in machine learning, such as classification, anomaly detection, and clustering. This section will introduce these classical problems.

1.6.1 Clustering

The problem of clustering is that of partitioning the rows of the $n \times d$ data matrix D into groups of similar rows. For example, imagine a setting where one has data records in which the rows of D correspond to different individuals, and the different dimensions (columns) of D correspond to the number of units of each product bought in a supermarket.

Then, a clustering application might try to segment the data set into groups of similar individuals with particular types of buying behavior. The number of clusters might either be specified by the analyst up front, or the algorithm might use a heuristic to set the number of “natural” clusters in the data. One can often use the segmentation created by clustering as a preprocessing step for other analytical goals. For example, on closer examination of the clusters obtained from the transactions in a grocery store, one might discover that some groups of individuals are interested in household articles, whereas others are interested in fruits and vegetables. This insight can be used by the merchant to make recommendations or targeted sales promotions.

One can define the clustering algorithm as that of mapping data points to group identifiers. Many clustering algorithms are inherently probabilistic because they assume that the samples of observations are drawn from a particular probability distribution², and then they try to estimate the parameters of the model in a data-driven manner. Various clustering algorithms like the *expectation-maximization algorithm* are based on probabilistic models in which each point has a probability of belonging to a cluster, where a cluster is modeled as a multivariate normal distribution. This algorithm is discussed in Chapter 9. This approach follows the framework of Figure 1.6 because we are modeling a population with a simplified distribution, and then estimating the parameters of the distribution in a data-driven manner in order to make useful inferences (e.g., making recommendations from clusters). The process of making inferences us shown in the rightmost block of Figure 1.6. Clustering models are discussed in detail in Chapter 9.

Clustering is an example of *unsupervised learning* because the groups are created without any pre-conceptions of how the groups in the data are distributed. In other words, examples of group identifiers (along with data points) are not provided to supervise the group creation process in unsupervised learning. On the other hand, supervised learning methods build a model to map the data examples to either group identifiers or numerical values based on a *training data set* of examples containing both observations and group identifiers. Such methods are the topic of discussion in the next section.

1.6.2 Classification and Regression Modeling

The problem of classification is closely related to that of clustering, except that more guidance is available for grouping the data with the use of the notion of *supervision*. In the case of clustering, the data is partitioned into groups without any regard for the types of clusters we wish to find. In the case of classification, the *training* data are already partitioned into specific types of groups. Therefore, in addition to the $n \times d$ data matrix D , we have an n -dimensional array of labels denoted by \vec{y} . The i th entry in \vec{y} , denoted by y_i , is the label of the i th row \vec{x}_i in the data matrix D , and the former is a categorical label defining a semantic name for the cluster (or *class*) to which the i th row of D belongs. In the case of the grocery example above, we might decide up front that we are interested in the classes $\mathcal{L} = \{ \text{fruits}, \text{poultry}, \text{all else} \}$. Note that the observations belonging to a particular class might be clustered in the data³, although it is not guaranteed.

In general, the labeling of data is done on the basis of application-specific considerations, and the mapping between classes and clearly distinguishable clusters is only approximate.

²As discussed in later chapters, the classical distribution used for this type of modeling is a mixture of different multivariate normal distributions. Each such normal distribution can be viewed as the representation of a cluster. Many of these concepts will be introduced more rigorously in later chapters. For now, we focus on setting up the intuition.

³In other words, the corresponding rows are similar in data matrix D .

For example, it might be possible that other distinct clusters might exist that are corresponding to specific sub-categories within the *all else* label. This might be the case because the end-user (e.g., merchant) might not have any interest in identifying items in the *all else* category, whereas the other labels might help the merchant identify candidate customers for a promotion. Therefore, in the classification problem, the training data defines the clusters of *interest* with the use of examples. The actual segmentation of the rows is done on a separate $n_t \times d$ test data matrix D_t , in which the labels are not specified. Therefore, for each row of D_t , one needs to map it one of the labels from the set \mathcal{L} . This mapping is done with the use of a classification *model* that was constructed on the training data matrix denoted by D and label vector \vec{y} . Because of the association between the rows of D and the elements of \vec{y} , it is possible to build a model that maps how the attributes in D are associated with the labels in \vec{y} . The test data is *unseen* during the process of model construction, as the rows of D and D_t are not the same. The constructed model is then used to map the rows of D_t to labels.

A common setting in classification is that the label set is *binary* and only contains two possible values. In such a case, it is common to use the label set \mathcal{L} from $\{0, 1\}$ or from $\{-1, +1\}$. This setting is also referred to as *binary classification*. The goal is to *learn* the i th entry y_i in \vec{y} as a function of the i th row \vec{x}_i of D :

$$y_i \approx f(\vec{x}_i)$$

The function $f(\vec{x}_i)$ is often parameterized with a weight vector $\vec{w} = [w_1, w_2, \dots, w_d]^T$, which is assumed to be a d -dimensional column vector. Consider the following example of binary classification into the labels $\{-1, +1\}$:

$$y_i \approx f_{\vec{w}}(\vec{x}_i) = \text{sign}\{\vec{w} \cdot \vec{x}_i^T\}$$

Note that we have added a subscript to the function to indicate its parametrization. Such models are referred to as *linear classification models*, because the function $f_{\vec{w}}(\vec{x}_i)$ is linear in its argument \vec{x}_i , and uses the parameter vector \vec{w} to learn the data-driven relationships. How does one compute \vec{w} ? The key idea is to penalize mismatching between the *observed value* y_i and the predicted value $f(\vec{x}_i)$ with the use of carefully constructed *loss functions*. Therefore, many machine learning models reduce to the following optimization problem:

$$\text{Minimize}_{\vec{w}} \sum_i \text{Mismatching between } y_i \text{ and } f_{\vec{w}}(\vec{x}_i)$$

As we will see in Chapter 8, the underlying loss functions often have a probabilistic interpretation. Once the weight vector \vec{w} has been computed by solving the optimization model, it is used to predict the value of the class variable for instances in which only the feature variables are known but the class variable is not known. For example, consider the case where \vec{z} is a row vector containing the features. Then, the class label of the test instance is predicted as $\text{sign}\{\vec{w} \cdot \vec{z}^T\}$. One can also predict the labels for the entire $n_t \times d$ test data matrix D_t whose rows contain unlabeled observations like \vec{z} . In such a case, the column vector of n_t predictions can be obtained by applying the element-wise sign function to the column vector $D_t \vec{w}$.

Classification is also referred to as *supervised learning*, because it uses the training data to build a model that performs the classification of the test data. In a sense, the training data serves as the “teacher” providing supervision. The ability to use the knowledge in the training data in order to classify the examples in *unseen* test data is referred to as *generalization*. There is no utility in classifying the examples of the training data again,

because their labels have already been observed. The only utility of the model is to generalize the knowledge about the associations between \vec{x}_i and y_i to unknown regions of the space. Classification models are discussed in detail in Chapter 8.

1.6.2.1 Regression

The label in classification is also referred to as *dependent variable* or *outcome*, which is categorical in nature. This is because each y_i in classification is a label, corresponding to a category. In the regression modeling problem, the $n \times d$ training data matrix D is associated with an n -dimensional column vector \vec{y} of dependent variables, which are *numerical*. Therefore, the only difference of regression modeling from classification is that the array \vec{y} contains numerical values (rather than categorical ones), and can therefore be treated as a vector. The dependent variable is also referred to as a *response variable*, *outcome*, *target variable*, *dependent variable*, or *regressand* in the case of regression. The independent variables are also referred to as *regressors*. Binary response variables in classification are closely related to regression, and some models solve binary classification *directly* with the use of a regression model (by pretending that the binary labels are numerical). This is because binary values have the flexibility of being treated as either categorical or as numerical values. However, more than two classes like $\{\text{Red}, \text{Green}, \text{Blue}\}$ cannot be ordered, and are therefore treated differently from regression.

The regression modeling problem is closely related to linear algebra, especially when a *linear optimization model* is used. In the linear optimization model, we use a d -dimensional column vector $\vec{w} = [w_1 \dots w_d]^T$ to represent the weights of the different dimensions. The i th entry y_i of \vec{y} is obtained as the dot product of the i th (transposed) row \vec{x}_i of D and \vec{w} . In other words, the function $f(\cdot)$ to be learned by the following linear model:

$$y_i = f(\vec{x}_i) = \vec{w} \cdot \vec{x}_i^T$$

In other words, the outcome y_i is a linear function of \vec{x}_i^T using the *regression coefficients* in the column vector \vec{w} . One can also state this condition across all training instances using the full $n \times d$ data matrix D and the dependent variable vector \vec{y} :

$$\vec{y} \approx D\vec{w} \quad (1.5)$$

Note that this is a matrix representation of n linear equations. In most cases, the value of n is much greater than d , and therefore, this is an *over-determined* system of linear equations. In over-determined cases, there is usually no solution for \vec{w} that *exactly* satisfies this system. However, we can minimize the sum of squares of the errors to get as close to this goal as possible:

$$J = \frac{1}{2} \|D\vec{w} - \vec{y}\|^2 \quad (1.6)$$

This type of optimization function is referred to as a *loss function*. As we will see in Chapter 7, this loss function is rooted in a probabilistic assumption about the elements of the vector $(D\vec{w} - \vec{y})$, and an optimal value of \vec{w} yields the “most likely” probability distribution that fits the elements of this vector. Therefore, this optimization problem is defined as follows:

$$\text{Minimize}_{\vec{w}} J = \frac{1}{2} \|D\vec{w} - \vec{y}\|^2$$

On solving the aforementioned optimization problem, the solution \vec{w} can be shown to be the following:

$$\vec{w} = (D^T D)^{-1} D^T \vec{y} \quad (1.7)$$

As in the case of classification, an $n_t \times d$ test data matrix D_t is available, for which the response variable values need to be inferred. Then, for each row \vec{z} of the test data matrix D_t , the dot product of \vec{w} and \vec{z}^T is the corresponding prediction of the real-valued dependent variable. It is also possible to predict the numerical response of all n_t rows simultaneously as the n_t -dimensional column vector $D_t\vec{w}$. The regression problem is discussed in detail in Chapter 7. Regression modeling methods are important because they can be generalized to settings in which the response variable is of a different type (e.g., ordinal or categorical). The resulting models are often used for classification as well after modification. Therefore, regression serves as one of the most important “bridge” problems of linear algebra, machine learning, and statistics.

1.6.3 Outlier Detection

In the outlier detection problem, we have an $n \times d$ data matrix D , with rows denoted by $\vec{x}_1 \dots \vec{x}_n$ — we would like find rows of D that are very different from most of the other rows. This problem has a natural relationship of complementarity with the clustering problem, in which the aim is to find groups of similar rows. In other words, outliers are rows of D that do not naturally fit in with the other rows. Therefore, clustering methods are often used to find outliers. Many of the methods used to find outliers are also probabilistic in nature. A probabilistic definition of outliers was provided by Hawkins [35]:

An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different [probabilistic] mechanism.

The majority of the points that are indeed generated by a particular probabilistic mechanism are referred to as *inliers*. The aforementioned definition by Hawkins forms the basis for probabilistic algorithms for outlier detection. For example, it is assumed that the data is generated from a particular probability distribution. Then, the parameters of this probability distribution are estimated in order to maximize the likelihood that the data was generated from the model. This is a standard approach used in machine learning, referred to as maximum-likelihood estimation, which is also used in other machine learning models. The resulting probability distribution can be used to generate either a probability or a probability density of a data point being generated from the model. Data points that have low probability (or probability density) of being generated from this model are referred to as outliers. Interestingly, this approach is also used to find clusters in the data. The overall framework of maximum likelihood estimation is discussed in Chapter 6, and its use for clustering and outlier detection is discussed in Chapter 9. Chapter 9 also introduces a number of other unsupervised learning methods like *matrix factorization*, together with their connections to outlier detection. Matrix factorization methods can not only find outlier rows of the data matrix, but it can also find outlier entries and outlier columns.

Several statistical methods are also used in order to perform a particular type of outlier detection, referred to as *extreme value analysis*. Finding extreme values is closely related to techniques used in hypothesis testing and it corresponds to points that lie at the fringes of a data distribution, whereas outliers are points that may lie in the interior of the data distribution as well (as long as the probability density in those regions is low). The fringes of a data distribution are also referred to as its *tails*. Some distributions such as the normal distribution have clearly identifiable tails, whereas others such as the uniform distribution do not have tails. Extreme values are therefore special cases of outliers. Extreme value analysis is a somewhat simpler problem than general outlier analysis (because it only focuses on distribution tails), although it is no less important. Several extreme value analysis methods

are discussed in Chapters 4 and 5. Further discussion on extreme values in the context of tail inequalities is provided in Chapter 11.

Example 1.5 You have data from different emails, including their length, frequency of different words, time of receipt, sender, and so on. The emails are tagged with labels indicating whether they are spam. You want to use this data in order to construct a model that predicts whether or not a new (untagged) email is spam. Which of the models introduced in this section would you find useful?

Solution: The classification model would be useful because the spam labels can be viewed as categorical dependent variables. The characteristics of the email would be considered independent variables. ■

Example 1.6 Suppose that you have all the data for emails as the previous example (such as length and word frequency) but you do not have tags indicating whether or not the emails are spam. You want to find out whether a new email is spam. Which of the models introduced in this section would you find useful?

Solution: The outlier detection model would be useful because this is an unsupervised setting. The main assumption in being able to use an outlier detection model successfully is that the vast majority of the emails are not spam. As a result, anomalous characteristics of spam emails will be detected. ■

Example 1.7 Suppose that you have all the data for emails as the previous example (such as length and word frequency) and you want to organize them into folders by topical content. Which of the models introduced in this section would you find useful?

Solution: This is also an unsupervised setting because there are no topical tags available. In this case, the clustering model would be useful in order to partition them into similar groups of emails. ■

1.7 Summary

Probability and statistics are useful tools in both modeling and evaluation of machine learning methods. These methods are used in the entire machine learning pipeline, beginning from data visualization to modeling of data-driven algorithms and finally their evaluation. Hypothesis testing methods provide an important tool both in the evaluation and in the design of machine learning algorithms.

Probability distributions are used to model samples drawn from a potentially infinite population. These probability distributions often provide a concise and simplified representation of the data, which can be used by machine learning algorithms. The parameters of the probability distributions are often learned by a process referred to as maximum-likelihood estimation, and they are reconstructed from data samples. These reconstructed distributions are often directly or indirectly used for making predictions in machine learning. Examples of machine learning algorithms that use the reconstructed probability distributions for making predictions include recommendations, classification, regression modeling, and outlier detec-

tion. The main differences among these techniques lie in terms of how the corresponding probability distributions are modeled and leveraged. The subsequent chapters will introduce the different types of probabilistic and statistical methods, together with their applications in machine learning settings.

1.8 Further Reading

Numerous generic books on probability and statistics may be found in [9, 19, 38]. Generic discussions on data mining and machine learning are available in [1, 3, 32, 63]. An outstanding book that is specifically focused on regression is the text by Draper and Smith [20]. Data visualization methods are discussed in [60]. Two popular books on statistical learning are found in [33, 39], whereas a probabilistic approach to machine learning is discussed in [10]. Statistical learning methods with R applications are presented in [39].

1.9 Exercises

1. Toss a coin n times (possibly virtually) and plot the fraction of heads with increasing n . Explain the trend illustrated by the plot.
2. Suppose you toss a coin once and it shows up heads. Would it make you suspect that the coin is biased? What if you tossed it ten times and it shows up heads each time? [A systematic way to make the judgement is *hypothesis testing*.]
3. Suppose that you conduct a survey of all your classmates about their political affiliation and tabulate the results in order to estimate the fraction of people in the United States who are Democrats. Discuss the sources of error in this type of approach.
4. Suppose you have data containing numeric attributes. However, you have a machine learning application that works only with categorical values. What type of preprocessing can you perform on the data in order to make the application work on your numeric data set?
5. You have data containing demographic attributes and political affiliation. Given a new person for which you know only demographic attributes, you want to predict political affiliation. Which machine learning model would you use?
6. Consider a modification of the previous problem with the difference that your data now contains demographic attributes and height. You are now trying to predict the height. Which machine learning model would be useful in solving this problem?
7. Suppose that you have customer buying behavior along with demographic data. You want to segment the customers into homogeneous groups based on their demographic attributes so that you can analyze the buying behavior of each group separately. What type of machine learning model would be useful for this type of segmentation?
8. While entering numeric multidimensional data into a spreadsheet you suspect that some egregious errors were made in the data entry process. You would like to delete these rows from your spreadsheet. What type of machine learning model would you find useful for detecting such rows?

9. Suppose that you have a machine learning algorithm that can perform regression with numerical dependent variables. At the same time, you have a classification data set in which the labels are Blue, Green, and Red. Propose one or more simple ways in which you can modify the regression algorithm for classification.
10. Suppose that you score each point in a data set as the Euclidian distance to its nearest neighbor. Argue why this score is a measure of outlierness.



Chapter 2

Summarizing and Visualizing Data

“A picture is worth a thousand words.”— Frederick Barnard

2.1 Introduction

In the modern era, statisticians and analysts are often faced with large amounts of data, which creates the need to summarize and visualize the data before analysis. Therefore, methods are required for creating easily digestible summaries and visual representations of data. The creation of easily digestible summaries and visual representations enables the analyst to obtain a better idea of the broad patterns in the data. This chapter will introduce algorithms for summarizing and visualizing data. The discussions in these chapters primarily address two issues:

- *Summarizing data:* Data summarization discusses the generation of key statistical properties of the data, such as its measures of *central tendency* or measures of *dispersion*. Measures of central tendency try to quantify measures of central location with respect to the data values, whereas measures of dispersion quantify how much the data is spread around these central locations. Other forms of data summarization include measures of association between numeric variables, or the generation of contingency tables for categorical data.
- *Data visualization:* The analyst can often obtain a better understanding of the data with the use of data visualization techniques. Examples of such visualization techniques include the construction of histograms, scatter plots, and bar charts. Visualization techniques often help the analyst get a better understanding of the details of how the data is distributed (e.g., regions of high frequency and low frequency). In some cases, this type of understanding can provide insights about the type of model that will work best in a particular machine learning application.

Beyond providing insights about the data distribution, data summarization can be used for various types of preprocessing that are used before machine learning applications. For example, the data is often *standardized* with translation and scaling, which requires computation of the measures of central tendency and dispersion on the original data — such measures help in selecting scaling and translation quantities, so that the standardized data has specific measures of central tendency and dispersion. These types of preprocessing methods allow off-the-shelf algorithms to have more robust behavior in real-world applications.

2.1.1 Chapter Organization

This chapter is organized as follows. The next section introduces several important measures of summarization for both univariate and multivariate data. Section 2.3 discusses numerous visualization techniques for univariate and multivariate data. Applications of summary statistics to data preprocessing and standardization are introduced in section 2.4. A summary is given in section 2.5.

2.2 Summarizing Data

Various types of univariate and multivariate summary statistics are useful for characterizing important properties of the data. In this section, various methods for constructing summary statistics of the data will be discussed.

Chapter 1 discusses the distinction between samples and populations in statistics. It is noteworthy that all the summary statistics discussed below are defined with respect to *samples* of the population, and these are only estimates of their true values in the entire population. Since the entire population can never be collected in most real-world applications, these values provide excellent estimates of their true values across the entire population. In some cases, the population summaries *can* be known, if specific domain knowledge is available. An example of such domain knowledge would be that the number of births is equally partitioned into males and females, and therefore one can directly know the average fraction of females in the population of births. This type of domain knowledge is relatively rare in real-world applications, and therefore one has to estimate these values with sample statistics.

2.2.1 Univariate Summarization

The two most common forms of univariate summarization include the generation of measures of central tendency and measures of dispersion. This section will introduce several measures of each type.

2.2.1.1 Measures of Central Tendency

The simplest measure of central tendency is the mean of the data, which is a simple arithmetic average of the values in the data:

Definition 2.1 (Sample Mean) *Given a sample of n values x_1, x_2, \dots, x_n , the sample mean $\hat{\mu}_X$ of the values is defined as follows:*

$$\hat{\mu}_X = \frac{\sum_{i=1}^n x_i}{n}$$

Note the circumflex on top of $\hat{\mu}_X$ to indicate that it is an estimated value from a sample, and it is not the population mean μ_X . Throughout the book, the circumflex is used to denote an estimated value from a sample as opposed to its true value (from the population). One can also compute the mean of a set of multivariate values $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$ by the expression $\sum_{i=1}^n \vec{x}_i/n$, which yields a multidimensional vector of means in which each entry corresponds to the mean of a particular attribute.

As an example, the sample mean of the set of values 2, 3, and 7 obtained as instantiations of the random variable X is given by $\hat{\mu}_X = (2 + 3 + 7)/3 = 4$. One problem with the mean is that it is obtained by averaging the raw values, and therefore it is sensitive to a few extreme values of the points. The addition of a single extreme value (possibly caused by data collection error or an unusual event) can change the mean significantly enough to become unrepresentative of the vast majority of the remaining distribution. A more stable measure of central tendency is the *median*, which *ranks* the points and reports the central value of this set of points.

Definition 2.2 (Sample Median) *Given a sample of n values x_1, x_2, \dots, x_n , the sample median of the values is defined by sorting the points in increasing order and reporting the value of the central point in this sort order.*

In the event that n is odd, the central point is the $[(n+1)/2]$ th point in the sort order. On the other hand, if n is even, then any point x between the $(n/2)$ th number and $(n/2+1)$ th number is technically a median, because half the points are less than x and the other half are greater than x . A common approach to resolve this ambiguity is to use *interpolation tie-breaking*, wherein the $(n+1)/2$ th value is still used as the median (as in the case of odd n), and the fractional value of $(n+1)/2$ is interpreted via interpolation with respect to the two surrounding integer ranks. In particular, the median is defined as the average of the $(n/2)$ th and $(n/2+1)$ th points in the sort order, when n is even. For example, the median of 2, 3, and 7 is 3, because there are an odd number of points in the data. However, if we have the four points 2, 3, 4, and 7, then both 3 and 4 are central points. As a result, the value of the median is $(3+4)/2 = 3.5$. Therefore, when the number of data points is even, an additional *interpolation* step is needed to compute the median, because the central point is not directly available in the data. One way to think of this interpolation is as a computation of the $[(n+1)/2]$ th point in the data in the sort order, where a fractional value of $(n+1)/2$ is resolved via interpolation of the values in the two surrounding ranks. This general principle can be generalized to many other rank-based statistics (as subsequently discussed in this section).

The mean is a preferred measure of central tendency when the data does not have extreme values. On the other hand, since a few extreme values can distort the mean, it is sometimes preferable to use the median as a more stable measure of the central tendency. In order to understand this point, consider a village of 50 families in which 49 families make 50,000 dollars an year, whereas a single family makes ten million dollars an year. Note that this single family is not really representative of the village, and the median value of 50,000 dollars is an excellent measure of central tendency of the village. On the other hand, the mean annual income of the village is 249,000 dollars an year. This value is roughly five times the average income of 98% of the villagers, and is therefore not a good measure of the central tendency.

Example 2.1 Consider a very large set of numbers containing repetitions, which are drawn from the set $\{1, 2, \dots, 9\}$. Suppose that the frequency of integer i in the set is

exactly proportional to $1/i$, where i varies between 1 and 9. Compute the sample mean and sample median.

Solution: The total frequency is proportional to $\sum_{i=1}^9 (1/i) = 7129/2520 \approx 2.829$. In order to compute the sample mean, each number is multiplied with its total frequency, added, and then divided by the total frequency. Multiplying each integer i and its proportional frequency $(1/i)$ and adding, one obtains 9. Dividing 9 with the total frequency, one obtains the following mean:

$$\hat{\mu} = 9 * 2520/7129 \approx 3.193$$

Next, we compute the median. Since the sum of the frequencies is proportional to 2.829, one must find the minimum value of r for which $\sum_{i=1}^r (1/i) > 2.829/2 = 1.4145$. Note that setting $r = 2$ yields this minimum value. Therefore, the median is 2. It is noteworthy that the median is less than the mean because the majority of the data points have values 1 and 2 but the few large values like 9 increase the mean significantly. ■

Problem 2.1 *Compute the mean and median of the values 2, 2, 2, 3, 3, 3, 100. Reflect on the difference between the computed values of the mean and the median. Which measure is more representative of the central tendency?*

The notion of a median can be generalized to that of *percentiles* in the data. A median can be viewed as a 50th percentile of the data distribution because it is obtained by sorting the data, and selecting (or interpolating) a central point so that 50% of the points lie on either side of it. The notion of a percentile generalizes this concept. so that any percentage value $p \in (0, 100)$ can be used, and $p\%$ of the data lies below the p th percentile. The value of p is typically an integer although the definition applies to real values as well. It is noteworthy that there are some variations in the definition of the percentile, and the differences can sometimes be quite significant when the data contains fewer than 100 items. This particular exposition works with the following convention. First the n points are sorted from the smallest to the largest value and tagged with integer rank values from 1 to n (breaking ties arbitrarily). The 0th percentile maps to the rank-1 point (i.e., smallest point) in the data and the 100th percentile maps to the rank- n point (i.e., largest point) in the data. All 99 other integer percentile values are converted into (possibly decimal-valued ranks) by linearly mapping all integer percentiles in $[1, 99]$ to “ranks” in the (continuous) range $(1, n)$; note that this linear mapping needs to be consistent with the mapping of the 0th percentile and 100th percentile points to ranks 1 and n , respectively. The resulting sequence of 101 (possibly fractional) ranks for $p \in \{0, 1, \dots, 100\}$ is as follows:

$$\underbrace{\frac{(0 \cdot n + 100)}{100}}_1, \frac{(n + 99)}{100}, \frac{(2n + 98)}{100}, \dots, \frac{pn + (100 - p)}{100}, \dots, \underbrace{\frac{(100n + 0)}{100}}_n$$

Since 101 percentile values are mapped to the rank range $[1, n]$ of length $(n - 1)$ (with 100 gaps between the various ranks), the difference between the ranks of successive percentile values is $(n - 1)/100$. The percentiles that do not map to integer-valued ranks are interpolated from the two immediately surrounding integer ranks in a weighted way.

The sample percentile¹ is rigorously defined as follows:

Definition 2.3 (Sample Percentile) *Given a sample of n values x_1, x_2, \dots, x_n , the p th percentile (or $f = p/100$ fractional percentile) of the values is defined by sorting the points in increasing order and reporting the value of the $k = (n \cdot f + 1 - f)$ th item in this sort order. In the event that the calculated value of k is a fraction, a weighted interpolation from the two immediately surrounding integer ranks with respect to the computed value of k is used.*

What does “weighted interpolation” in the aforementioned definition mean? For example, if the computed value of k is 10.3, the corresponding value is a weighted average of the values at the 10th and 11th ranks, where the weights of the two ranks are derived from the fractional part of 10.3 as $(1 - 0.3) = 0.7$ and 0.3, respectively. The 10th rank is given the larger weight of 0.7 (since it is closer to 10.3 than the 11th rank) and the 11th rank is given the smaller weight of 0.3. Then, if x'_{10} and x'_{11} are the 10th and 11th ranks, the weighted average for the percentile rank is as follows:

$$x'_{10.3} = 0.7x'_{10} + 0.3x'_{11}$$

There are two particular values of the percentile that are extremely important in data analysis. The *first quartile* corresponds to a percentile value of 25%, whereas the *third quartile* corresponds to a percentile value of 75%. The difference between the third quartile and the first quartile is a nonnegative value, referred to as the *inter-quartile range*. Note that the middle 50% of the data is contained between the first and third quartiles, and it often corresponds to the most “representative” part of the data.

Example 2.2 Compute the $[100/3]$ th-percentile of the set of data points 1, 4, 5, 7. Repeat this computation for the set of points 1, 4, 5, 7, 9.

Solution: The fractional value of the percentile is $f = 1/3$. One needs to find the k th item in the sort order of the first set of $n = 4$ points, where k is defined as follows:

$$k = n \cdot f + 1 - f = 4/3 + 1 - 1/3 = 2$$

Therefore, the 2nd point in the sort order is reported, which is 4.

For the second case of $n = 5$ points, the value of k is computed as follows:

$$k = 5/3 + 1 - 1/3 = 7/3 = 2.333333$$

Therefore, one needs to interpolate between the second and third points, where the second point value of 4 is weighted by $2/3$ (since 2 is closer to $7/3$) and the third point value of 5 is weighted by $1/3$. The interpolated value is $4(2/3) + 5(1/3)$, which is $13/3 \approx 4.33$. ■

¹It is noteworthy that there are different definitions of the percentile. In some definitions, $p\%$ of the points must be *strictly* less than the p th percentile. Therefore, the 100th percentile does not exist by this definition. The definition in this book assumes that the least value is always the 0th percentile and the largest value is the 100th percentile. The other percentile values are interpolated between these extremes. This definition is simple and is used by default in many forms of commercial software (including the most basic data-processing applications like Excel). In most cases, the differences between the definitions are quite small unless the number of data values is very small as well.

Problem 2.2 Compute the 40th percentile for the same sets of points in the above example. In other words, you want to use the sets of data points 1, 4, 5, 7 and 1, 4, 5, 7, 9.

Problem 2.3 A data set with n distinct points is such that both quartiles can be selected from the data without the use of interpolation. Show that $(n - 1)$ must be divisible by 4.

Another important summary statistic is the *mode*, which is nominally considered a measure of central tendency, although the location of the mode may sometimes be far from central regions of the data (especially if the high-frequency regions occur at the extremes of the data distribution). Furthermore, the mode can be rigorously defined only for either categorical data or it can be defined for discrete numeric data, when only samples of the data are available and the actual population or generating distribution is not available (as is commonly the case). However, it can be defined in a heuristic manner for samples of continuous numeric data, and the choice of the mode will depend on the heuristic used. For discrete data, the mode is the value in the data with the largest frequency.

Definition 2.4 (Sample Mode for Discrete Data) Given a sample of n (numeric discrete or categorical) values x_1, x_2, \dots, x_n (with possible repetitions), the sample mode is the discrete value in the set with the largest frequency. If multiple values have the largest frequency, the data may have multiple modes.

Data sets in which a single mode exists are referred to as unimodal, whereas data sets with multiple modes are referred to as multimodal. A data set with two modes is referred to as bimodal.

The notion of mode is sometimes extended to samples of continuous data in a heuristic manner by creating bins of the entire numeric data range, and then reporting the middle of the bin with the largest frequency as the mode. However, such an approach can sometimes result in ambiguity, since the answer can depend drastically on the scheme used for creating the ranges of values, especially when the amount of available data is limited. However, if the true probability distribution of the entire population is known (rather than the knowledge on only a sample), the mode of the continuous distribution can be defined precisely as the peak of the distribution. The mode of a continuous-valued probability distribution corresponds to the value at which the peak probability density of the distribution is attained. As in the case of categorical data, it is possible to define unimodal, bimodal, and multimodal distributions in cases where the corresponding probability distributions respectively have a single peak, two peaks, or multiple peaks of equal height. In some cases, this definition is used in a relaxed and loose way when the distribution contains multiple peaks are of *roughly* equal height, but the corresponding distribution is still referred to as multimodal.

Example 2.3 Compute the mode(s) of the set of discrete data values 1, 2, 2, 3, 4, 4, 5, 9, 9. Is this data set unimodal, bimodal, or multimodal?

Solution: The values 2, 4, and 9 are all modes with a frequency of 2. Since there are three modes, this is a multimodal distribution. This example also shows why a mode is not a reliable measure of central tendency. The rounded value of the mean of this data set is 4.33 and the median is 4. However, some of the modes are far away from these values of the mean and median. ■

Problem 2.4 Compute the mode of the discrete data set 1, 2, 3, 3, 3, 3, 4, 4, 5, 6, 6, 6, 7, 7, 8. Is this data set unimodal, bimodal, or multimodal?

2.2.1.2 Measures of Dispersion

Next, we will introduce measures of dispersion. Measures of dispersion quantify the level of spread in a data distribution. One measure of dispersion that has already been introduced earlier is the inter-quartile range of a sample.

Definition 2.5 (Sample Inter-Quartile Range) *The inter-quartile range of a sample is a nonnegative value that is the difference between the third quartile and the first quartile of a sample. This value is the length of the interval containing the middle 50% of the data.*

Like the median, the inter-quartile range is not very sensitive to the points at the extreme ends of the data. For example, increasing the maximum value or decreasing the minimum value in the data sample does not change its inter-quartile range. Furthermore, translating the top 25% of the largest values of the data by positive offsets will also not change the inter-quartile range, even though the data distribution does qualitatively look much more dispersed to most observers. The insensitivity of the inter-quartile range to as much as 50% of the data (on the two ends) can be viewed as its weakness of this measure of dispersion.

Some measures of dispersion are more sensitive to the values at the two ends of the data distribution. Examples of such measures include the *mean absolute deviation*, the *variance* and its close relative, which is the *standard deviation*. The simplest measure of dispersion is the *mean absolute deviation*, which is the mean of the absolute values of the distances of the points from the sample mean:

Definition 2.6 (Sample Mean Absolute Deviation) *The mean absolute deviation (MAD) of a sample of n values x_1, x_2, \dots, x_n is defined as the average of the absolute distances of these points from their sample mean $\hat{\mu}_X = \sum_{i=1}^n x_i/n$:*

$$\hat{MAD}_X = \frac{\sum_{i=1}^n |x_i - \hat{\mu}_X|}{n}$$

Averaging the *absolute* values of the deviations is important; simply averaging the deviations without taking absolute values will result in the positive and negative deviations around the mean canceling out to 0. It is further noteworthy that the above estimate is typically an underestimate of the true deviation of data points from the mean of the population, especially when the value of n is small. This type of bias arises because the estimated mean of the sample of data points is typically much closer to the individual points in the sample on average than the true population mean is. For example, when a sample of only 1 point is available, the above formula will always provide an estimate of 0 for the mean absolute deviation, which is obviously an underestimate of the average deviation of the entire population. This is because the sample mean has the same value as the single sample of available data, and this sample mean does not accurately reflect the population mean. There are no rigorous ways of correcting for the bias in estimation of the mean absolute deviation, although a heuristic approach is to replace the value of n in the denominator (for averaging) with $(n - 1)$. Nevertheless, the estimate is still quite accurate when the value of n is large (irrespective of which denominator is used for averaging).

A more popular approach for estimating the measure of dispersion is the *variance*, which averages the squared distances from the sample mean instead of using the absolute deviations from the sample mean. Using the squared values to compute the measure of dispersion turns out to have some nice properties, since some widely used probability distributions (such as the normal distribution) directly use the variance of the underlying distribution as a parameter to control the level of dispersion in the generated samples. We consider

two cases for computing the variance — in the first case, the population mean is known from domain knowledge (e.g., fraction of heads in a population of coin tosses), and another in which the population mean is not known. Although the latter is the standard case for measuring the variance, it is instructive to examine the case in which the population mean is known a priori.

Definition 2.7 (Sample Variance with Known Population Mean) *Given a sample of data values $x_1 \dots x_n$, the variance is defined as the mean-squared deviations of the values from the known population mean μ_X :*

$$\hat{\sigma}_X^2 = \frac{\sum_{i=1}^n (x_i - \mu_X)^2}{n}$$

In most real-world settings, the population mean is not known in advance but must be estimated from the sample. In such a case, it is important to use an adjustment, which reflects the fact that the sample mean is usually closer to the sample values than the population mean. This adjustment is referred to as the *Bessel correction*, and the corrected approach divides the sum of squared deviation by $(n - 1)$ instead of n for the purpose of averaging:

Definition 2.8 (Sample Variance) *Given a sample of data values $x_1 \dots x_n$, the variance is defined as the corrected mean-squared deviations of the values from the sample mean $\hat{\mu}_X = \sum x_i/n$:*

$$\hat{\sigma}_X^2 = \frac{\sum_{i=1}^n (x_i - \hat{\mu}_X)^2}{n - 1}$$

The Bessel correction *rigorously* corrects the bias caused by using the sample mean in lieu of the population mean. It is noteworthy that it is not possible to estimate the variance with only one sample, because both the numerator and denominator of the equation evaluate to 0 in this case. This is because a single sample provides no information about the distances among different data values. For large values of n , the correction to the standard deviation has very little effect from a percentage-wise point of view. In fact, for any data set that is more than a few hundred points, the difference between the two estimations is negligible. As a result, one often uses the simpler formula with n (rather than $(n-1)$) in the denominator in real-world applications. Ignoring the Bessel correction also has the advantage of simplifying the computational formula for the variance. These simplifications are discussed later in this chapter (page 34).

An important observation about the variance is that it uses the squares of the deviations, which has an effect on the units in terms of which the variance is expressed. For example, if the original values are measurements in meters, the variance is in the units of squared meters. This makes the variance somewhat harder to interpret as compared to the mean absolute deviation. The standard deviation is a related measure that uses the square-root of the variance instead in order to ensure that the measure of dispersion is in the same units as the data values:

Definition 2.9 (Sample Standard Deviation) *Given a sample of data values $x_1 \dots x_n$, the sample standard deviation is defined as the corrected root-mean-squared deviations of the values from the sample mean $\hat{\mu}_X = \sum x_i/n$:*

$$\hat{\sigma}_X = \sqrt{\frac{\sum_{i=1}^n (x_i - \hat{\mu}_X)^2}{n - 1}}$$

As in the case of the mean absolute deviation, the Bessel correction is used in case of small sample sizes, although its practical use in machine learning is limited. It is noteworthy that the mean and standard deviation are also associated with probability distributions, and they represent the *expected estimates* of these values on a large number of samples. These distribution specific versions are discussed in section 3.9.1 of Chapter 3.

The standard deviation is equal the mean absolute deviation as long as there are at most two (possibly repeated) distinct values in the data set — in the case of two distinct values, both need to have equal frequency. When there is only one distinct value in the data set, the mean absolute deviation and standard deviation are both 0. In every other case, the standard deviation is larger than the mean absolute deviation (see Example 2.7). When there are outlier points in the data, the difference between the standard deviation and mean absolute deviation is particularly large. This is because of the squaring effect of the variance, which is not present in the case of the mean absolute deviation.

Example 2.4 Find the mean absolute deviation and standard deviation of 1, 1, 1, 3, 3, 3. Do not use the Bessel correction. Repeat the exercise for the set of points 1, 3, 3, 3, 3, 3.

Solution: The mean in the first case is 2, and each point has an absolute deviation of 1. As a result, both the mean absolute deviation and standard deviation are 1. In the second case, the mean of the set of points is $8/3$ with mean absolute deviation $[5(1/3) + (5/3)]/6 = 5/9$. The variance is $[5(1/3)^2 + (5/3)^2]/6 = 5/9$. The corresponding standard deviation is $\sqrt{5/9}$, which is larger than the mean absolute deviation value of $5/9$. Other than in very special cases, the standard deviation is almost always larger than the mean absolute deviation. ■

Example 2.5 Suppose that you have a set of n data values $x_1 \dots x_n$, and each value lies in $[a, b]$. Show that the sample variance of these values is bounded above by $\frac{n(b-a)^2}{4(n-1)}$.

Solution: The variance is largest when each data value is either a or b . This point can be proven by method of contradiction. Suppose that some x_i is strictly between a and b . The partial derivative of the variance with respect to x_i can be shown to be proportional to $(x_i - \hat{\mu}_X)$. This means that the variance of the data set can be increased by reducing x_i if x_i is below the current mean and increasing x_i , otherwise. Therefore, each x_i cannot lie strictly between a and b in order for the variance to be as large as possible.

Next, assume that a fraction f of the values x_1, x_2, \dots, x_n take on the value of a and the remaining take on the value of b . Then, the mean is $af + b(1-f)$, and the Bessel-uncorrected variance can be shown to be $(b-a)^2 f(1-f)$. Since the correction factor is $n/(n-1)$, it follows that the corrected variance is $(b-a)^2 f(1-f)n/(n-1)$. By using differential calculus, it can be shown that the optimal value of f is 0.5, which yields the above bound. ■

Problem 2.5 Show that the sample standard deviation of two points is proportional to the distance between them. What is the proportionality factor?

Problem 2.6 Find the standard deviation of the points 0, 3, 3.

2.2.2 Multivariate Summarization

The univariate summary statistics discussed thus far are only useful for modeling the behavior of a single attribute. A more useful concept is that of *multivariate summary statistics*. In multivariate summary statistics, we attempt to find the relationships among different attributes. Does a person's age increase with their salary? How about their height? Relating different attributes of the data to one another is a critical step in machine learning applications. This is because machine learning applications often predict missing values in observations. As we will see in Chapter 7, machine learning applications like simple regression have deep connections with some of the multivariate summary statistics that quantify associations between attributes.

2.2.2.1 Covariance and Correlation

The simplest form of multivariate summary statistics is the *covariance*, which generalizes the concept of variance to multiple attributes, and it is defined as follows:

Definition 2.10 (Sample Covariance) *Given two paired samples $x_1 \dots x_n$ and $y_1 \dots y_n$ of data values (where x_i is paired with y_i), the covariance between these paired sets is defined as the average product of deviations of these values from their corresponding sample means $\hat{\mu}_X = \sum_i x_i/n$ and $\hat{\mu}_Y = \sum_i y_i/n$:*

$$\hat{\sigma}_{XY} = \frac{\sum_{i=1}^n (x_i - \hat{\mu}_X)(y_i - \hat{\mu}_Y)}{n - 1}$$

As in the case of variance, a value of $(n - 1)$ is used for averaging instead of n as a correction mechanism for the fact that sample means are used instead of population means in the covariance calculation. However, since most machine learning applications work with relatively large values of n anyway, it is common to use n for averaging rather than $(n - 1)$ without a significant change to the estimate. One advantage of using n for averaging (i.e., ignoring the Bessel correction) is that it allows the use of some simple and intuitive formulae for computation of the covariance and variance without cumbersome tweaks to the formula:

Lemma 2.1 *Consider two paired samples $x_1 \dots x_n$ and $y_1 \dots y_n$ of data values (where x_i is paired with y_i). The unadjusted covariance between these paired sets satisfies the following relationship:*

$$\frac{\sum_{i=1}^n (x_i - \hat{\mu}_X)(y_i - \hat{\mu}_Y)}{n} = \frac{\sum_{i=1}^n x_i y_i}{n} - \hat{\mu}_X \hat{\mu}_Y$$

Here, $\hat{\mu}_X$ and $\hat{\mu}_Y$ represent the sample means of the two attributes. In other words, the covariance can be intuitively understood as the difference between the mean of products and the product of means.

Proof: By distributing the products on the left-hand side, one obtains the following expression:

$$\frac{\sum_{i=1}^n (x_i - \hat{\mu}_X)(y_i - \hat{\mu}_Y)}{n} = \frac{\sum_{i=1}^n x_i y_i}{n} - \frac{\sum_{i=1}^n x_i}{n} \hat{\mu}_Y - \hat{\mu}_X \frac{\sum_{i=1}^n y_i}{n} + \hat{\mu}_X \hat{\mu}_Y \sum_{i=1}^n \frac{1}{n}$$

On simplification of the above expression by using $\hat{\mu}_X = \sum_i x_i/n$ and $\hat{\mu}_Y = \sum_i y_i/n$, the result follows. ■

The aforementioned Lemma is extremely useful for processing data incrementally in the streaming context. Suppose that you have pairs of data points arriving continuously over time and you wish to incrementally maintain the covariance without having to process all the data points. If one used the original covariance formula, one would be forced to process all data points from scratch because each quantity $(x_i - \hat{\mu}_X)$ and $(y_i - \hat{\mu}_Y)$ would change as the means change. However, the above lemma ensures that the covariance can be expressed as a separable sum over data points, which makes incremental maintenance possible. The precise approach for doing so is discussed in Example 2.9.

It is noteworthy that the notion of covariance is a generalization of the concept of variance, since the variance of a variable is the covariance of the variable with itself.

Observation 2.1 *The variance of a set of observations is equal to the covariance of the observations paired with themselves.*

Since the variance is a special case of the covariance, the results of Lemma 2.1 also apply to the case of variances. We summarize this special case as a corollary:

Corollary 2.1 *Let x_1, x_2, \dots, x_n be a set of n data values. The variance of these values (without adjusting for the Bessel correction) satisfies the following relationship:*

$$\frac{\sum_{i=1}^n (x_i - \hat{\mu}_X)^2}{n} = \frac{\sum_{i=1}^n x_i^2}{n} - \hat{\mu}_X^2 = \hat{\mu}_{X^2} - \hat{\mu}_X^2$$

Here, $\hat{\mu}_X$ is the sample mean of the data values. In other words, the variance can be intuitively understood as the difference between the mean of the squared observations and the squared mean.

The proof of this result is exactly similar to the proof of the analogous result for covariances, because the variance is a special case of the covariance.

The covariance is a measure of the association between variables — a positive covariance indicates that the variables are positively associated, whereas a negative covariance indicates that the variables are negatively associated. For example, the age and height of a person will typically be positively associated and therefore pairwise samples of age and height will have positive covariance. Unfortunately, however, the absolute value of the covariance is sensitive to the units used to measure it. For example, changing the units of measurement of height from meters to centimeters will increase the age-height covariance by a factor of 100. Therefore, the calculated value of covariance between two quantities does not provide a good understanding of the strength of the association between the two quantities.

Another way of thinking about the covariance is in terms of dot products between the mean-centered n -dimensional vectors $\vec{x}_c = [x_1 - \hat{\mu}_X, x_2 - \hat{\mu}_X, \dots, x_n - \hat{\mu}_X]$ and $\vec{y}_c = [y_1 - \hat{\mu}_Y, y_2 - \hat{\mu}_Y, \dots, y_n - \hat{\mu}_Y]$. The covariance is equal to $1/(n-1)$ times the dot-product between the mean-centered vectors \vec{x}_c and \vec{y}_c . Furthermore, the variance of a vector is equal to $1/(n-1)$ times the squared norm of its mean-centered representation.

Example 2.6 Consider a very large set of numbers drawn over $\{1, 2, \dots, 9\}$. Suppose that the frequency of integer i in the set is exactly proportional to $1/i$, where i varies between 1 and 9. Compute the variance of the set. You may use the result from Example 2.1 that the mean of the data is 3.193 and the sum of all frequencies is proportional to 2.829.

Solution: This problem is an excellent opportunity to apply Corollary 2.1. It is already known that the sum of the frequencies is 2.829. The value of $\hat{\mu}_{X^2}$ can be computed as follows:

$$\hat{\mu}_{X^2} = \frac{\sum_{i=1}^9 (1/i) \cdot i^2}{\sum_{i=1}^9 (1/i)} = \frac{\sum_{i=1}^9 i}{2.829} = \frac{45}{2.829} \approx 15.91$$

One can then use Corollary 2.1 to compute the sample variance as follows:

$$\hat{\sigma}_X^2 = \hat{\mu}_{X^2} - \hat{\mu}_X^2 \approx 15.91 - 3.193^2 \approx 5.71 \quad \blacksquare$$

Example 2.7 Use Corollary 2.1 to show that the variance of a set of values is always at least equal the square of the mean absolute deviation. Furthermore, show that the difference between the two is equal to the variance of the absolute deviations. Use this result to identify the conditions under which the variance is equal to the mean absolute deviation. The Bessel correction is ignored for the purpose of this example.

Solution: Let x_1, x_2, \dots, x_n be the data values (corresponding to instantiations of the random variable X), and let $a_1, \dots, a_n \geq 0$ be the absolute deviations of the data points x_1, x_2, \dots, x_n from the sample mean $\hat{\mu}_X$. One can treat these deviations a_1, a_2, \dots, a_n as samples from the new random variable A . Then, the sample mean absolute deviation of X can be expressed in terms of A as follows:

$$M\hat{AD}_X = \frac{\sum_{i=1}^n a_i}{n} = \hat{\mu}_A$$

The sample variance of X can be expressed in terms of A as follows:

$$\hat{\sigma}_X^2 = \frac{\sum_{i=1}^n a_i^2}{n} = \hat{\mu}_{A^2}$$

Combining the above two results, the difference between the sample variance of X and the square of the sample mean absolute deviation can be expressed in terms of A as follows:

$$\hat{\sigma}_X^2 - M\hat{AD}_X^2 = \frac{\sum_{i=1}^n a_i^2}{n} - \left(\frac{\sum_{i=1}^n a_i}{n} \right)^2 = \hat{\mu}_{A^2} - \hat{\mu}_A^2$$

Using Corollary 2.1 to deduce that the expression on the right is the sample variance of A , the following can be inferred:

$$\hat{\sigma}_X^2 - M\hat{AD}_X^2 = \hat{\mu}_{A^2} - \hat{\mu}_A^2 = \hat{\sigma}_A^2 \geq 0$$

Therefore, the difference between the variance of X and the squared mean absolute deviation of X is equal to the variance of the absolute deviations. Furthermore, since the variance of the mean absolute deviations is a nonnegative value, it follows that the variance of a sample is at least equal to its squared mean absolute deviation. The two are equal when the variance of the mean absolute deviation is zero. This occurs when all values of a are the same, and therefore all samples of X are equidistant from the sample mean of X . This situation can occur when there are only two distinct values

in the samples of X and there are an equal number of them. It is also possible to have a trivial case in which there is only one distinct (repeated) value of the samples of X (and therefore both the variance and mean absolute deviations of the samples are 0). ■

Example 2.8 Compute the covariance between the ordered sets $X = \{1, 3, 5\}$ and $Y = \{2, 4, 6\}$, where the i th element of the first set is paired with the i th element of the second set. Now double each element of the first set to $\{2, 6, 10\}$ and compute the covariance with $\{2, 4, 6\}$. What do you observe about the effect of scaling on the covariance?

Solution: The means of X and Y are $\hat{\mu}_X = 3$ and $\hat{\mu}_Y = 4$. The value of n is 3. The corresponding values of $(X - \hat{\mu}_X)$ and $(Y - \hat{\mu}_Y)$ are both $\{-2, 0, 2\}$. On using the covariance formula, the value obtained is as follows:

$$\hat{\sigma}_{XY} = \frac{2^2 + (-2)^2}{3 - 1} = 4$$

On doubling the elements of the first set, the covariance doubles as well because the elements of $(X - \hat{\mu}_X)$ double. ■

Example 2.9 Suppose that you have pairs of data points (x_i, y_i) arrive continuously over time. Show how you can maintain the means and covariances of these points incrementally by using only a constant number of calculations for each point arrival. Ignore the Bessel correction.

Solution: Lemma 2.1 is very useful in the context of this problem. When k points have been received, the following incremental statistics are maintained: (i) $SS_{xy} = \sum_{i=1}^k x_i y_i$, (ii) $S_x = \sum_{i=1}^k x_i$, (iii) $S_y = \sum_{i=1}^k y_i$, and (iv) the number of points k that have arrived thus far. A key point is that when the $(k+1)$ th point arrives, these statistics can be maintained incrementally using a constant number of operations simply by adding the contribution of the $(k+1)$ th point to each statistic. This process requires a constant number of operations. This is because the statistics are *additively separable*.

The means and covariances can be computed from these additive statistics. Specifically, the means can be computed first as $\hat{\mu}_X = S_x/k$ and $\hat{\mu}_Y = S_y/k$. After computing these means the covariance $\hat{\sigma}_{xy} = SS_{xy}/k - \hat{\mu}_X \hat{\mu}_Y$. As in the case of updates, the computation of the means and covariance also requires a constant number of operations. This approach can also be used to incrementally maintain variances. ■

One problem with the covariance is that it is sensitive to the absolute magnitudes of the points. Therefore, multiplying the values of all points in one of the sets by 2 causes the covariance to be scaled by a factor of 2 as well. A quantity that offers a better understanding of the strength of the association between two quantities is the *coefficient of correlation*, which is a normalized version of the covariance. The correlation is also equal to the covariance after normalizing each set of points to unit standard deviation (by dividing each point with the standard deviation of its corresponding set). Therefore, the coefficient of correlation is defined as follows:

Definition 2.11 (Sample Correlation) Given two paired samples $x_1 \dots x_n$ and $y_1 \dots y_n$ of data values (where x_i is paired with y_i), the coefficient of correlation ρ_{XY} between these paired sets is defined as a function of their sample covariance $\hat{\sigma}_{XY}$ and sample standard deviations $\hat{\sigma}_X$, $\hat{\sigma}_Y$ as follows:

$$\hat{\rho}_{XY} = \frac{\hat{\sigma}_{XY}}{\hat{\sigma}_X \hat{\sigma}_Y}$$

This correlation is also referred to as the *Pearson correlation coefficient*. Dividing each measured quantity with its standard deviation has the effect of making the quantities unit-less and the also dividing the co-variance between the original quantities by the product of the standard deviations (thereby making the covariance between the unit-less quantities unit-less as well). The coefficient of correlation between two sets of paired values always lies between -1 and $+1$ as it has the same algebraic form as the cosine of the angle between the vectors $\vec{x}_c = [x_1 - \hat{\mu}_X, \dots, x_n - \hat{\mu}_X]$ and $\vec{y}_c = [y_1 - \hat{\mu}_Y, \dots, y_n - \hat{\mu}_Y]$. One can therefore express the Pearson correlation coefficient as follows:

$$\hat{\rho}_{XY} = \frac{\sum_{i=1}^n (x_i - \hat{\mu}_X)(y_i - \hat{\mu}_Y)}{\sqrt{\sum_{i=1}^n (x_i - \hat{\mu}_X)^2} \sqrt{\sum_{i=1}^n (y_i - \hat{\mu}_Y)^2}} = \frac{\vec{x}_c \cdot \vec{y}_c}{\|\vec{x}_c\| \cdot \|\vec{y}_c\|} \quad (2.1)$$

It is easy to see that the above definition of the correlation computes the cosine of the angle between the vectors \vec{x}_c and \vec{y}_c . A coefficient of correlation that is close to 1 is indicative of a strong positive association, whereas a coefficient of correlation close to -1 is indicative of a strong negative association. When the coefficient is close to 0, it implies that the two attributes do not have a strong relationship and tend to vary independently of one another.

Example 2.10 Compute the coefficient of correlation between the ordered sets $X = \{1, 3, 5\}$ and $Y = \{2, 4, 6\}$, where the i th element of the first set is paired with the i th element of the second set. Can you explain why the result you obtain takes on the sign that it does? Now double each elements of the first set to $\{2, 6, 10\}$ and compute the correlation with $\{2, 4, 6\}$. What do you observe about the change in the correlation?

Solution: As in the case of Example 2.8, the value of $\hat{\sigma}_{XY}$ is 4, whereas $\hat{\sigma}_X$ and $\hat{\sigma}_Y$ are each 2. Therefore, the coefficient of correlation $\hat{\rho}_{XY}$ is 1. On doubling the first set, $\hat{\sigma}_{XY}$ doubles, but so does $\hat{\sigma}_X$. Therefore, the coefficient of correlation remains fixed at 1. ■

Problem 2.7 Show that the correlation between paired observations for $n = 2$ observations is always $+1$, -1 , or undefined. Under what condition is the correlation undefined? Can you make a general statement about when the correlation is undefined for $n > 2$?

2.2.2.2 Rank Correlation Measures

In some applications, the raw values of the attributes may not be available, but their ranks may be available. This is particularly true if the attributes are ordinal in nature. In other cases, the application at hand might necessitate computation of correlations between ranks rather than between raw values. Rank-based correlations generally tend to be more stable to outliers (i.e., unusual extreme values). A number of rank-based correlation measures enable such computations. For simplicity in notation, we assume that the attributes are numeric, although the generalization to ordinal attributes is straightforward. The simplest generalization of correlation to ranks is obtained by replacing the raw numerical values with

their ranks and then computing the Pearson coefficient of correlation between the ranks. The resulting measure of correlation is referred to as the *Spearman rank correlation*. The key point here is to first define the ranks appropriately in order to account for ties. The first step is to convert raw values to ranks, while breaking ties arbitrarily. Therefore, all values in x_1, \dots, x_n as well as the values in y_1, \dots, y_n get an integer rank from 1 through n . Subsequently, the ranks of the tied values are averaged to (possibly) fractional ranks, so that tied values map to tied ranks. For example, if ranks 2, 3, 4, and 5 are tied in raw value, these ranks are all replaced with four occurrences of $(2+3+4+5)/4 = 3.5$. Therefore, each pair of raw values (x_i, y_i) now maps to a rank pair (a_i, b_i) . Subsequently, the Spearman rank correlation coefficient is defined as follows:

Definition 2.12 (Spearman Rank Correlation) Let x_1, x_2, \dots, x_n and y_1, y_2, \dots, y_n be a set of n paired values, so that x_i is paired with y_i . Let a_i be the rank of x_i in x_1, x_2, \dots, x_n and let b_i be the rank of y_i in y_1, y_2, \dots, y_n . The Spearman rank correlation coefficient between x_1, x_2, \dots, x_n and y_1, y_2, \dots, y_n is defined in exactly the same way as the Pearson correlation coefficient, except that the computation is applied to the ranks (a_i, b_i) rather than the raw numeric values (x_i, y_i) :

$$\hat{\rho}_{XY}^s = \frac{\sum_{i=1}^n (a_i - \hat{\mu}_A)(b_i - \hat{\mu}_B)}{\sqrt{\sum_{i=1}^n (a_i - \hat{\mu}_A)^2} \sqrt{\sum_{i=1}^n (b_i - \hat{\mu}_B)^2}}$$

A simpler formula exists for calculating the Spearman rank correlation coefficient $\hat{\rho}_{XY}^s$ by using the fact that the rank averages $\hat{\mu}_A$ and $\hat{\mu}_B$ are each $(n+1)/2$:

$$\hat{\rho}_{XY}^s = 1 - \frac{6 \sum_{i=1}^n (a_i - b_i)^2}{n(n^2 - 1)}$$

Another popular measure of rank correlation is the *Kendall rank correlation*. Intuitively, the Kendall rank correlation is a value between -1 and $+1$, which tells us how frequently pairs of observations have the same ordering of the rank across two attributes.

Definition 2.13 (Kendall Rank Correlation) Let $x_1 \dots x_n$ and $y_1 \dots y_n$ be two sets of paired numeric observations, where x_i is paired with y_i . Then, the Kendall rank correlation coefficient, $\hat{\tau}_{XY}$, is defined as follows:

$$\hat{\tau}_{XY} = \frac{\sum_{i=1}^n \sum_{j=i+1}^n \text{sign}\{(x_i - x_j)(y_i - y_j)\}}{n(n-1)/2}$$

Note that the sign of $(x_i - x_j)(y_i - y_j)$ will be 0 in case of ties between the pair (x_i, x_j) or between the pair (y_i, y_j) .

A pair (i, j) is said to be *concordant* if x_i and x_j has the same ordering of rank as y_i and y_j . A pair (i, j) is said to be *discordant* if x_i and x_j has a different ordering of rank as y_i and y_j . If $x_i = x_j$ or $y_i = y_j$, then the pair (i, j) is neither concordant nor discordant. The Kendall rank correlation coefficient is the normalized difference between the number of concordant and the number of discordant pairs, where the normalization factor is the total number of pairs. Pairs that are neither concordant nor discordant are ignored in the numerator, although they are counted in the denominator, which is exactly equal to $\binom{n}{2} = n(n-1)/2$.

The Spearman rank correlation coefficient is used more popularly because of its relationship with the Pearson correlation coefficient. The Kendall rank correlation coefficient is generally more robust. In most cases, the Spearman rank correlation coefficient tends to be larger than the Kendall rank correlation coefficient.

Example 2.11 Calculate the Spearman and Kendall rank correlations between the ordered sets 1.2, 2.3, 1.6 and 3.3, 5.7, and 6.5.

Solution: The ranks in the first case are $[a_1, a_2, a_3] = [1, 3, 2]$, and the ranks in the second case are $[b_1, b_2, b_3] = [1, 2, 3]$. The value of n is 3. Plugging in the formula for the Spearman rank correlation, one obtains the following:

$$\hat{\rho}_{XY}^s = 1 - 6 \frac{0^2 + 1^2 + (-1)^2}{3(3^2 - 1)} = 0.5$$

The number of concordants is 2 and the number of discordants is 1 between the three pairs. Therefore, the Kendall rank correlation coefficient is $\hat{\tau}_{XY} = (2 - 1)/3 = 1/3$. ■

Problem 2.8 Find the Spearman and the Kendall rank correlation coefficient between 2.1, 1.3, 3.5, 4.7 and 4.1, 1.3, 3.6, 2.8 (where the two sets are ordered). Which one turns out to be larger?

A variant of the Kendall rank correlation is often used to measure the quality of rankings with respect to binary data. This type of approach has great usefulness in evaluation of machine learning algorithms like binary classification, anomaly detection, and search engines. Many of these algorithms output rankings of instances, which need to be evaluated. Consider a set of points (e.g., demographic information about customers) of which only a subset of the points are relevant to a sales promotion by the merchant. This information about the customer relevance is not known at the time of the promotion, but is known at a later stage after the time for sales promotion has passed. Nevertheless, the collected data is useful for evaluating the merchant's algorithm for future promotions. The information about relevance is assumed to be represented in binary form when it becomes available (as class labels), with a value of 1 indicating relevance and a value of 0 indicating non-relevance. The merchant has a ranking algorithm which outputs a larger numerical value for items that are considered more relevant. The Kendall rank correlation between the numerical values output by the merchant and the binary "ground-truth" values provides an evaluation of the quality of the merchant's algorithm. One problem with this approach is that most customers are assumed to be non-relevant (corresponding to a ground-truth value of 0), which causes very few concordants or discordants but a comparatively large denominator $n(n - 1)/2$ of the Kendall coefficient. Note that there are only $n_1(n - n_1)$ pairs if values out of the $n(n - 1)/2$ pairs that are either concordant or discordant. As a result, the Kendall coefficient will have very small absolute values that are never close to 1, and it is hard to interpret the correlation strength between the binary ground truth and the ranking for a specific value of the Kendall coefficient. The main problem is that the value of the Kendall rank correlation coefficient is strongly affected by the value of n_1 . Therefore, for a binary data set containing $n_1 \ll n$ relevant items and $(n - n_1)$ non-relevant items (along with a corresponding ranking), the denominator of the Kendall coefficient is changed from $n(n - 1)/2$ to $n_1(n - n_1)$:

$$n_1(n - n_1) \ll \binom{n}{2}$$

Intuitively, this change corresponds to computing the Kendall coefficient only over pairs of observations in which one observation is relevant (corresponding to a value of 1), and the other observation is 0 (corresponding to a value of 0). This change removes the effect of

ties both from the numerator and the denominator of the modified Kendall rank correlation coefficient $\hat{\tau}_{XY}^{(0/1)}$ for pairs in which one of the sets of values is binary:

$$\hat{\tau}_{XY}^{(0/1)} = \frac{\sum_{i=1}^n \sum_{j=i+1}^n \text{sign}\{(x_i - x_j)(y_i - y_j)\}}{n_1(n - n_1)}$$

This modified Kendall coefficient continues to take on a value between -1 and $+1$. Furthermore, the value $(\hat{\tau}_{XY} + 1)/2$ lies in $(0, 1)$, and can be shown to have a very intuitive interpretation — it can be interpreted as the *probability* that a randomly selected pair of relevant and non-relevant items are ranked in the correct order.

Example 2.12 (Outlier Algorithm Evaluation) Consider a set of ten points associated with binary ground-truth values indicating whether or not they are outliers. Two of them are labeled 1, because they are outliers. The other eight inliers are labeled 0. An outlier detection algorithm that does not know the ground-truth is used to score these points, where higher values indicate outlierness. The two outliers have scores of 3.1 and 7.2. The remaining eight points have scores of 1.1, 1.6, 2.1, 2.3, 2.5, 2.8, 3.3, and 3.7. Find the modified Kendall rank correlation between binary labels and outlier scores.

Solution: Since there are two outliers and eight inliers, there are a total of $2 * 8 = 16$ pairs. Among these pairs, there are 14 consistent pairs and two inconsistent pairs corresponding to the inliers with scores 3.3 and 3.7. These inliers have score greater than one of the outliers. Therefore, the modified Kendall rank correlation coefficient is $(14 - 2)/16 = 0.75$. Therefore, the modified Kendall rank correlation coefficient is 0.75, which is quite high. One can also convert this measure to a probability of correct ranking of scores of randomly sampled outlier-inlier pairs as $(1 + 0.75)/2 = 0.875$. In other words, if an outlier-inlier pair is selected at random, the algorithm will correctly give a higher score to the outlier with probability 0.875. A probability greater than 0.5 means that the algorithm performs better on average than an algorithm^a that randomly assigns scores/ranks to data points. ■

^aThis quantification is also referred to as the area under the *Receiver Operating Characteristic* (ROC AUC) of the outlier detection algorithm [5]. While the use of the ROC AUC is ubiquitous in machine learning, its relationship to the Kendall coefficient is not very well known.

2.2.2.3 Correlations among Multiple Attributes

The aforementioned discussion is focused on two attributes, and therefore it is designed to find associations between pairs of attributes. However, most machine learning applications have multiple variables, and it may be desirable to simultaneously quantify the correlations among different attributes. While such higher-order measures of correlation do exist, they tend to be cumbersome to compute. Nevertheless, even the second-order relationships (i.e., covariances) across all pairs of attributes can be used to obtain some understanding of the relationships between multiple attributes via the use of a process from linear algebra, referred to as *eigendecomposition* [6]. Although this type of quantification is a simplified one (relative to higher-order correlations), it has significant practical use in real-world applications.

Consider an $n \times d$ data matrix D with n observations and d attributes. In such cases, one can compute a $d \times d$ matrix of covariances, where the (i, j) th entry is the covariance

between feature i and feature j . This matrix is also referred to as the *covariance matrix*. The covariance matrix is defined without the Bessel correction. Such a matrix is symmetric because its (i, j) th entry and (j, i) th entry are both equal to the covariance between the attributes i and j . Furthermore, the diagonal elements of the covariance matrix are always positive because they contain the variances of the corresponding attributes. It is common to use unadjusted covariances (without the Bessel correction) to retain simplicity in notation. In most cases, such matrix-based techniques are used only when there are sufficient data points (which precludes the need for the Bessel correction in the first place). One can define the co-variance matrix succinctly in matrix notation:

Observation 2.2 (Covariance Matrix) *Let D be an $n \times d$ data matrix, and let D_c be the mean-centered data matrix obtained by subtracting the mean of each column from each element of the corresponding column. Then, the covariance matrix C (without Bessel correction) can be computed as $C = \frac{D_c^T D_c}{n}$.*

The covariance matrix is a very useful construct in machine learning, because it has a key linear-algebra property (referred to as the *positive semi-definite property*), which allows it to be decomposed into different matrices, including one that can be used to visualize the key directions of correlation in the data [6]. In particular, the $d \times d$ covariance matrix can be decomposed in order to identify the *principal component directions* in the data, which happen to be mutually orthogonal vectors. The principal component directions are *eigenvectors* of the matrix C . Formally, a d -dimensional vector \vec{e} is an eigenvector of matrix C , if it satisfies the following condition:

$$C\vec{e} = \lambda\vec{e}$$

Here, λ is a scalar, which is referred to as an *eigenvalue*. A $d \times d$ matrix can have at most d eigenvectors (that cannot be expressed as linear combinations of other eigenvectors). Symmetric matrices like C are guaranteed to have exactly d *orthonormal* eigenvectors. Furthermore, covariance matrices are guaranteed to have nonnegative eigenvalues because of the property of positive semidefiniteness [6]. These eigenvectors define the principal component directions in the data.

Definition 2.14 (Principal Component Directions) *The $d \times d$ covariance matrix of data matrix D can be decomposed as $C = P\Delta P^T$, where P is an orthogonal matrix whose columns contain the orthonormal eigenvectors and Δ is a diagonal matrix containing the nonnegative eigenvalues. The orthogonal eigenvectors with the top- k eigenvalues (for any $k < d$) represent the principal component directions in the data, so that projecting the data onto these directions results in the representation with the maximum aggregate variance along these k directions.*

Since P is orthogonal with mutually perpendicular column vectors, DP is an $n \times d$ matrix containing a rotated and reflected version of the data points in D in its rows. Observe that DP contains dot products of the rows of D with the columns of P , which correspond to the coordinates of the rows of D in this new orthogonal axis system. An important property of this new data set is that the pairwise covariances among the columns of DP are 0, and the variances of the columns in DP are the corresponding eigenvalues. However, DP_k selects and retains only k of the d coordinates contained in DP . Intuitively, these d -dimensional vectors are the “important” multivariate directions of correlation in the data. This is because the variances along the small eigenvalues are small and the attribute values along those directions in DP will be roughly constant (and not very informative). Dropping these

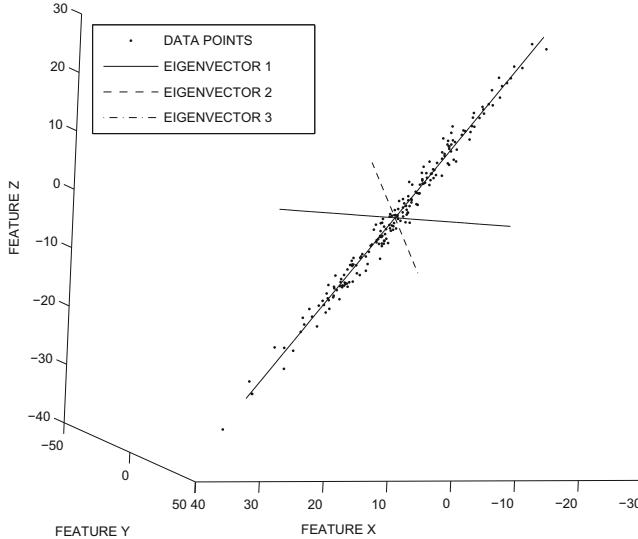


Figure 2.1: How eigenvectors relate to the data distribution

directions from the data representation will not change the pairwise Euclidean distances between points very much, and therefore many machine learning algorithms will provide similar results on the k -dimensional representations of points contained in the n rows of DP_k .

Let P_k be the $d \times k$ matrix containing the top- k eigenvectors. Then, a *reduced* $n \times k$ data matrix with the maximum *residual* variance can be represented in the $n \times k$ matrix $D_k = DP_k$. The matrix D_k represents a rotated and reflected version of the data matrix D (with $(n - k)$ dimensions dropped), and the pairwise covariances between the columns of D_k are 0. The rows of D_k contain the transformed k -dimensional representation of the data. In order to provide an understanding of how the eigenvectors of a matrix relate to the directions of correlation in the data, the eigenvectors of the covariance matrix for a 3-dimensional data set are illustrated in Figure 2.1. It is evident that most of the variance of the data is captured along the first two eigenvectors, and there is little variation along the third eigenvector. In other words, dropping the smallest eigenvectors does not have any significant effect on important statistical properties of the data, such as inter-point distances, which allows their use for machine learning and search applications. The new 2-dimensional coordinates of the data points are presented only along these two directions. Therefore, representing the data along the first pair of eigenvectors will provide a *reduced* and rotated representation of the data; this approach provides a compact and efficient representation for machine learning applications. As we will see later, the covariance matrix has significant importance in probability theory, as it can be used to model the parameters of a multivariate normal distribution from which a data set is generated.

We make some important observations that are relevant to the use of principal component analysis in machine learning.

Observation 2.3 (Properties of Principal Component Analysis) *The principal component analysis of an $n \times d$ data matrix D with covariance matrix $C = P\Delta P^T$ (with $d \times d$ eigenvector matrix P and diagonal eigenvalue matrix Δ) satisfies the following properties:*

1. If the d -dimensional points (rows) of data matrix D are projected along a single eigenvector direction \vec{e}_i as $D\vec{e}_i$ (i.e., projected along the i th column vector of P), then the variance of this 1-dimensional data set is the corresponding eigenvalue (i.e., i th diagonal element in Δ).
2. If the data D is projected along any pair of eigenvectors to create a 2-dimensional data set, the corresponding covariance between the resulting pairs of dimensions is 0. As a result, principal component analysis is often used for decorrelating transformations in data preprocessing.
3. If the top- k eigenvectors are selected for data representation, the aggregate variance of the resulting data set is greater than or equal to any rotated representation of data set D in which only k transformed dimensions are retained. This is because the values on each of the remaining $(n - k)$ dimensions are roughly constant across different data points. As a result, principal component analysis is often used for dimensionality reduction. It finds a k -dimensional representation DP_k , which is such that it retains good approximations of the distances between pairs of points. As a result, the reduced representation can be used for various machine learning applications in lieu of the original data set.

Principal component analysis is one of the most fundamental operations in machine learning, and it arises in all sorts of applications. With some further (simplified) assumptions on the probability distribution of the data along each principal component direction, the data is assumed to be distributed in an ellipsoidal shape with its axes in mutually orthogonal directions (although this assumption may not be actually satisfied for a real-world data set). The axes directions are the principal components. This type of shape of either the full data or parts of the data is a key assumption in machine learning applications. For further discussion on methods for principal component analysis, the reader is referred to a textbook on linear algebra [6].

Example 2.13 Consider a mean-centered data set with the following points:

$$\{(4, 4), (3, 3), (2, 2), (1, 1), (-1, -1), (-2, -2), (-3, -3), (-4, -4), (1, -1), (-1, 1)\}$$

Compute the covariance matrix, principal components, and eigenvalues. Use any Web calculator for eigenvector computation. Find a 1-dimensional representation of the points that loses the least variance.

Solution: Since the data is mean centered, the covariance is obtained by computing the dot product between the X -coordinate vector $[4, 3, 2, 1, -1, -2, -3, -4, 1, -1]$ and the Y -coordinate vector $[4, 3, 2, 1, -1, -2, -3, -4, -1, 1]$, and then dividing by the number of points (which is 10). The result is 5.8. Similarly, the variance (self-covariance) of each attribute can be shown to be 6.2 by scaling the self-dot products of the aforementioned vectors by 10. The covariance matrix of this data set is therefore the following:

$$C = \begin{bmatrix} 6.2 & 5.8 \\ 5.8 & 6.2 \end{bmatrix}$$

The two eigenvectors of this covariance matrix are $\vec{e}_1 = [1/\sqrt{2}, 1/\sqrt{2}]^T$ and $\vec{e}_2 = [1/\sqrt{2}, -1/\sqrt{2}]^T$. The corresponding eigenvalues are $\lambda_1 = 12$ and $\lambda_2 = 0.4$, which can be verified using the condition $A\vec{e}_i = \lambda_i\vec{e}_i$ for each eigenvector \vec{e}_i . Most of the

points are primarily aligned along the \vec{e}_1 . The coordinates of the points using the eigenvector axis system $P = [\vec{e}_1, \vec{e}_2]$ can be obtained by representing the data as a 10×2 matrix and computing DP . These coordinates are as follows:

$$\{(4\sqrt{2}, 0), (3\sqrt{2}, 0), (2\sqrt{2}, 0), (\sqrt{2}, 0), (-\sqrt{2}, 0), \\ (-2\sqrt{2}, 0), (-3\sqrt{2}, 0), (-4\sqrt{2}, 0), (0, \sqrt{2}), (0, -\sqrt{2})\}$$

The variance along \vec{e}_1 is 12, which is the first eigenvalue. The variance along the second eigenvector \vec{e}_2 is 0.4, which is the second eigenvalue. The total variance is 12.4. The covariances in this new representation are 0. Since the first eigenvector has the higher variance, it can be retained in a 1-dimensional representation. The corresponding 1-dimensional coordinates are as follows:

$$\{4\sqrt{2}, 3\sqrt{2}, 2\sqrt{2}, \sqrt{2}, -\sqrt{2}, -2\sqrt{2}, -3\sqrt{2}, -4\sqrt{2}, 0, 0\}$$

Note that the fraction of the variance retained is $12/12.4 \approx 0.97$. ■

Problem 2.9 Consider the 2-dimensional data set showing the distribution of the points in the scatter plot of Figure 1.2 of Chapter 1. Make an approximate estimation of the vector representing the first principal component direction by placing a straight edge across the figure.

Problem 2.10 Consider a covariance matrix in which all non-diagonal elements are 0s. What will be the eigenvectors and eigenvalues of this covariance matrix? Make a guess on the eigenvectors and eigenvalues (by using the properties of principal components discussed above) if you do not know much linear algebra. Provide an explanation of why you obtain these eigenvectors in terms of one of the key properties of the principal components.

2.2.2.4 Contingency Tables for Categorical Data

All the forms of summarization described in the chapter thus far are designed for numeric data. A natural question arises as to how one can address categorical data or a mixture of categorical and numeric data. Categorical data are best summarized with the use of *contingency tables*. For two categorical attributes containing p and q possible categorical values, a $p \times q$ table is created, where the (i, j) th entry of the table is the frequency of the combination of the i th value of the first attribute and the j th value of the second attribute. Imagine a company that manufactures umbrellas in three different colors and sells them in two different countries. Each observation contains the country of sale as an attribute along with the color of the umbrella. The retailer wants to know if there are differences in the sales patterns of the different colors in different countries. The contingency table provides a natural way to summarize the frequency of sales of different colors and countries. In such a case, one can create a 2×3 contingency table containing the frequency of sales of the different umbrellas as shown in Table 2.1. Additional rows and columns are added to the table showing the totals across colors and countries. Contingency tables can also be written in terms of relative frequencies in either row-wise or column-wise form. The row-wise form of relative frequencies divides the table entries with the row sum, so that each row sums to 1. The column-wise form of relative frequencies divides the table entries with the column sum, so that each column sums to 1. Such fractional values can be viewed as statistical

Table 2.1: Contingency table of umbrella sale frequency (thousands) across countries and colors

	Black	Blue	Red	Totals
Country A	134	82	105	321
Country B	75	102	33	210
Totals	209	184	138	531

estimates of *conditional probabilities* of different categorical attribute values with respect to one another.

Is there a way of finding out whether the two attributes are associated with one another in a *statistically significant* manner? One can quantify the level of association between categorical attributes with the use of a measure referred to as the χ^2 -statistic. A detailed discussion of the use of the χ^2 -statistic to quantify “statistical association” between pairs of categorical attributes is provided in section 5.6.3 of Chapter 5.

Problem 2.11 Generate a relative-frequency version of the contingency table of Table 2.1 in which each row sums to 100 percent. Generate a relative-frequency version of the contingency table of Table 2.1 in which each column sums to 100 percent.

2.3 Data Visualization

Data visualization is often a key step that is used by analysts before applying more detailed machine learning techniques to data. There are numerous types of visualizations that are leveraged for univariate data and for multivariate data. This chapter will provide an overview of some of the most important methods for both types of visualizations.

2.3.1 Univariate Visualization

Univariate visualization methods provide a visual illustration of a single attribute of the data at one time. In this section, two important forms of univariate data visualization, which are the histogram and the box plot, will be introduced.

2.3.1.1 Histogram

The simplest univariate form of a data distribution is that of a histogram. A histogram divides the range between the minimum and maximum values of the attribute into b intervals, which are referred to as bins. The value of b is a parameter, which is selected by the user. The frequency of the number of points in each bin is computed and then plotted on the Y -axis, whereas the X -axis contains the bin ranges of the variable of interest. Examples of histograms of a data set containing the salaries of 200 individuals are shown in Figure 2.2 for varying numbers of bins. It is evident from the figure that using 5 bins does not provide sufficient details of the frequency variations across the range of salaries, whereas using 100 bins is excessive because only a few individuals fall in each bin on average (resulting in many empty bins or bins with noisy and irrelevant details). What we want is a histogram containing sufficient detail about the broad trends in the data distribution without its shape being too sensitive to the specific sample of the population that the analyst might have collected. In this particular case, the use of 20 bins turns out to be optimal as it provides an excellent overview of the broad trends in the data. In general, choosing the correct number of bins is

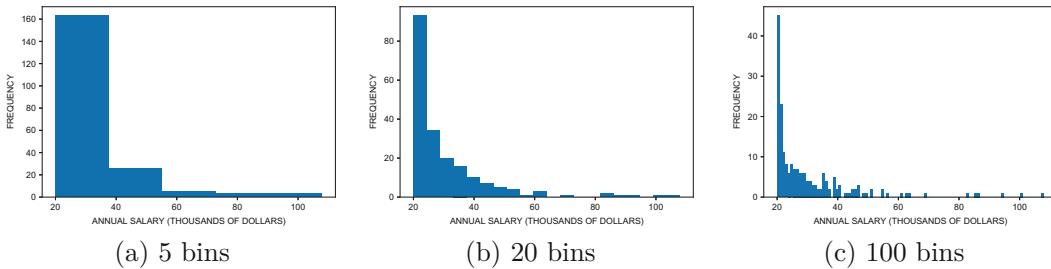


Figure 2.2: Histograms of the same data set with 200 points and varying numbers of bins

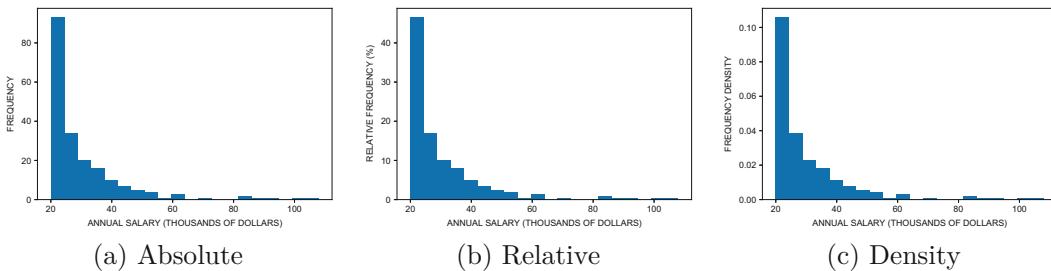


Figure 2.3: Histograms of different types for the same data set with 200 points and 20 bins

often achieved by repeatedly plotting the data with different values of b and selecting the most informative option based on visual estimation. A good rule of thumb is to choose a number of bins that is less than the number of points by a factor of 10.

It is noteworthy that all histograms shown in Figure 2.2 are *absolute frequency histograms*, because the scale of the Y -axis contains the absolute number of samples. There are two other common approaches for illustrating histograms, which are shown in Figure 2.3 along with the absolute frequency histogram (cf. Figure 2.3(a)) that has already been introduced in Figure 2.2. The *relative frequency histogram* is shown in Figure 2.3(b), in which the Y -axis is scaled so that the bars add up to 100 percent. In other words, the value of each bar corresponds to the *percentage* of items falling in each bin and the scale of the Y -axis provides a better understanding of the percentage of all the items that lie in each range. Finally, the most mathematically rigorous (but least intuitive) representation of the histogram is the density plot, in which the area of the histogram sums to 1. This histogram is shown in Figure 2.3(c). The significance of the area summing to 1 is that each bar represents a statistical estimate of the *probability distribution* from which the data was drawn. In this case, the density is sensitive even to the units in which the X -axis is represented. For example, the X -axis of Figure 2.3(c) represents the salary in thousands of dollars. If the salary were to be represented in dollars rather than thousands of dollars, all densities on the Y -axis would get divided by a factor of 1000. *A density histogram is a statistical estimate of the probability density of the population using a sample of the data.* The notion of probability densities has been introduced briefly in Chapter 1, and it will be explored in greater detail in the next chapter.

The histogram provides an understanding of the overall shape of the data distribution along with the type of *skew* in the data. A distribution is *right-skewed* if the extreme points of the distribution occur towards the right and most of the data lumps to the left

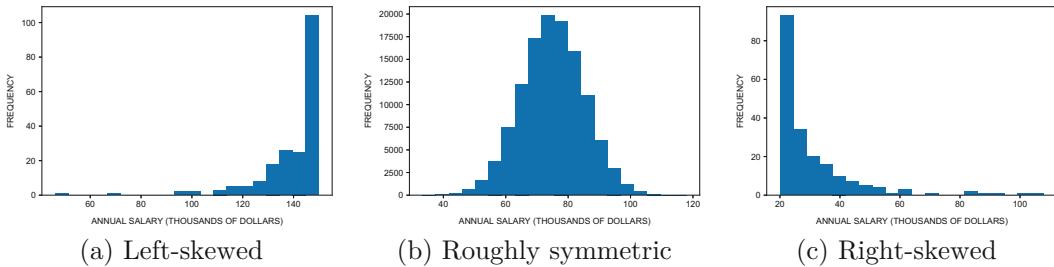


Figure 2.4: Histogram representations of left-skewed, right-skewed, and symmetric distributions

of the distribution. In such a case, the few extreme points that occur to the right of the distribution are referred to as its *tail*. A distribution is *left-skewed* if the extreme points of the distribution occur towards the left and most of the data lumps to the right. In this case, the tail of the distribution occurs to the left. Examples of left-skewed, symmetric, and right-skewed distributions are shown in Figures 2.4(a), (b), and (c), respectively. Skewed distributions are common in real-world data, because outliers tend to occur at one end of the distribution in many cases. As an example, if the frequency distribution of the net worth of the US population were to be represented by a histogram, the majority of the US population would occur in bins with ranges that are subsets of $[0, 100000]$. Less than 5% of the population would lie in bins corresponding to a net worth over a million dollars and a handful of people would lie in bins corresponding to a net worth over 10 billion dollars.

One heuristic way of identifying the nature of the skewness on the distributions (without explicitly visualizing the data) is to examine the relative effects on the mean and the median by the type of skew. The mean of a right-skewed distribution gets pulled to the right (relative to the median) by the few extreme points on the right, whereas the mean of a left-skewed distribution gets pulled to the left (relative to the median) by the few extreme points on the left. Therefore, the mean of a right-skewed distribution is typically greater than the median, and the mean of a left-skewed distribution is typically less than the median. In the case of symmetric distributions, the mean and median are situated at roughly similar places. This approach is only a heuristic, because small differences between the mean and the median may not necessarily provide a clear indicator of the nature of the skew in the distribution. As we will see in Chapter 4, there is a concrete way of identifying the skewness of distributions in a more reliable manner.

Finally, histograms can be used to identify unimodal, bimodal, or multimodal distributions. A histogram is unimodal, bimodal, or multimodal, depending on whether it has one peak, two peaks, or more than two peaks in the underlying histogram. Here, it is important to note that the number of peaks in a histogram is sensitive to the number of bins, and therefore one has to be careful while identifying the number of true peaks in the distribution without getting misled by noisy artifacts, which might appear as peaks in a specific data set. Furthermore, multiple peaks might sometimes coalesce into a single peak, if the number of bins selected is too small. The peaks may not be of exactly equal size, and therefore the notion of multimodal distributions is more of an intuitive definition while dealing with data-driven statistics (rather than one that is precisely defined). Examples of unimodal, bimodal, and multimodal distributions are shown in Figure 2.5. Although these data sets were generated from actual probability distributions with one, two, and three peaks, respectively, noisy data-specific variations result in all sorts of spurious peaks that are visible

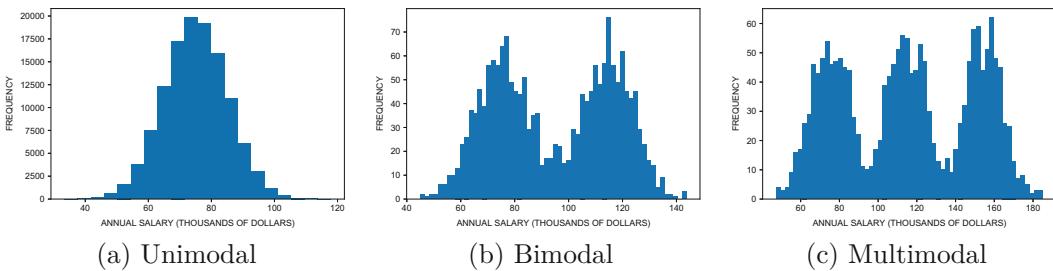


Figure 2.5: Histogram representations of data generated from unimodal, bimodal, and multimodal distributions

in Figure 2.5. This is an example of how judgements of modality often turn out to be a heuristic decisions with the notion of histograms. It is possible to smooth out some of these artifacts with visualization methods such as *kernel density estimation* [57]. Kernel density estimation methods are discussed in section 6.5 of Chapter 6.

Problem 2.12 *Provide an example of how the careless placement of bins can cause misleading results if the data is numeric but takes on a few discrete values. [This is one of the reasons that discrete numeric data are often treated differently from continuous numeric data by histogram-plotting software.]*

2.3.1.2 Box Plot

A second useful form of visualization is the *box plot*, and it is sometimes also referred to as the box-and-whisker plot. A box-and-whisker plot is not as effective at identifying the key details of a data distribution (as the histogram is), but it succinctly summarizes various quartiles and extreme points. It can be viewed as a representation of the extent of variation on both sides of the median, but it gives much less detail than a histogram does. It is, nevertheless, extremely popular with analysts because of its simplicity and the ability to merge it with other types of plots, when the quantity on the Y-axis has a randomized element to it. For example, multiple runs of the same machine learning algorithms may sometimes lead to different accuracy values depending on the choice of random seed or chosen data subset. A box plot is sometimes used as a more informative marker in straightforward algorithmic performance charts showing the accuracy of a particular machine learning algorithm against an algorithm parameter (with variations in accuracy over multiple independent runs occurring purely as a result of different choices of random seeds).

In a box plot, the statistics of a univariate data set are summarized in terms of five quantities representing different values along the Y-axis. As the name suggests, a box-plot uses a box, two whiskers, and a horizontal line in the interior of the box to represent these quantities. The five key quantities in the box-plot are the “minimum/maximum values” (whisker locations), the upper and lower quartiles (the location of the upper and lower ends of the box), and the median (line in middle of box). We have enclosed quotations around two of these quantities because they are defined in a non-standard way. The distance between the upper and lower quartiles is the inter-quartile range (IQR) introduced earlier in this chapter. The “minimum” and “maximum” are defined in a (non-standard) trimmed way in order to define the location of the whiskers. If there are no points more than 1.5 IQR above the top quartile value (upper end of the box), then the upper whisker is the true maximum. Otherwise, the upper whisker is set at 1.5 times the IQR from the upper end of the box.

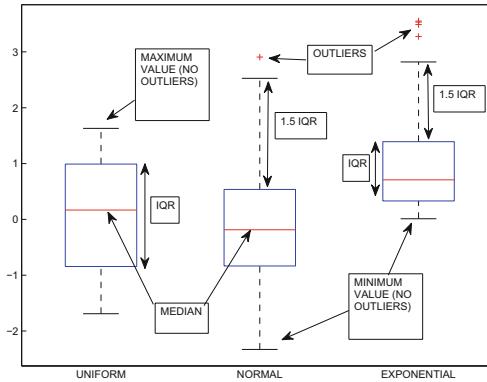


Figure 2.6: Visualizing univariate data with box plots

An exactly analogous rule holds true for the lower whisker, which is set at 1.5 IQR from the lower end of the box, unless the minimum is less than 1.5 IQR from the bottom of the box. Furthermore, any extreme points that do not lie between the upper and lower whiskers are shown explicitly with markers. These are the outliers in the data set, and they are also referred to as *fliers* in the box plot.

An example of a box plot is illustrated in Figure 2.6. In this case, we have shown 100 data points, which were generated by repeatedly sampling points from (i) a uniform distribution with zero mean and unit variance, (ii) a standard normal distribution with zero mean and unit variance, and (iii) an exponential distribution with unit mean. The last of these distributions has not yet been introduced — the exponential distribution has non-zero probability density only over non-negative values of the variable x , and decays exponentially at a rate equal to $\lambda \exp(-\lambda x)$ based on a decay-parameter $\lambda > 0$, causing a right-skewed distribution (like Figure 2.4(c)). Note that the first two distributions are symmetric about the mean, whereas the last distribution is not. The choice of these distributions shows the effect of different underlying data distributions on the box plot. In each case, the upper and lower ends of the box represent the upper and lower quartiles. In the case of the uniform distribution, no outliers were generated because of the nature of the distribution, and therefore, the upper and lower whiskers represent the true maximum and minimum values of the 100 generated data points. On the other hand, there are outliers at the upper end of the box plot in the case of the normal and exponential distributions. Therefore, the whiskers are placed at 1.5 IQR above the upper ends of the boxes in each of the cases. Another interesting observation is the asymmetric nature of the placement of the quartiles and whiskers for a right-skewed distribution. Therefore, the box plot retains information about the skewness of the underlying distribution, although this information is not as detailed as in the case of a histogram.

Many other conventions exist on the placement of whiskers, such as the use of the actual minimum/maximum or the use of particular percentiles of the data distribution. The specific convention used in this book is referred to as the *Tukey box-plot*. Aside from visualizing extreme values, this type of diagram is useful for visualizing the performance of many types of randomized algorithm with various algorithm parameters. This is because multiple runs of the same algorithm and same parameter values often show different performance because of the randomized nature of the underlying algorithm. A box plot is able to show the details of these variations.

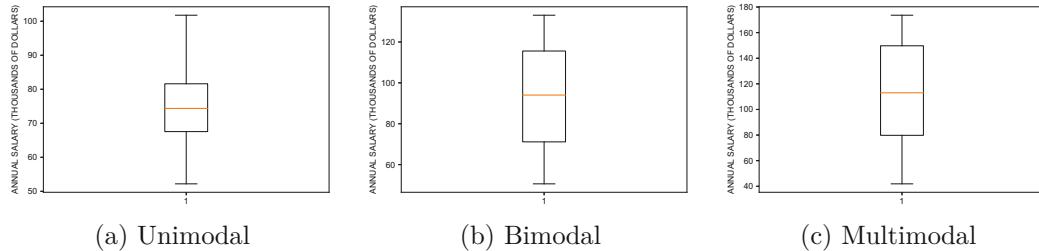


Figure 2.7: Box plots of data generated from unimodal, bimodal, and multimodal distributions. The same distributions are used as Figure 2.5 for generation, except that fewer points are used to avoid outlier clutter. There are virtually no indications of the modality of the different distributions in the corresponding plots.

A box plot carries significantly less information as compared to a histogram. To illustrate this point, we generate the box-plots of the same types of unimodal, bimodal, and multimodal distributions as shown in Figure 2.5. These box-plots are illustrated in Figure 2.7. It becomes immediately evident that it is hard to distinguish among the different shapes of distributions with the use of a box plot. This is a direct consequence of the fact that a box plot encodes less information than a histogram. The box-plot opens the door to a detailed form of multivariate analysis in which the effect of the changes in one variable on the mean value and dispersion of the other variable is exposed. For example, consider the case where both the age and the salary of individuals are available. In such a case, one can construct different box-plots over different age ranges in order to show the effect of variation of age with salary. An example of such a box plot is shown in Figure 2.8(a). It is evident that the average salary increases with age but the IQR of the salary also increases with age. In other words, there is greater variability in salary for older people. In fact, the difference between the 50–60 age group and the 60–70 age group is primarily one of dispersion. This might be the case because² older individuals can often be in senior positions but often have greater difficulty at getting new jobs because of discrimination. It is much harder to make such detailed interpretations of multi-attribute trends when markers are used instead of box-plots. A second (synthetic) toy example is shown for classifier accuracy performance with respect to a (randomized) classifier algorithm parameter p in the range $(0, 1)$. It is assumed that for any particular choice of the algorithm parameter, the randomized classifier algorithm is executed 500 times, and a box-plot of accuracies is created from the different performances. This approach is repeated for 9 different parameter choices and the results are shown in Figure 2.8(b). It is evident that the most accurate median performances of the algorithm are achieved at $p = 0.3$ and $p = 0.7$. However, it makes more sense to use $p = 0.7$ because the algorithm shows stable performance. Again, it is the additional dispersion information in the box plots that yield these types of useful insights. The simplicity of box plots enables their combination with other types of plots. Even though histograms carry more detailed distribution information than box plots, they are harder to combine with other plots for the purpose of visualization.

²The data in both parts of Figure 2.8 are synthetic and purely illustrative in nature.

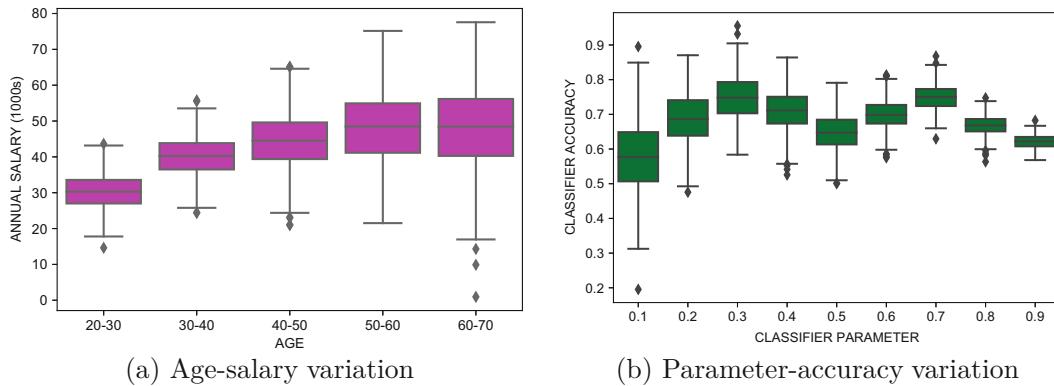


Figure 2.8: The importance of box-plots in showing detailed variable relationships

Example 2.14 Is it possible for a Tukey box plot to have no box (i.e., collapsed box with thickness 0) but to have clear whiskers at some distance on either side of the collapsed box? If not, explain why not. If so, provide an example.

Solution: It is not possible for such a situation to occur because the whiskers are placed at distances of *at most* 1.5 IQR from box ends. In this case, the collapsed box indicates that the IQR is 0. ■

Problem 2.13 In what type of data set would the lower whisker of the box plot align with the lower end of the box in the box plot? Give an example.

A hint for solving this problem is to examine cases in which the lower quartile of a data set is the same as the minimum value.

2.3.2 Multivariate Visualization

In multivariate data, the data has multiple attributes, and one would often like to know how the different attributes are associated with one another. This section will discuss the three most popular visualization techniques, which correspond to the *line plot*, the *scatter plot* and the *bar chart*.

2.3.2.1 Line Plot

The line plot is generally used when one of the variables is an *independent variable* (which is directly controlled in an experiment) and the other variable is a *dependent variable* (which is a numeric quantity representing the outcome of the experiment). The independent variable is represented on the *X*-axis, whereas the dependent variable is represented on the *Y*-axis. The line plot essentially draws a line through the different (x, y) coordinates that are collected in the experiment. In the event that these coordinates do not lie on a straight line, a piecewise linear line is drawn through the points. In many cases, these piecewise linear lines do not represent satisfactory trends because they do not account for the random variations and natural errors that arise in data collection. As you will learn in the chapter on regression (cf. Chapter 7), it is also possible to draw a (straight) line of best fit, which

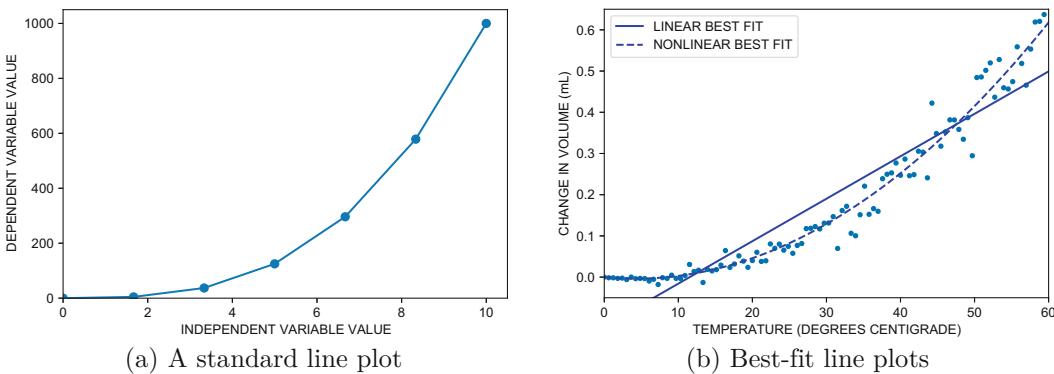


Figure 2.9: Examples of line plots of different types

does nor necessarily pass through all the points. Such a line is preferable when the collected data is known to have experimental errors or other random variations.

In some cases, the data may not naturally follow a linear trend. In such cases, it is possible to draw a nonlinear curve of best fit, which typically does a better job of passing through all the points but it is not a straight line — instead it is a smooth curve representing the typical trends in the data. An example of a standard (piece-wise linear) line plot is illustrated in Figure 2.9(a). This type of line can be constructed with the use of nonlinear regression techniques. Both linear and nonlinear line plots (corresponding to best-fit lines) are illustrated in Figure 2.9(b). In this case, the results of a laboratory experiment that measures the effect of temperature on the volume of water are illustrated. The temperature is shown on the X -axis, whereas the change in volume of water is shown on the Y -axis. In general, the volume of water is known to decrease from 0°C to 4°C , and then increase in a non-linear way. These types of known relationships in science are originally discovered using precisely this type of experiment that repeatedly measures temperature and change in volume (by repeatedly heating and cooling water). The different dots in Figure 2.9(b) illustrate the results of different experiments. There is significant variation across different experiments because of human error in measurement or other confounding factors. One can use these data to generate a line of best fit. The linear and nonlinear best-fit lines show different types of relationships between the temperature and volume of water in Figure 2.9(b). In this case, it is evident that the nonlinear curve fits the trends in the data much better. Methods for finding such best-fit lines are discussed in Chapter 7.

2.3.2.2 Scatter Plot

The simplest visualization is the *scatter plot*, which plots two of the features of the data with one another and therefore provides an idea of how the features vary with respect to each other. Each of the axes corresponds to one of the features. It is possible for the X -axis to represent the independent variable and the Y -axis to represent the dependent variable (as in the case of the line plot), although it is not necessary for the variables to be classified as independent or dependent in the case of the line plot. Each observation in the data is denoted with a marker. The scatter plot is a very simple form of visualization that exposes important patterns in the data. Examples of scatter plots are illustrated in Figure 2.10. The scatter plot of Figure 2.10(a) is an age-salary scatter-plot, which shows a positive correlation between the age and the salary. The scatter plot of Figure 2.10(b) shows the latitude and

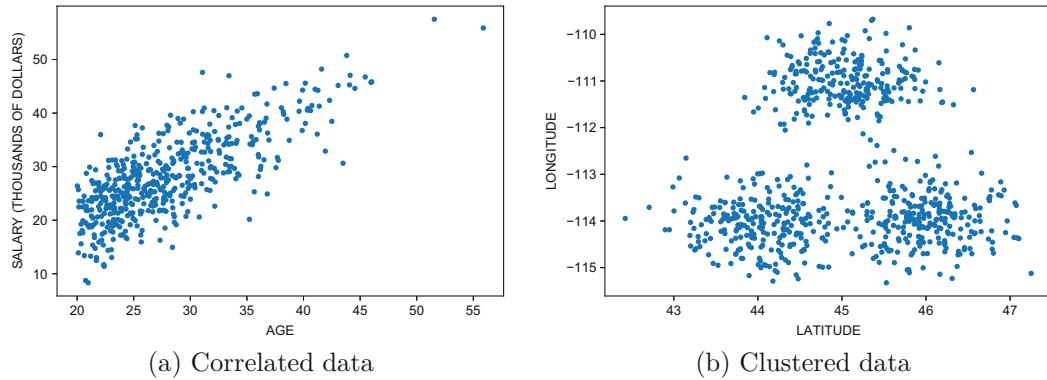


Figure 2.10: Examples of scatter plots of different data patterns

longitude of mobile phone users, and it tends to be naturally clustered in crowded regions. Both these types of patterns make a lot of sense for the data set at hand. Each scatter plot tells its own story that is specific to the scenario that generated the data (i.e., the generating mechanism).

2.3.2.3 Bar Chart

All visualization techniques that have been discussed thus far are designed for numerical data. The preponderance of numeric data for visualization is not particularly surprising because of the ability to intuitively represent numeric data spatially. Nevertheless, a number of techniques have also been developed for both multivariate categorical data and for mixed-attribute data. First, we discuss the visual representation of the association between a numerical attribute and one or more categorical attributes with the use of *bar charts* or *bar plots*. A bar plot first computes a summary statistic of a numeric attribute (typically the mean value but can also be another statistic including one involving dispersion-centric analysis) for data points corresponding to each value of the categorical attribute. Then, it plots this summary statistic of the numeric attribute for each value of the categorical attribute with a separate bar. Therefore, a bar plot always presents the statistic of a numeric attribute with respect to one or more categorical attributes. The specific statistic of the numeric attribute that is presented is referred to as the *estimator*, and it is typically the mean of the numeric attribute, but can chosen to be another value such as the sum or the standard deviation. It is noteworthy that even though this framework seems to be designed only for numeric-categorical pairs, it can easily be adapted to purely categorical data. This adaptation will be discussed later.

Consider the Adult data set from the UCI Machine Learning Repository [66], which contains several numerical and categorical demographic attributes, such as age, years of education, race, and salary-level. The age and years of education are obviously numeric attributes. Race is a categorical attribute. While salary level would normally be numeric, it turns out to be categorical in this particular data set because of how it has been preprocessed — the attribute has only two possible values, corresponding to people making less than 50,000 dollars and those making more than 50,000 dollars. Therefore, it is a binary attribute that should be treated as a categorical value. Since the data set was collected to construct a model of the salary level of individuals from demographic/educational attributes (such as age, race, and years of education), it is useful to visualize the variations in these attributes

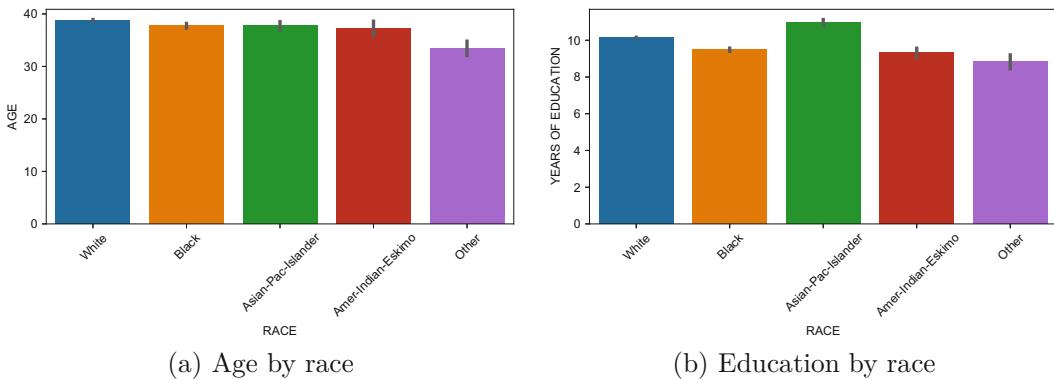


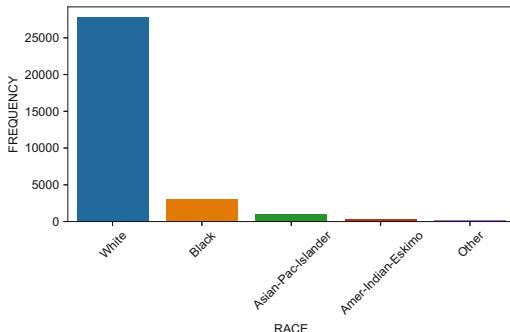
Figure 2.11: Simple bar plot showing age and years of education by race for the Adult data set in the UCI Machine Learning Repository

across different races to see how much of the salary differences can be explained by these attributes.

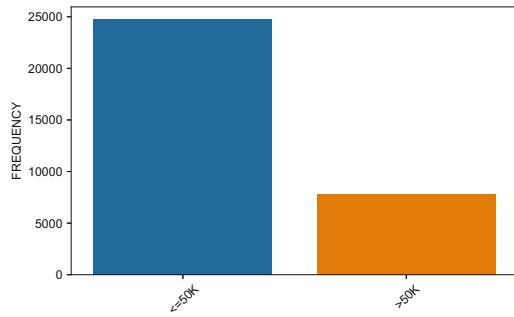
The bar plot in Figure 2.11(a) shows the average age of people in the Adult data set by race. Note that a small vertical line appears on the top of each bar, which is an *error bar*, the length of which corresponds to the standard deviation of the numeric attribute within that group. This type of error bar is particularly useful when the mean of that group is analyzed and presented as the primary statistic. Note that the ages of the different respondents are roughly similar with small variations. For example, Black respondents tend to be slightly younger than White respondents in this particular data set, whereas Asians tend to be slightly older. A similar bar chart of the variations in the average number of years of education across different races is shown in Figure 2.11(b), in which there are small differences across races in this data sample.

The aforementioned exposition presents the statistic of a numeric attribute with respect to categorical values. What if we wanted an analysis of the summary statistics of the categorical values (such as the frequency)? This case is similar to that of constructing a histogram of frequencies except that the base attribute is now categorical instead of numeric — recall that histograms are constructed only over ranges of numeric attributes. In such a case, one can change the estimator of the bar plot to count the number of occurrences of each value of the categorical attribute. This type of statistic can be computed by creating a dummy numeric attribute of ones, and then using the sum estimator with respect to the categorical attribute of interest. Using this approach on the Adult data set for the race and salary level yields the raw frequencies of the different races and salary levels, as shown in Figure 2.12.

The aforementioned bar chart is a *simple bar chart*, because it shows the variation of a numeric attribute across only one categorical attribute. By using the concept of grouping, it is possible to show the variation of a numeric attribute across multiple categorical attributes (which is typically a pair of categorical attributes). In a *grouped bar chart*, the numerical attribute statistics (e.g., mean of age or salary) are computed for each combination of a pair of categorical attribute values and the statistics are shown by nesting one of the categorical attributes within the other as a grouped family. For example, Figure 2.13(a) shows the variation in age across different races and salary levels, while using the salary-level attribute for the inner-level nesting of the categorical grouping. A similar type of

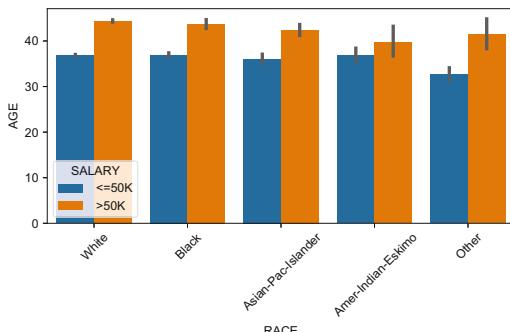


(a) Frequency by race

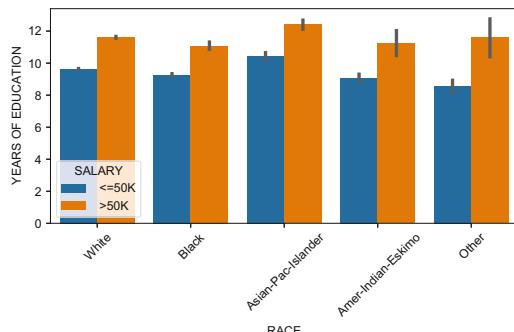


(b) Frequency by salary level

Figure 2.12: Simple bar plot: Showing the frequencies of different races and salary levels for the Adult data set in the UCI Machine Learning Repository.



(a) Age by salary/race



(b) Education by salary/race

Figure 2.13: Grouped bar plot showing age and years of education by race over different salary levels for the Adult data set in the UCI Machine Learning Repository

nested bar chart in shown in Figure 2.13(b), except that the primary numeric attribute analyzed in this case is the number of years of education. Both figures show that the salary level is sensitive to the age and the number of years of education. Furthermore, it can be inferred that within each race, a higher salary level implies greater age and number of years of education. One limitation of the data set is that since the salary level is provided only as a binary value, it is impossible to analyze how the salary level varies across races while fixing the age and the number of years of education. Nevertheless, the bar charts do provide excellent insights into the complex relationships among different categorical and numeric attributes. One can also use the grouped bar chart in order to present the frequencies of pairs of categorical attributes. As in the case of the generation of frequencies of numeric attributes, a dummy numeric attribute of ones can be created, and then the sum estimator is used on the dummy attribute with respect to the two nested categorical attributes. On applying this approach on the race and salary-level attributes the bar chart of Figure 2.14 is obtained. In this case, the frequencies of the different race- and salary-level combinations are presented in the chart. One can view this chart as a pictorial representation of the contingency table discussed earlier in this chapter. One problem with this chart is that some of the frequency bars are too small to be properly visible. In such cases, it makes

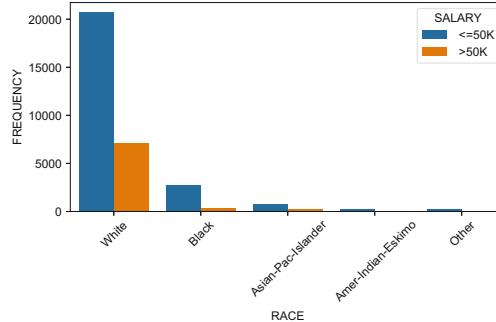


Figure 2.14: Grouped bar plot showing the frequencies of different races and salary level combinations for the Adult data set in the UCI Machine Learning Repository.

sense to construct relative frequency charts. For example, the *percent* or *fraction* of people of that specific race making more or less than 50,000 can be presented in each case. Such an approach allows a more direct comparison across the different races in terms of their salary levels. To achieve this goal, the dummy attribute value can be modified in order to make it sensitive to the attribute over which the relative frequency normalization is done. Since the sum of the two bars over each race must equal 100, the value of the dummy attribute for an individual belonging to the i th race is set to $100/f_i$, where f_i is the number of respondents belonging to the i th race in the data set. Then, using the sum estimator with respect to the two nested categorical attributes will yield a relative-frequency bar plot.

2.4 Applications to Data Preprocessing

The summarization techniques discussed in this chapter are often used for data preprocessing. Data preprocessing is a critical step used in machine learning applications because it critically affects the performance of various machine learning algorithms. In other words, the nature of the preprocessing can affect both the accuracy and the speed of execution of a particular machine learning algorithm.

2.4.1 Univariate Preprocessing Methods

In univariate preprocessing methods, the features are processed independently from one another. These methods use various types of scaling and translation as follows:

1. *Standardization*: A common type of normalization is to subtract the feature mean from each feature value and then divide each resulting value by the feature standard deviation. If x_{ij} is the value of the j th feature for the i th observation, the standardization process uses the estimated mean $\hat{\mu}_j$ and standard deviation $\hat{\sigma}_j$ of attribute j in order to compute the normalized feature value as follows:

$$x_{ij} \leftarrow \frac{x_{ij} - \hat{\mu}_j}{\hat{\sigma}_j} \quad (2.2)$$

The basic idea is that each feature is presumed to have been drawn from a *standard* normal distribution with zero mean and unit variance (which is why this form of feature preprocessing is referred to as standardization). For unimodal and symmetric

distributions, most of the resulting values of the features will lie in the range $(-3, 3)$, although a few extreme values may lie outside this range.

2. *Min-max normalization:* This type of feature normalization maps all feature values to the range $(0, 1)$. Let \min_j and \max_j be the minimum and maximum values of the j th attribute. Then, each feature value x_{ij} is scaled by min-max normalization as follows:

$$x_{ij} \Leftarrow \frac{x_{ij} - \min_j}{\max_j - \min_j} \quad (2.3)$$

The main advantage of min-max normalization is that features are mapped to the specific range $(0, 1)$. This can be useful in some applications where nonnegativity or boundedness is assumed. However, standardization is a more principled statistical approach.

Feature normalization methods often enable better performance of machine learning algorithms both in terms of accuracy and running times. If the different features are on different scales that vary by orders of magnitude, the feature with large variability will dominate in terms of predictive influence. For example, the Euclidean distances between a pair of points will be dominated by the attribute on a larger scale. As a result, the knowledge available in other features may be de-emphasized. Another problem with unnormalized features is that computational algorithms like gradient descent converge slowly [6].

Example 2.15 Consider a data set with a few extreme values on some features. Discuss what kind of problems might arise in Euclidean distance-based algorithms on using min-max normalization on such a data set. Would this problem arise in standardization to the same extent?

Solution: A few extreme values can cause a lack of discrimination along a subset of the features where the extreme values exist. Such features will be de-emphasized when using Euclidean distances. This is because the scale factor along each dimension in min-max normalization is decided exclusively by extreme values (which could easily be errors in recording the data). This problem does not arise to the same extent in standardization in which the scale factor is decided by all values.

From a probabilistic point of view, standardization assumes that each feature is generated from the normal distribution introduced in Chapter 1 (where extreme values occasionally occur) and the squared Euclidean distance on standardized data turns out to have the same form as the exponent of the normal distribution from which the data is generated. For example, the density function of the i th data point $\vec{x}_i = [x_{i1}, \dots, x_{id}]$ is assumed to be the following:

$$f(x_{i1}, x_{i2}, \dots, x_{id}) \propto \exp \left(-\frac{1}{2} \underbrace{\sum_{j=1}^d \left[\frac{x_{ij} - \mu_j}{\sigma_j} \right]^2}_{\text{distance}} \right)$$

Therefore, Euclidean distance-based algorithms effectively work with the more principled perspective of log-probabilities, when standardization is used. ■

2.4.2 Whitening: A Multivariate Preprocessing Method

Another form of feature pre-processing is referred to as *whitening*, which is based on principal component analysis. Principal component analysis has already been discussed on page 42. This section discussed further details in the context of whitening. In whitening, the axis-system for data representation is rotreflected using principal component analysis to create a new set of *de-correlated* features; the new data values along each decorrelated feature are then scaled to unit variance. Univariate forms of normalization do not decorrelate the data, because they process individual features at a time without regard to the other features. On the other hand, whitening methods create decorrelated features that are linear combinations of the original features. An important point is that very strong correlations among features can cause the same types of problems associated with the varying scales of features — correlated features may tend to get overemphasized in algorithmic computations and may also cause ill-conditioning. Therefore, it is important for feature normalization methods to account for the underlying correlations.

Let D be an $n \times d$ data matrix *that has already been mean-centered*. Let C be the $d \times d$ co-variance matrix of D in which the (i, j) th entry is the co-variance between the dimensions i and j . Because the matrix D is mean-centered, we have the following:

$$C = \frac{D^T D}{n} \propto D^T D \quad (2.4)$$

Note that the relationship above holds only for mean centered data, because the covariance between a pair of attributes is the mean of the pairwise products of (centered) attributes. Since $D^T D$ is a $d \times d$ matrix whose (i, j) th entry contains the sum of the product of the i th and j th attribute (across observations), dividing each entry of this matrix by n results in a covariance matrix. The eigenvectors of this matrix are the principal components, corresponding to the de-correlated directions in the data (with respect to the sample at hand).

Let P_k be a $d \times k$ matrix in which each column contains one of the top- k eigenvectors. The eigenvectors of a positive semi-definite matrix (like the covariance matrix) are always orthonormal. Let Δ_k be a $k \times k$ diagonal matrix containing the corresponding nonnegative eigenvalues in the same order as the eigenvector columns of P_k . Then, the covariance matrix can be approximately represented by the following *eigendecomposition*, with exact equality guaranteed at $k = d$:

$$C \approx P_k \Delta_k P_k^T$$

Then, the data matrix D can be transformed into the k -dimensional axis system corresponding to the vectors in the columns of P_k . This is achieved by post-multiplying D with the matrix P_k . The resulting $n \times k$ matrix U_k , whose rows contain the transformed k -dimensional data points, is given by the following:

$$U_k = DP_k \quad (2.5)$$

The new data representation in the $n \times k$ matrix U_k has k columns just like the $d \times k$ eigenvector matrix P_k . The columns of the matrix P_k form the new *basis directions* (i.e., a rotreflected axis system) for the new data representation coordinates in U_k . The variances of the columns of U_k are the eigenvalues of the corresponding eigenvectors in P_k , because this is the property of the de-correlating transformation of principal component analysis. In whitening, each column of U_k is scaled to unit variance by dividing it with its standard deviation (i.e., the square-root of the corresponding eigenvalue). The transformed features can then be used by a variety of machine learning algorithms.

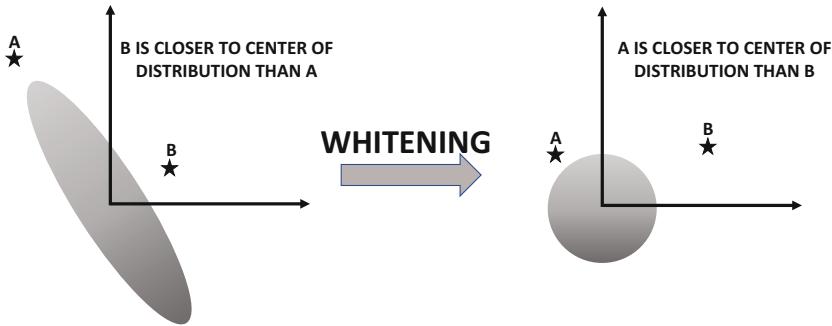


Figure 2.15: An example of how whitening can expose outliers in the data

Therefore, if one uses the top- k eigenvectors (i.e., largest k eigenvalues) of the covariance matrix, most of the variance in the data will be retained and the noise will be removed. One can also choose $k = d$, but this will often result in the variances along the near-zero eigenvectors being dominated by numerical errors in the computation. It is a bad idea to include dimensions in which the variance is caused by computational errors, because such dimensions will contain little useful information for learning application-specific knowledge. Furthermore, the whitening process will scale each transformed feature to unit variance, which will blow up the errors along these directions. At the very least, it is advisable to use some conservative threshold like 10^{-6} on the magnitude of the eigenvalues and remove such directions from the representation (assuming that the corresponding variances are caused primarily by numerical errors). Therefore, as a practical matter, the value of k will rarely be exactly equal to that of d . Alternatively, one can add a small constant $\lambda > 0$ to each scaling factor (square-root of eigenvalue) as a form of regularization.

An important aspect of whitening is that one might not want to make a pass through a large data set to estimate its covariance matrix. In such cases, the covariance matrix and column-wise means of the original data matrix can be estimated on a sample of the data. The $d \times k$ eigenvector matrix P_k is computed using this approximate covariance matrix. Subsequently, the following steps are used for transforming each data point into the new representation defined by P_k : (i) The mean of each column is subtracted from the corresponding feature; (ii) Each d -dimensional row vector representing a data point is post-multiplied with P_k to create a k -dimensional row vector; (iii) Each feature of this k -dimensional representation is divided by the square-root of the corresponding eigenvalue.

The basic idea behind whitening is that the data is assumed to be generated from an independent normal distribution along each principal component. By whitening, one assumes that each such distribution in the whitened data is a *standard* normal distribution, and therefore readjusts each variance to 1. After whitening, the scatter plot of the data will have a spherical shape, even if the original data is elliptically elongated with an arbitrary orientation. The idea is that the uncorrelated concepts in the data have now been scaled to equal importance (on an a priori basis). This makes the machine learning algorithm provide equal importance to the different features during the learning process. In order to understand this point, we provide an example from the problem of outlier detection in Figure 2.15. Consider an outlier detection algorithm that reports the distance of a point from the center of the data in order to identify whether a point is an outlier. A larger distance from the center of the data is more indicative of outlierness. According to this approach, the point A is more likely to be considered an outlier than point B (based on the original

data on the left of Figure 2.15). However, since the data distribution is elongated along the direction of point A, it is more likely that small variations from the original generating process can create point A as opposed to point B. The whitening process can resolve this problem. When whitening is used, the original data distribution now becomes spherical, and point B moves farther away from the center of the distribution. As a result, it is now more likely to be considered an outlier, which is the correct conclusion. In general, principal component analysis makes algorithms based on Euclidean distance much more sensitive to the aggregate distribution of the data. A detailed discussion of the aforementioned outlier detection algorithm is provided in section 9.4.1 of Chapter 9. Whitening has a variety of other benefits in machine learning. For example, algorithms that use *optimization techniques* like *gradient descent* are also much better behaved in terms of being able to converge to an optimal solution more rapidly when the data is preprocessed with whitening [6].

Example 2.16 Consider the mean-centered data set of Example 2.13:

$$\{(4, 4), (3, 3), (2, 2), (1, 1), (-1, -1), (-2, -2), (-3, -3), (-4, -4), (1, -1), (-1, 1)\}$$

Compute the whitened representation and comment on it.

Solution: Based on the discussion in the solution to Example 2.13, the decorrelated representation is as follows (with standard deviations of $\sqrt{12}$ and $\sqrt{0.4}$ along the two eigenvector dimensions):

$$\{(4\sqrt{2}, 0), (3\sqrt{2}, 0), (2\sqrt{2}, 0), (\sqrt{2}, 0), (-\sqrt{2}, 0), \\ (-2\sqrt{2}, 0), (-3\sqrt{2}, 0), (-4\sqrt{2}, 0), (0, \sqrt{2}), (0, -\sqrt{2})\}$$

On dividing by the standard deviations along the two dimensions, the following whitened representation is obtained:

$$\{(2\sqrt{6}/3, 0), (\sqrt{6}/2, 0), (\sqrt{6}/3, 0), (\sqrt{6}/6, 0), (-\sqrt{6}/6, 0), \\ (-\sqrt{6}/3, 0), (-\sqrt{6}/2, 0), (-2\sqrt{6}/3, 0), (0, \sqrt{5}), (0, -\sqrt{5})\}$$

The data representation has been contracted along the first dimension and stretched along the second dimension. Based on the Euclidean distance using the original representation, the last two points are the most inlier-like because they are closest to the data mean. However, on whitening, the changed Euclidean distance causes these points have the highest outlier scores. This change is because these points deviate along a direction of low variance, which causes them to be recognized as outlier-like in a probabilistic sense. ■

2.5 Summary

This chapter introduces a number of common summarization and visualization techniques in statistics and machine learning. Such methods are very useful for obtaining an understanding of the overall data distribution before designing machine learning algorithms. Summarization and visualization techniques can be designed for either univariate data or multivariate data. Univariate summarization methods include measures of central tendency and

measures of dispersion. Multivariate summarization techniques include methods like principal component analysis. Numerous visualization techniques such as histograms, box-plots, scatter plots, and bar-charts provide insights into the nature of data distributions. Univariate and multivariate summarization methods are often used for data preprocessing. A particularly common method for data preprocessing is that of whitening, which is achieved with the use of principal component analysis. Whitening methods are also used directly in machine learning algorithms like outlier detection.

2.6 Further Reading

A discussion of key summarization methods in statistics may be found in [38]. Principal component analysis and its relationship to machine learning is discussed in detail in [6]. A classical exposition of principal component analysis may be found in [41]. An excellent book on data visualization may be found in [60]. Variations of different types of box plots are discussed in [48]. One of the best softwares for data visualization is the Seaborn library written in Python [61]. Many of the plots constructed in this chapter are based on Seaborn.

2.7 Exercises

1. Compute the mean and the median of 1, 2, 3, 5, 7, 8, 9. You will find that the mean and the median are the same. Give an intuitive reason why the mean and the median are the same in this case.
2. Compute the mean, median, quartiles, and inter-quartile range of the data sample 1, 3, 3, 4, 7.
3. Compute the mode of the set of data points 1, 4, 4, 4, 5, 7, 9, 9, 9. Is this data set unimodal, bimodal, or multimodal?
4. Find the standard deviation of the points 3, 4, 5.
5. Compute the covariance and correlation between the ordered sets 5, 4, 3 and 2, 4, 6.
6. The pairs (x_i, y_i) satisfy $y_i = mx_i + b$. Answer the following:
 - (a) Show that the variance of $y_1 \dots y_n$ is m^2 times the variance of $x_1 \dots x_n$.
 - (b) How does the covariance of (x_i, y_i) relate to the variance of $x_1 \dots x_n$?
 - (c) Show that the correlation between $x_1 \dots x_n$ and $y_1 \dots y_n$ is either 1 or -1. When is the correlation positive and when is the correlation negative?
7. Download the Adult data set from the UCI machine learning repository [66]. Generate the contingency tables of race and salary-level attribute. Generate both absolute frequency and relative frequency versions of the contingency table.
8. Write a Python program to generate random values uniformly in $(0, 1)$ and create a histogram with 10 bins. Observe the changes in your histograms when you generate (i) 20 points (ii) 200 points, and (iii) 2000 points. Do the histograms become more or less representative of the underlying data distribution with increasing sample size?
9. Repeat Exercise 8 by generating data from a 1-dimensional normal distribution. [Those who are not yet fully familiar with normal distributions can skip this exercise.]

10. Create box-plots on the data of Exercises 8 and 9.
11. Download the Adult data set from the UCI Machine Learning Repository [66]. Generate a scatter plot of the age and the number of years of education attributes.
12. Consider a 1-dimensional data set, which has been standardized. Show that no more than one-ninth of the resulting values can have absolute magnitude greater than 3.
13. Compute the Kendall rank correlation between 2, 1, 3, 4, 5 and 5, 4, 1, 2, 3.
14. We define the modified variance and mean absolute deviation (MAD) of x_1, x_2, \dots, x_n about centrally located point a as follows:

$$\sigma^2(a) = \frac{\sum_{i=1}^n (x_i - a)^2}{n}$$

$$MAD(a) = \frac{\sum_{i=1}^n |(x_i - a)|}{n}$$

Show that the value of $\sigma^2(a)$ is minimized when a is the mean of the points in the data set. Show that $MAD(a)$ is minimized when a is the median of the points.

15. Let $x_1 \dots x_n$ and $y_1 \dots y_n$ be sets of samples with means $\hat{\mu}_X$ and $\hat{\mu}_Y$, respectively. Let the sample standard deviation of the composite data set be $\hat{\sigma}$. Show that the sample standard deviations and means satisfy the following relationship:

$$\hat{\sigma}^2 = \frac{\hat{\sigma}_X^2}{2} + \frac{\hat{\sigma}_Y^2}{2} + \left(\frac{\hat{\mu}_X - \hat{\mu}_Y}{2} \right)^2$$

Assume that n is large and you can ignore the Bessel correction.

16. Let $x_1 \dots x_n$ and $y_1 \dots y_n$ be sets of samples with means $\hat{\mu}_X$ and $\hat{\mu}_Y$, respectively, as in Exercise 15. Assume that n is large and you can ignore the Bessel correction. Use the result of Exercise 15 to argue that the standard deviation $\hat{\sigma}$ of the composite data set containing $x_1 \dots x_n$ and $y_1 \dots y_n$ is at least equal to $|\hat{\mu}_X - \hat{\mu}_Y|/2$. Give an example of a data set where this inequality is satisfied as a strict equality.
17. Argue why the distance between the sample median and the sample mean of a data set is at most equal to half the absolute distance between the sample means of the first half and second half of the data set. Now use the result of Exercise 16 to show the famous result [17, 30] that *the distance between the mean and median is at most equal to the standard deviation*. Strictly speaking, this result can be rigorously shown only for probability distributions rather than samples (i.e., populations or data sets of infinite size), and one has to ignore the Bessel correction to make it hold for samples. Also assume for simplicity that the data set has an even number of points.
18. A 1-dimensional data set of n points $x_1 \dots x_n$ is partitioned into k groups $\mathcal{G}_1 \dots \mathcal{G}_k$, so that the j th group has sample mean $\hat{\mu}_j$ and has $n_j = |\mathcal{G}_j|$ points.
 - (a) Show that $\sum_{j=1}^k \sum_{x_i \in \mathcal{G}_j} (x_i - \hat{\mu}_j) \hat{\mu}_j = 0$.
 - (b) Use the result of part (a) to show the following:

$$\sum_{i=1}^n x_i^2 = \sum_{j=1}^k \sum_{x_i \in \mathcal{G}_j} (x_i - \hat{\mu}_j)^2 + \sum_{j=1}^k n_j \hat{\mu}_j^2$$

Hint: Break up x_i on the left-hand-side into a sum of two carefully chosen algebraic terms.

- (c) Use the result of part (b) to show the following for any scalar a :

$$\sum_{i=1}^n (x_i - a)^2 = \sum_{j=1}^k \sum_{x_i \in G_j} (x_i - \hat{\mu}_j)^2 + \sum_{j=1}^k n_j (\hat{\mu}_j - a)^2$$

[**Note:** When $a = \hat{\mu} = \sum_i x_i/n$, the first term in the above result is popularly referred to as the total sum-of-squares (*TSS*). The second term is the within sum-of-squares (*WSS*), and the third term is the between sum-of-squares (*BSS*). A useful result in statistics is that $TSS = WSS + BSS$.]

- 19. [Average squared inter-point distance is twice the sample variance:]** The sample mean of a 1-dimensional data set $x_1 \dots x_n$ of n points is $\hat{\mu}_X$.

- (a) Show that the *aggregate* squared inter-point distance (LHS below) satisfies the following:

$$\sum_{i=1}^n \sum_{j=i+1}^n (x_i - x_j)^2 = (n-1) \sum_{i=1}^n (x_i - \hat{\mu}_X)^2 - 2 \sum_{i=1}^n \sum_{j=i+1}^n (x_i - \hat{\mu}_X)(x_j - \hat{\mu}_X)$$

- (b) Show the following:

$$\sum_{i=1}^n (x_i - \hat{\mu}_X)^2 + 2 \sum_{i=1}^n \sum_{j=i+1}^n (x_i - \hat{\mu}_X)(x_j - \hat{\mu}_X) = 0$$

- (c) Combine the results of part (a) and (b) to show that the *average* squared inter-point distance is equal to twice the sample variance:

$$\frac{\sum_{i=1}^n \sum_{j=i+1}^n (x_i - x_j)^2}{\binom{n}{2}} = 2 \underbrace{\frac{\sum_{i=1}^n (x_i - \hat{\mu}_X)^2}{n-1}}_{\hat{\sigma}_X^2}$$

- 20.** Consider the following 2-dimensional data set:

$$\{(5, -3), (4, -2), (3, -1), (2, 0), (0, 2), (-1, 3), (-2, 4), (-3, 5), (2, 2), (0, 0)\}$$

Compute the whitened representation of the data set. Use any Web calculator for finding eigenvectors.

- 21.** Let the pairwise disjoint sets $A_1 \dots A_k$ each contain n numbers with corresponding medians $m_1 \dots m_k$. Give a counterexample to show that the median M of m_1, \dots, m_k may not be the median of $\cup_{r=1}^k A_r$. Furthermore, show that M lies in the inter-quartile range of $\cup_{r=1}^k A_r$.

- 22.** Given the cardinalities, means, and variances of pairwise disjoint sets A_1, \dots, A_k of numbers, outline a procedure to compute these statistics for $\cup_{i=1}^k A_i$.

- 23.** If the variance of 3, 1, 5, 4, and x is 2, find the value(s) of x . Ignore the Bessel correction.

- 24.** If the Bessel-uncorrected variance of the first n positive integers is v , find an expression for n in terms of v . What is the value of n for $v = 8.25$?



Chapter 3

Probability Basics and Random Variables

“God does not play dice.”— Albert Einstein

3.1 Introduction

Probability theory predicts the expected frequencies of specific outcomes of experiments. On the other hand, statistical methods view data as outcomes of probabilistic experiments. Therefore, there is a natural connection between probability and statistics in terms of the relationship between theory and practice. Machine learning is a field that draws ideas from both these fields; the data available to an analyst represents the outcomes of various “experiments” in the real world. Furthermore, the data-driven models of machine learning often use probability theory as a base to build the models that predict various outcomes of interest. In order to set the stage for introducing these models, this chapter will introduce the basics of probability theory.

A rigorous discussion of probabilistic models for machine learning requires the introduction of machinery, notations, and concepts from probability theory. The basic idea is that the data generated in machine learning are assumed to be the outcomes of (grossly simplified) probabilistic processes. Actual data reflects realized outcomes, whereas probability theory talks in terms of *expected* outcomes. Many of the statistical concepts that have already been introduced in the previous chapter (such as mean and standard deviation) have probabilistic connotations in terms of expected outcomes when the raw data is modeled using a probability distribution. For example, when we toss a coin 100 times, we *expect* 50 of them to be heads and the *realized outcomes* from the raw data of die throws might result in 54 heads. There is an inherent variability in the number of heads in a particular number of coin tosses, which is captured by the variance. Like the *sample* variance that is defined over a realized outcome, one can also define the probabilistic concept of *expected* variance

of the number of heads in a particular number of coin tosses. This chapter will concretely define a probabilistic view of several statistical concepts that have already been discussed in the previous chapter.

This chapter will introduce the basics of sample spaces, probability, and random variables. These random variables are often generated by simplified probabilistic *processes* (e.g., coin tosses), and the corresponding probabilities of various outcomes can often be described in closed form by functions referred to as *probability distributions*. A number of common distributions define the mathematical basis on top of which machine learning models are built. However, in order to understand distributions, it is important to first introduce a more general view in which specific closed forms of distributions are not assumed. This chapter constructs this general view, and the next chapter introduces specific distributions that are used commonly in machine learning.

3.1.1 Chapter Organization

This chapter is organized as follows. The next section introduces the concept of sample spaces and events. The computation of probabilities with sample-space analysis is discussed in section 3.3. A set-theoretic view to computing probabilities of events are given in section 3.4. Conditional probabilities and independence are introduced in section 3.5. The Bayes rule is introduced in section 3.6. Probability distributions are introduced in section 3.7. The independence of random variables and conditional distributions are discussed in section 3.8. The summary statistics of probability distributions are introduced in section 3.9. The generation of compound distributions is discussed in section 3.10. Functions of random variables are discussed in section 3.11. A summary is given in section 3.12.

3.2 Sample Spaces and Events

The principles of probability theory are defined with the help of a number of key concepts, such as sample spaces, events, and their associated probabilities. The outcome of a (probabilistic) experiment is a value that is observed after performing the experiment. For example, if tossing a coin yields heads, the outcome of the experiment is heads. In this case, there are two possible outcomes, which are heads and tails — these two outcomes define a set, referred to as the *sample space*. A sample space is defined as follows:

Definition 3.1 (Sample Space) *A sample space Ω is the set of all possible outcomes of an experiment.*

The set Ω may contain either a finite or infinite number of elements. Some examples are as follows:

1. If we toss a coin, the sample space is $\Omega = \{\text{Heads}, \text{Tails}\}$. If we throw a die, the sample space is $\Omega = \{1, 2, 3, 4, 5, 6\}$. If two dice are thrown simultaneously, the sample space contains all possible 6^2 pairs of outcomes. If we observe the color of a ball after selecting it randomly from an urn containing blue, green and red balls, then the sample space is $\Omega = \{\text{Blue}, \text{Green}, \text{Red}\}$. In all these cases, the sample space is *discrete*, and the set of possible outcomes is finite. It is noteworthy that discrete outcomes may or may not be ordered, depending on the application at hand.

2. If we throw a dart that is somehow guaranteed to hit the real number line in the range $[0, 1]$, then the sample space is the set of all real values in $[0, 1]$:

$$\Omega = \{v : 0 \leq v \leq 1\}$$

In this case, the sample space is *continuous*, and the set of all possible values in the sample space is infinite. In such settings, the probability of any particular outcome is infinitesimally small, and a non-zero probability exists only for a *range* of outcomes. For example, the probability of the dart hitting *exactly* at 0.5 will typically be infinitesimally small (irrespective of the manner of dart throw), although a non-zero probability will typically exist for the dart to hit somewhere in the range $(0.4, 0.6)$.

3. One can define a multidimensional sample space using more than one variable as follows:

$$\Omega = \{(v_1, v_2) : v_1^2 - v_2 \geq 1\}$$

Intuitively, the above sample space contains all the points (x, y) lying below the parabola $y = x^2 - 1$.

While continuous sample spaces are always infinitely large, it is also possible for a discrete sample space to be infinitely large. Consider a setting in which you repeatedly toss a coin until it first shows up as tails (denoted by ‘T’) after a sequence of heads (denoted by ‘H’). In such a case, the sample space is the infinitely large set $\{T, HT, HHT, HHHT, HHHHT, \dots\}$. The probability of each successive outcome in this sample space reduces by a factor of 2, and the sum of the probabilities over all outcomes can be shown to be 1. The experiment that forms the basis of the construction of a specific sample space (and corresponding samples) is referred to as a *generating process*.

The notion of a generating process is an important theoretical construct in machine learning because real-world data is always assumed to be the outcome of some unknown generating process, which creates a (possibly infinite) population behind the scenes. Every time an analyst collects a data set, they are implicitly “sampling” data from this generating process. Although the data collection process usually looks nothing like sampling, it is the theoretical construct with which statistical machine learning is enabled. The true generating processes in machine learning applications are usually unknown and impossible to model in a practical sense; therefore, simplified models of generating process are constructed using the observed data as evidence. These models form the holy grail of machine learning because they are used to make all types of predictions that are not directly available in the observed data. This process is referred to as *generalization*, wherein the models are applied to portions of the sample space where observed data are not available.

Example 3.1 Suppose you have ten distinct coins. You flip them in order to create ten different 0–1 outputs (where the outcome of heads is a 1). You apply a multivariate function to these outputs to create a single randomized numerical value X , whose range is unknown. What is the sample space of this experiment and what is its size?

Solution: The sample space can be represented in multiple ways corresponding to a combination of ten coin toss outcomes. If the range of X were known, it could also be used to define the sample space as a set of 1-dimensional possibilities. However, any way of representing the outcome of the experiment in a manner that uniquely defines X can be considered a sample space. Here, it is the raw outcomes of the experiments that define X . The variable X is referred to as a *random variable*, which is often

viewed distinctly from the notion of a sample space.

One possible way to represent the sample space of X is with the use of a 10-dimensional binary vector like $[0, 1, 0, 0, 1, 0, 0, 0, 0, 1]$. Each position of the vector corresponds to a coin, and each position value in the vector corresponds to an outcome of the corresponding coin toss. The size of the sample space is equal to the number of such vectors, which is $2^{10} = 1024$. ■

The next concept that is defined is that of an *event*.

Definition 3.2 (Event) *Any subset $A \subseteq \Omega$ of the sample space Ω is an event, and it corresponds to the occurrence of any one of the outcomes in set A . For finite sample spaces of size $p = |\Omega|$, the number of possible events is 2^p , since an event is any element of the powerset of the sample space.*

Intuitively, an event provides a less fine-grained way of representing outcomes than the sample space, because it represents the occurrence of *any one* of the outcomes in $A \subseteq \Omega$. For example, the individual die faces are the most fine-grained way of representing the outcomes of a die roll, and the event corresponding to the occurrence of an odd face contains a subset of three outcomes. Not all of the 2^6 subsets of outcomes can be defined with an intuitive English language description (like “occurrence of odd face”). Both ϕ and Ω are valid events corresponding to the occurrence of no face (with probability 0) or the occurrence of any face (with probability 1). Since events are sets, they inherit definitions such as *exhaustiveness* and *mutual exclusivity* from set theory:

Definition 3.3 (Exhaustive Events) *A set of events $A_1 \dots A_r$ is exhaustive if $\cup_{i=1}^r A_i = \Omega$.*

In other words, an exhaustive set of events includes all outcomes in the sample space and therefore at least one of the events is guaranteed to occur. For example, the event corresponding to a face less than 5 showing up and the event corresponding to a face greater than 2 showing up are together exhaustive.

Definition 3.4 (Mutually Exclusive Events) *A set of events $A_1 \dots A_r$ is mutually exclusive, if for any pair of events A_i and A_j satisfying $i \neq j$ we have $A_i \cap A_j = \phi$.*

Since no outcome is common between any pair of events in $A_1 \dots A_r$, it is impossible for any pair of events in $A_1 \dots A_r$ to occur together in an experiment. For example, the events corresponding to a die face less than 2 showing up and a die face greater than 4 showing up are mutually exclusive. However, these events are not exhaustive because they do not cover the outcomes 2, 3, and 4. The event corresponding to an odd face showing up and the event corresponding to an even face showing up are both exhaustive and mutually exclusive.

Example 3.2 *Describe which events are exhaustive, mutually exclusive, or both. (i) Outcomes less than 5 and outcomes greater than 3 in die rolls; (ii) Outcomes divisible by 3 and those not divisible by 3 in die rolls; (iii) Outcome of 2 and outcomes greater than 4 in die rolls; (iv) Outcomes less than 3 and odd outcomes in die rolls.*

Solution: The case (i) is exhaustive but not mutually exclusive because the two events cover the entire sample space, but the outcome 4 is common to both. Therefore, both events can occur and at least one of these events must occur.

The case (ii) is both exhaustive and mutually exclusive. No outcome is common to both events, which cover the entire sample space.

The case (iii) is mutually exclusive but not exhaustive. The outcomes $\{2, 5, 6\}$ do not cover the entire sample space.

The case (iv) is not mutually exclusive and it is also not exhaustive. The outcomes 4 and 6 are not covered by either event. At the same time, the outcome 1 is common to both. Therefore, both events could occur together or neither event might occur (depending on the outcome). ■

One can view the sample space to be a collection of an exhaustive and mutually exclusive set of events of the finest granularity. Therefore, the outcomes in the sample space are referred to as *primitive events* in terms of which all other events are defined. The set of all possible events is referred to as the event space.

Definition 3.5 (Event Space) *The set of all possible events is referred to as the event space.*

The event space is the power set of Ω , which has size $2^{|\Omega|}$. For example, in a throw of two dice, the sample space Ω might contain 36 possible outcomes:

$$\Omega = \{(x, y) : x, y \leq 6; x, y \in \mathbb{N}\}$$

There are 2^{36} possible events in the event space, and not all of them can be described in an intuitive way with the use of a simple description. An example of an intuitively interpretable event is one in which the two faces sum to 5. The corresponding event A contains exactly the four outcomes $A = \{(1, 4), (2, 3), (3, 2), (4, 1)\}$.

Events can also be defined over infinite sample spaces. Rather than provide a formal definition, we provide a number of examples of events over both finite and infinite sample spaces in order to provide an understanding of this natural extension:

- *Examples of events defined over finite sample spaces:* If we toss a coin, the event space is $\{\phi, \{\text{Heads}\}, \{\text{Tails}\}, \{\text{Heads, Tails}\}\}$. If we throw a die, the event space contains 2^6 possible sets of events. An event such as $\{1, 2, 5\}$ corresponds to the case where face 1, 2, or 5 shows up. If we observe the color of a ball after selecting it randomly from an urn containing blue, green and red balls, then the event space contains $2^3 = 8$ elements. Note that a member event of the event space, such as $\{\text{Blue, Green}\}$ corresponds to the situation where either a blue or a green ball is drawn. Since the sample space is finite, the event space is finite as well.
- *Examples of events defined over infinite sample spaces:* Consider the earlier example of throwing a dart on the number line with sample space Ω defined as follows:

$$\Omega = \{v : 0 \leq v \leq 1\}$$

In this case, an event is any union of ranges from the sample space. For example, an event could be $(0.2, 0.3] \cup [0.6, 0.7)$, which corresponds to the scenario that the dart hits somewhere in the aforementioned range. As in the case of sample spaces of continuous outcomes, the event spaces of continuous outcomes are infinitely large.

Now consider the 2-dimensional sample space containing all points below the parabola $y = x^2 - 1$:

$$\Omega = \{(v_1, v_2) : v_1^2 - v_2 \geq 1\}$$

In this case, an example of a valid event A is as follows:

$$A = \{(v_1, v_2) : v_1^2 - v_2 \geq 1, v_1 + v_2 \geq 1\}$$

This event consists of all points lying below the parabola $y = x^2 - 1$, but above the line $y = 1 - x$. It is also noteworthy that the space of events $A_L = \{(v_1, v_2) : v_1 + v_2 \geq 1\}$ (corresponding to all points above the line $y = 1 - x$) is not a valid event inside the sample space $\Omega = \{(v_1, v_2) : v_1^2 - v_2 \geq 1\}$ — this is because the definition of A_L does not use the parabolic constraint to exclude points that lie above the parabola $y = x^2 - 1$. In the case of finite sample spaces, events are elements of the power-set of outcomes. In the case of infinite sample spaces, events must continue to be valid subsets of the original sample space of outcomes (although the notion of a power set does not exist).

Consider the earlier example of the throw of two dice in which the sample space Ω is of size 36. The probability of the event that the dice faces sum to 4 is given by $4/36 = 1/9$, because it contains the four outcomes $\{(1, 4), (2, 3), (3, 2), (4, 1)\}$ out of 36 equally likely outcomes. By summing the probabilities of the four outcomes in this event, one can obtain the probability of the event. Therefore, the probability of an event can be viewed as the generalization of the concept of the cardinality of a set, in which each element is given a probability weight. The process of mapping events to probabilities is referred to as *probability mapping*, which can be combined with the concepts of sample and event spaces to define a *probability space*:

Definition 3.6 (Probability Space) *A probability space is a mathematical construct containing the following elements:*

- A sample space Ω , which is a set containing all possible outcomes.
- An event space containing all possible subsets of the sample space including the null event ϕ and the universal event Ω .
- A mapping from each element A of the event space to a probability value $P(A)$ in $(0, 1)$ representing the probability of any outcome in that event occurring. Since events are sets and their probabilities are intuitively similar to the “cardinalities” of these sets, the probability mapping must satisfy the set-theoretic laws of exhaustiveness and mutual exclusivity in order to be considered valid.

It is common to work with a *random variable* view of probabilities. Random variables are defined by mapping each outcome in the sample space to a variable value.

Definition 3.7 (Random Variable) *A random variable is obtained by mapping each outcome in the sample space to a value (or a vector of values). The probability of a particular outcome of the random variable is the sum of probabilities of all sample-space outcomes that map to that variable value.*

Random variables are denoted by capital letters such as X , and their outcomes are denoted by corresponding small letters such as x . Multivariate random variables are denoted by capitalized vectors such as \vec{X} and their instantiations by small letters, such as \vec{x} . The data type of the random variable depends on the nature of the underlying experiment that generates the variable. For example, the outcomes of coin-tosses and die rolls are integer-valued variables like $\{0, 1\}$ and $\{1, 2, 3, 4, 5, 6\}$, whereas the ball color obtained from an urn draw is a categorical-valued variable such as $\{B, G, R\}$.

In some special types of experiments (such as throwing a dart on the real number line), the value of the outcome (position of dart) is indistinguishable from the random variable, but sample space outcomes are distinguishable from random variables in general. For example, if a random variable is denoted as the sum of the faces of two dice, then the subset of sample-space elements $(1, 6)$, $(2, 5)$, $(3, 4)$, $(4, 3)$, $(5, 2)$ and $(6, 1)$ map to the same random variable value of 7. Therefore, the mapping from sample spaces to random variables might be a many-to-one mapping in the general case. The experiments (i.e., dice rolls) that eventually map to variable values are referred to as generating processes. It would have been impossible to derive the probabilities of random variable outcomes without working with the more fine-grained sample space of dice-roll outcomes. This is the reason that it is sometimes convenient to distinguish random variables from the underlying sample spaces.

Events map to random variables in terms of constraints on the values of the variables. For example, consider an application in which the data on collected shark lengths are assumed to be outcomes of a random process. In such cases, the sample space may be viewed as all nonnegative real values, and an event may be viewed as a specific constraint on the length of a shark (say, between 15 feet and 17 feet). Therefore, if X corresponds to the random variable indicating shark length, the probability of this event can be denoted as $P(15 \leq X \leq 17)$.

The probability distribution of a discrete random variable with a finite number of outcomes can be represented in *tabular form* with two columns. The left column contains each outcome value x of the random variable X and the right column contains the numerical value of $P(X = x)$. The probability $P(X = x)$ is denoted by $p_X(x)$, and the table of outcome-probability pairs is referred to as the *probability mass function* (PMF) of the random variable. Since the PMF is a function, it can either be represented in tabular form or in closed form — the former is the most general way of defining an arbitrary function over discrete values. We define the concept of a probability mass function concretely below:

Definition 3.8 (Probability Mass Function) *The probability mass function (PMF) of a discrete random variable is a function $p_X(x)$ mapping each outcome x of a random variable X to the probability value $P(X = x)$.*

For discrete random variables with a finite set of outcomes, a probability mass function can always be represented as a table of outcome-probability pairs. An example of the probability mass function for the random variable indicating the outcomes of a die throw is shown in Table 3.1. The sum of the probabilities in a PMF table will always be 1, since the different outcomes of the random variable exhaustively cover the sample space. The number of rows in the PMF table may be fewer than the number of elements in the sample space, since multiple elements in the sample space may map to the same value of the random variable. For example, if two dice are thrown together, the sample space consists of all 36 combinations of the faces, but a random variable corresponding to the sum of the faces only has 11 possibilities, corresponding to integer values from 2 to 12. This is because multiple possibilities such as $5 + 2$ and $4 + 3$ map into the same outcome of the random variable of 7. In other words, the outcomes corresponding to the sample space are often of finer granularity than the outcomes of the random variable. The representations of continuous random variables cannot be expressed in tabular form because the sample space is infinitely large and probabilities are only associated with *ranges of values* rather than specific values. In such cases, one must use *probability density functions*, which are usually represented in closed form. A detailed discussion of probability density functions is provided in the next chapter.

Table 3.1: The PMF for the discrete random variable outcomes of a die throw.

Outcome x	$p_X(x)$
1	1/6
2	1/6
3	1/6
4	1/6
5	1/6
6	1/6

Example 3.3 Suppose that you toss three fair, clearly distinguishable coins simultaneously, and observe the triplet of values on the coins as a single outcome of the triple-coin toss experiment. How would you represent the sample space of this experiment as a 3-dimensional discrete random-variable vector of binary values? Give an example of an event from this sample space and describe it semantically in plain English. What is the size of the event space in this problem?

Solution: One can assume that “heads” corresponds to 1 and “tails” corresponds to 0. Then, the vector $[0, 1, 1]$ refers to an outcome where a tails is followed by two heads. The corresponding sample space is as follows:

$$\Omega = \{[0, 0, 0], [0, 0, 1], [0, 1, 0], [0, 1, 1], [1, 0, 0], [1, 0, 1], [1, 1, 0], [1, 1, 1]\}$$

An example of an event $A \subseteq 2^\Omega$ is as follows:

$$A = \{[0, 1, 1], [1, 0, 1]\}$$

There are many semantic ways of describing this event but one possible way is as follows:

Exactly one of the first two tosses is a heads and the third toss is also a heads.

Not all events will have simple semantic descriptions, especially if the outcomes in the event do not have common characteristics. The event space may contain any subset of outcomes from the sample space (which may not always be easy to relate to one another). For example, it can even contain the empty set, which is not particularly interesting (because this event never occurs). Since the size of the sample space Ω is 8, the number of possible events is equal to the size of the powerset of Ω , which is $2^8 = 256$. ■

Problem 3.1 Assuming that each outcome in the sample space of Example 3.3 is equally likely, construct the PMF table of the corresponding random variable vector. Make sure that the probability entries of the table sum to 1.

Problem 3.2 Suppose that you define a new random variable that is equal to the number of heads in the three tosses of Example 3.3. Construct a PMF table of the random variable using the results of Problem 3.1.

3.3 The Counting Approach to Probabilities

An important part of probabilistic analysis is to be able to map random variables to probability values. Choosing an appropriate sample space is often a critical part in constructing the table corresponding to the probability mass function. A particularly simple case is one in which the sample space contains a number of equally likely outcomes, and each outcome x of the random variable X maps to one or more outcomes of the sample space. In this simplified setting, the mapping from a discrete random variable value $X = x$ to the probability $p_X(x)$ can be expressed by dividing the number of outcomes resulting in a value of x by the total number of outcomes in the sample space.

$$P(X = x) = \frac{\text{Number of outcomes resulting in } X = x}{\text{Total number of outcomes}}$$

Note that the constraint $X = x$ can be viewed as an event A containing a subset of outcomes from the sample space. One can state this result more generally in terms of events (sets of outcomes) rather than in terms of random variables:

$$P(A) = \frac{\text{Number of outcomes in set } A}{\text{Total number of outcomes in sample space}} = \frac{|A|}{|\Omega|}$$

The counting approach to probabilities is very useful in cases where casewise analysis needs to be used on the sample space. This point is best illustrated with the following example:

Example 3.4 Consider the case where two fair dice are thrown, and the discrete random variable X is the sum of the values on their faces. Find the probability $P(X = 7)$.

Solution: The total number of possible outcomes in the sample space of face pairs is $6^2 = 36$. The outcomes that result in a sum of 7 are $(1, 6)$, $(2, 5)$, $(3, 4)$, $(4, 3)$, $(5, 2)$ and $(6, 1)$; therefore, there are a total of six outcomes in which the face values add to 7. Since the outcomes in the sample space are equally likely, the probability of $X = 7$ is given by the following:

$$P(X = 7) = \frac{6}{36}$$

■

The following example with coin throws serves to illustrate the same point:

Example 3.5 Consider the case when a fair coin is thrown 10 times. Find the probability that heads shows up 6 times.

Solution: In this case, the sample space can be defined as an arrangement of ten H or T values, where a value of H indicates a heads outcome and a value of T indicates a tails outcome. Therefore, each element of the sample space is a string like HHTHTTTTHT. First, note that the size $|\Omega|$ of this sample space Ω of arrangements is $2^{10} = 1024$. The number of ways in which one can arrange 6 heads and 4 tails is given by $\binom{10}{6} = 210$. For fair coins, all possible arrangements are equally likely and correspond to various outcomes of the sample space. Therefore, the probability of 6 heads is given by $210/1024$.

■

Problem 3.3 Consider the random variable X that is the sum of the faces of two fair dice rolled together. Use the counting method to express the PMF of X as a table. How many entries are there in the table? What do the probability values in the table sum to?

Problem 3.4 Consider the random variable X , which is the number of “heads” outcomes in 10 tosses of a coin. Use the counting method to express the PMF of X as a table. How many entries are there in the table? What do the probability values in the table sum to?

Note that in order for this counting approach to work, all outcomes in the sample space must be equally likely. It is possible to extend the counting approach to cases where all outcomes in the sample space do not have the same probability. Let us treat the probability of each outcome o_i in the sample space as a “weight” $p(o_i)$. Then, the probability of outcome set (event) A is expressed in terms of the aggregate weights of its constituent elements as follows:

$$\begin{aligned} P(A) &= \frac{\text{Weight of outcomes in set } A}{\text{Total weight of outcomes in sample space}} \\ &= \frac{\sum_{o_i \in A} p(o_i)}{\sum_{o_i \in \Omega} p(o_i)} = \sum_{o_i \in A} p(o_i) \end{aligned}$$

The expression above is simplified by observing that the sum of the probabilities of all outcomes in the sample space Ω is 1, and therefore the denominator of the above fraction disappears. The result $P(A) = \sum_{o_i \in A} p(o_i)$ is a special case of the *additive rule for mutually exclusive events* and is discussed in the next section.

Problem 3.5 Consider the random variable X , which is the sum of the faces of two fair dice rolled together. You have already created the PMF table in Problem 3.3. Use this table along with the weighted counting method to determine the probability of the event that the sum of the faces of the dice is divisible by 3.

3.4 Set-Wise View of Events

Because of the relationship between events and sets, the counting approach to probabilities helps set the stage for generalizing some well-known results on sets to probabilities. A well-known result on union of sets relates the cardinality of the union and intersection of sets A and B :

$$|A \cup B| = |A| + |B| - |A \cap B|$$

Intuitively, if A is the event that a person is Native American and B is the event that the person is a female, then the number of people who are either Native American or female is obtained by adding the number of Native American people and the number of females, and then subtracting the number of double-counted Native American females (who are included in both sets A and B). Note that this result can also be extended to the weights (rather than cardinality) of sets. Therefore, one can extend the result to probabilities as well (based on the weighted counting rule for probabilities):

Theorem 3.1 (Set Union Rule for Probabilities) Let A and B be two events in the event space. Then, the probabilities of their union and intersection are related as follows:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

In other words, the probability that either event occurs is obtained by summing the probabilities of the two events and subtracting the double-counted probability that both events occur. The set-union rule automatically leads to the additive rule for *mutually exclusive events*. In mutually exclusive events, $A \cap B = \phi$, and therefore $P(A \cap B) = 0$. Therefore, one can add the probabilities of mutually exclusive events in order to obtain the probability of either of the two events occurring. We summarize this result below:

Corollary 3.1 (Additive Rule for Mutually Exclusive Events) *Let A and B be two events in the event space that are mutually exclusive. Then, the probability that either of the two events occurs is the sum of the probabilities of the individual events:*

$$P(A \cup B) = P(A) + P(B)$$

In fact, one can generalize the above rule to any number of mutually exclusive events by induction:

Corollary 3.2 *Let A_1, A_2, \dots, A_r be events in the event space that are mutually exclusive. Then, the probability of their union is the sum of the probabilities of the individual events:*

$$P(\bigcup_{i=1}^r A_i) = \sum_{i=1}^r P(A_i)$$

As in set theory, the notion of the complement of an event A , which is denoted by A^c , can be defined as the set of outcomes in $\Omega - A$, where Ω is the sample space:

Definition 3.9 (Event Complement) *The complement A^c of an event A defined over the sample space Ω is the set of all outcomes $A^c = \Omega - A$ not included in event A . In other words, A^c occurs if and only if A does not occur.*

The events A and A^c are referred to as *complementary events*. In this case, we have $(A \cup A^c) = \Omega$ and $(A \cap A^c) = \phi$. The probability of the event complement is obtained by subtracting the probability of the event from 1, since the sum of the probabilities over all outcomes is 1.

Lemma 3.1 (Probability of Event Complement) *The probability $P(A^c)$ of the complement A^c of event A is given by $P(A^c) = 1 - P(A)$.*

It is not difficult to see that the aforementioned rule also arises from the counting approach to probabilities. One can also show this result by using the set-union rule:

$$\underbrace{P(A \cup A^c)}_1 = P(A) + P(A^c) - \underbrace{P(A \cap A^c)}_0$$

As in the case of event complement, the subtraction of events corresponds to all outcomes in one event but not in the other.

Definition 3.10 (Subtraction of Events) *The subtraction $A - B$ of the events A and B is the set of outcomes in event A but not in event B . In other words, $A - B$ is the same as $A \cap B^c$.*

One can generalize the set union rule to that of subtraction of events by using the fact that $(A - B)$ is equivalent to the set $A - A \cap B$. Therefore, the counting approach to probabilities can be used to show the following:

Lemma 3.2 (Probability of Subtraction) *The probability of the subtraction of events A and B is given by the following:*

$$P(A - B) = P(A) - P(A \cap B)$$

Proof: The event A can be expressed as the union of the two mutually exclusive events $A \cap B$ and $A \cap B^c$. Using the additive rule of mutually exclusive probabilities, one obtains the following:

$$\begin{aligned} P(A) &= P(A \cap B) + P(A \cap B^c) \\ &= P(A \cap B) + P(A - B) \end{aligned}$$

Rearranging the above relationship, one obtains the desired result. ■

Example 3.6 Suppose that two dice are thrown. Find the probability that the sum of the two faces is not 7.

Solution: Let the sample space be represented by (x, y) , where x and y are the values of the die faces. The relevant portion of the sample space that results in a sum of 7 is as follows:

$$A = \{(1, 6), (2, 5), (3, 4), (4, 3), (5, 2), (6, 1)\}$$

Note that there are 36 possible outcomes. The probability of the event A (i.e., sum of faces is 7) is $6/36$ using the counting approach to probabilities. Then, the probability of its complement event is $1 - 6/36 = 5/6$. ■

Example 3.7 John will eat eggs today with probability 0.5 and toast with probability 0.6. He will have both with probability 0.45. Find the probability of eating only eggs and the probability of eating only toast. What is the probability that he will have exactly one of them?

Solution: Let E be the event that John will eat eggs and T be the event that John will eat toast. The probability of eating only eggs is $P(E - T) = P(E) - P(E \cap T) = 0.5 - 0.45 = 0.05$. Similarly, the probability of eating only toast can be shown to be 0.15.

The probability of eating exactly one of the two is given by $P(E - T) + P(T - E)$. By using Lemma 3.2 to substitute for $P(E - T)$ and $P(T - E)$, one can derive an algebraic expression for this probability as follows:

$$P(E - T) + P(T - E) = P(E) + P(T) - 2P(E \cap T) = 0.5 + 0.6 - 0.9 = 0.2$$

Therefore, the probability that John will have exactly one of the two options is 0.2. Note that this value is also equal to the sum of the probabilities of eating only eggs (0.05) and eating only toast (0.15). ■

Problem 3.6 The probability that it is sunny in New York tomorrow is $1/3$ and the probability that it is sunny in Boston tomorrow is also $1/3$. The probability that it is sunny in

both cities tomorrow is $1/4$. Find the probability that it is sunny in at least one of the two cities tomorrow. What is the probability that it is sunny tomorrow in New York but not in Boston?

3.5 Conditional Probabilities and Independence

When an event A is known to have occurred, it often affects our estimation of the probability of a related event B to have occurred. For example, if we are told that a fair die shows an even face (event A), it changes the probability of that face value being 6 (event B) to $1/3$ from the original value of $1/6$. This change is a natural outcome of the fact that the probability of event B is now being *conditioned* on additional knowledge of related outcomes. We define the notion of conditional probabilities of events as follows:

Definition 3.11 *The conditional probability $P(B|A)$ is the probability that one of the outcomes representing event B occurs, given that one of the outcomes representing event A has been known to occur. The conditional probability $P(B|A)$ is related to the probability of A and B as follows:*

$$P(B|A) = \frac{P(B \cap A)}{P(A)}$$

The above mathematical relationship for conditional independence can be derived by understanding that the conditional occurrence of event B occurs with respect to occurrence A when an outcome o_i belongs to $B \cap A$. However, the “new” sample space Ω_A for occurrence of any outcome given event A is the set of outcomes in A . Therefore, by using the weighted counting view of probabilities, one derives the following:

$$\begin{aligned} P(B|A) &= \frac{\text{Weight of outcomes in } B \cap A}{\text{Total weight of outcomes in sample space } \Omega_A = A} \\ &= \frac{\sum_{o_i \in B \cap A} p(o_i)}{\sum_{o_i \in A} p(o_i)} = \frac{P(B \cap A)}{P(A)} \end{aligned}$$

In order to understand the notion of conditional probabilities, we will use an example.

Example 3.8 Consider the sample space of six outcomes defined by a die roll, in which event A is the set of outcomes in which the die face lands at a value of at most 3 and event B is the set of outcomes in which the die face lands at an odd value. Compute the probability $P(B|A)$.

Solution: We want to find $P(B|A)$. By conditioning on values 1, 2, and 3, we are restricting ourselves to a new sample space of only three outcomes (instead of the six outcomes of the die), and computing the probability of events corresponding to an odd outcome out of these three outcomes. Using the counting view of probabilities, we have:

$$P(B|A) = \frac{\text{Count of outcomes 1 and 3}}{\text{Count of restricted sample space of outcomes 1, 2, and 3}} = \frac{2}{3}$$

It is easy to verify that $P(B \cap A) = 1/3$ and $P(A) = 1/2$, which yields the same value of $P(B|A)$ using the aforementioned formula:

$$P(B|A) = \frac{P(B \cap A)}{P(A)} = \frac{1/3}{1/2} = \frac{2}{3}$$

■

One can rearrange the relationship in Definition 3.11 in order to compute the probability of events A and B occurring together in terms of conditional probabilities. The probabilities of events A and B occurring together can be expressed using conditional probabilities in the following alternative forms:

$$P(A \cap B) = P(A)P(B|A) \quad (3.1)$$

$$P(A \cap B) = P(B)P(A|B) \quad (3.2)$$

These alternative forms are very useful because they form the genesis of a fundamental rule in probability theory, referred to as the *Bayes rule*. This rule is used frequently in all types of settings in machine learning, the most famous of which is the *Bayes classifier*. The Bayes rule is discussed in detail in the next section.

In the earlier example of the die, the conditional probability of an odd face (given that the face value is at most 3) is $2/3$, whereas the unconditional probability of an odd face is $1/2$. Therefore, conditioning one event on the other changes the former's probability, which means that the two events are not *independent*. Correspondingly, the notion of independence of events is defined as follows:

Definition 3.12 (Independent Events: Conditional Form) *Two events A and B are independent if and only if $P(B|A) = P(B)$.*

The notion of independence can be expressed in several alternative forms. First, note that if we had switched the notation of A and B , we could very easily have written $P(A|B) = P(A)$ (which would also have been correct). Both of these forms come from the product rule of independence, which is symmetric in A and B and may be written as follows:

Definition 3.13 (Independent Events: Product Form) *Two events A and B are independent if and only if $P(A \cap B) = P(A) \cdot P(B)$.*

To understand why the product form and the conditional form of independence are the same, one can write the product form of independence as $P(B) = P(A \cap B)/P(A) = P(B|A)$. The product form of independence is often preferred because of its symmetry across events and ease of generalizability to multiple events.

Definition 3.14 (Independent Events: General Product Form) *The events $A_1 \dots A_r$ are independent if and only if $P(\cap_{i=1}^r A_i) = \prod_{i=1}^r P(A_i)$.*

For independent events, the occurrence of one event does not affect the probability of occurrence of the other and vice versa. An excellent example of a pair of independent events A and B is that of heads turning up in the first and second tosses of the coin, respectively. Note that $P(A \cap B)$ is $1/4$ because there are four possibilities for how the coins show up and only one of them results in the outcome of both of them showing up as heads. We already know that the probability of the individual tosses showing up as heads is $P(A) = P(B) = 1/2$. Therefore, it is clear that in this case, we have $P(A \cap B) = P(A) \cdot P(B) = 1/4$.

A closely related notion to that of independence is that of *conditional independence*. The notion of conditional independence is defined as follows:

Definition 3.15 (Conditional Independence) *Two events A and B are conditionally independent with respect to event C, if the following is satisfied:*

$$P(A \cap B|C) = P(A|C) \cdot P(B|C)$$

To provide a concrete example of conditional independence, consider two biased coins, the first of which turns up heads 90% of the time and the second turns up tails 90% of the time. One of the two coins is selected randomly with equal probability and then the selected coin is tossed twice. Because of the bias in the coins, the outcomes of the two tosses are highly correlated, and therefore knowing the outcome of one toss affects the conditional probability of the outcome of the second toss. Therefore, the events of heads showing up in the two cases are not independent. However, if we condition on the fact that a particular coin was known to be selected, the outcomes of the two tosses become conditionally independent. The details of the probabilities of this setting are explored in Problem 3.10.

Example 3.9 *John will eat eggs today with probability 0.5 and toast with probability 0.6. He will have both with probability 0.45. What is the probability that he will have toast today given that he will have eggs. Are the two events independent?*

Solution: Let E be the event that John will eat eggs and T be the event that John will eat toast. We want to compute the probability $P(T|E)$, which can be computed as follows:

$$P(T|E) = \frac{P(T \cap E)}{P(E)} = \frac{0.45}{0.5} = 0.9$$

The events T and E are not independent because $P(T|E) \neq P(T)$. One can also verify using the alternative definition of independence that $P(T \cap E) \neq P(T) \cdot P(E)$. After all, the two definitions are equivalent. ■

Problem 3.7 *If you throw a die in succession three times, find the probability that you obtain 3, 4, and 5 in that specific order. What is the probability that you get 3, 4, and 5 in arbitrary order in the three throws? What is the probability that you never get 3, 4, or 5 in any throw? What is the probability that you get at least one 3, 4, or 5 in three throws?*

Problem 3.8 *Two dice are thrown simultaneously. Find the conditional probability that the sum of their outcomes is 2, given that it is known to be 5 or less.*

Problem 3.9 *Consider an urn containing two green balls and two red balls. Balls are selected from the urn without replacement. Event A corresponds to selecting a red ball on the first try and event B corresponds to selecting a green ball on the second try. Are events A and B independent? Would your answer change if balls were selected from the urn with replacement?*

Problem 3.10 *Suppose that C is the event that you select the heads-favoring coin when selecting with equal probability from two biased coins — one shows up heads 90% of the time and the other shows up tails 90% of the time. You toss the selected coin twice. The event that the first toss is heads is A and the event that the second toss is heads is B. Compute $P(A)$, $P(B)$, and $P(A \cap B)$. Are the events A and B independent? Now compute $P(A|C)$, $P(B|C)$, and $P(A \cap B|C)$. Are the events A and B conditionally independent given event C?*

3.6 The Bayes Rule

The notion of conditional probabilities helps in the development of a rule known as the Bayes rule. Before discussing the Bayes rule, we set up an important rule known as the *Total Probability Rule*. This rule decomposes the probability of an event as a weighted sum of its conditional probabilities over two complementary events, where the weights of the conditional probabilities are the probabilities of these complementary events:

Lemma 3.3 (Total Probability Rule) *Let A and B be two events. Then the probability of A can be expressed as a weighted sum of the conditional probabilities of A with respect to B and B^c as follows:*

$$P(A) = P(B)P(A|B) + P(B^c)P(A|B^c)$$

Proof: The event A can be decomposed into two mutually exclusive events corresponding to $A \cap B$ and $A \cap B^c$. Using the additive rule for mutually exclusive events, one obtains the following:

$$P(A) = P(A \cap B) + P(A \cap B^c)$$

Then, by using $P(A \cap B) = P(A|B)P(B)$ and $P(A \cap B^c) = P(A|B^c)P(B^c)$, one obtains the desired result. ■

The total probability rule can be generalized to the case where the sample space Ω is decomposed into more than two events (rather than just B and its complement):

Lemma 3.4 (Total Probability Rule: General Version) *Let A be an event, and let B_1, B_2, \dots, B_r be mutually exclusive and exhaustive events so that $\cup_{i=1}^r B_i = \Omega$. Then the probability of A can be expressed as a weighted sum of the conditional probabilities of A with respect to $B_1 \dots B_r$ as follows:*

$$P(A) = \sum_{i=1}^r P(B_i)P(A|B_i)$$

The Bayes rule relates $P(B|A)$ to $P(A|B)$ and $P(A|B^c)$ using the definition of conditional probabilities and the total probability rule. First, note that the definition of conditional probabilities allows us to express $P(A \cap B)$ in the alternative forms $P(A)P(B|A)$ and $P(B)P(A|B)$. These two equivalent forms allow us to express $P(B|A)$ in terms of $P(A|B)$ as follows:

$$P(B|A) = \frac{P(B)P(A|B)}{P(A)}$$

Using the total probability rule to expand the value of $P(A)$ in the denominator, one obtains the following:

$$P(B|A) = \frac{P(B)P(A|B)}{P(B)P(A|B) + P(B^c)P(A|B^c)}$$

This rule is referred to as the Bayes rule:

Lemma 3.5 (Bayes Rule) *Let A and B be two events such that $P(B)$, $P(A|B)$, and $P(A|B^c)$ are known. Then, $P(B|A)$ can be computed using the following relationship:*

$$P(B|A) = \frac{P(B)P(A|B)}{P(B)P(A|B) + P(B^c)P(A|B^c)}$$

The Bayes rule can be generalized to the case where the sample space Ω is decomposed into more than two events (rather than just B and its complement):

Theorem 3.2 (Bayes Rule: General Version) *Let A be an event, and let B_1, B_2, \dots, B_r be mutually exclusive and exhaustive events so that $\cup_{i=1}^r B_i = \Omega$. Then, each $P(B_i|A)$ can be expressed in terms of the various $P(B_i)$ and $P(A|B_i)$ as follows:*

$$\overbrace{P(B_i|A)}^{Posterior} = \frac{\overbrace{P(B_i)}^{Likelihood} \overbrace{P(A|B_i)}^{Likelihood}}{\sum_{j=1}^r P(B_j)P(A|B_j)}$$

The value $P(B_i)$ is referred to as the *prior probability* (because it is the value of $P(B_i)$ before we know anything about A), whereas the value of $P(B_i|A)$ is the *posterior probability* because it quantifies the probability of the same event B_i after we know the outcome of event A . The quantity $P(A|B_i)$ is referred to as a *likelihood*. In order to understand the significance of prior and posterior probabilities, we will use a real-world example from infectious disease testing:

Example 3.10 *A university institutes an infectious disease testing program in which students are tested weekly with a rapid test and those who test positive for the disease are quarantined immediately. Unfortunately, tests are not completely accurate and the results are accurate 98% of the time whether or not the student has the disease. The university is very successful in its testing program, as a result of which only 1% of the students taking a test on any given day actually have the disease (whether or not they test positive). What is the probability that a quarantined student has the disease?*

Solution: At first glance, it would seem that 98% of the quarantined students will be truly positive, since this is the accuracy rate of the test. However, such an approach would be incorrect because it does not account for the fact that the *prior* probability of a person having the disease is very low. Let B be the event that a person truly has the disease, and let A be the event that a person tests positive (and is therefore quarantined). The fraction of truly infected students among quarantined students is given by $P(B|A)$. Using the Bayes rule, we have the following:

$$P(B|A) = \frac{P(B)P(A|B)}{P(B)P(A|B) + P(B^c)P(A|B^c)} = \frac{0.01 * 0.98}{0.01 * 0.98 + 0.99 * 0.02} \approx 0.33$$

In other words, only 33% of the quarantined students are truly positive! ■

It is noteworthy that many testing programs for rare and serious illnesses (e.g., AIDS) recommend a second test after a positive result because the large prior probability of a negative result causes the doctor to see a large percentage of false positives. Assuming that the outcomes of two tests are independent events (conditional on the patient's disease status), using a succession of two tests greatly boosts the accuracy of the testing process. Because of this reason, the Bayes rule is often taught (at least at a qualitative level) to medical students.

Example 3.11 A company has three factories A, B, and C that produce 50%, 30%, and 20% of their widget product. The probability of a manufactured widget being defective for factories A, B, and C are 1%, 3%, and 2%, respectively. You buy a widget in the market and find it to be defective. What is the probability that it was produced by factory A?

Solution: The general version of the Bayes rule can be used to find the probability that the defective widget is produced by factory A:

$$P(\text{Factory A|Defective}) = \frac{0.5 * 0.01}{0.5 * 0.01 + 0.3 * 0.03 + 0.2 * 0.02} = \frac{5}{18}$$

Even though factory A produces half the widgets, its production quality is relatively high and therefore the posterior probability of a defective widget being produced by factory A is only 5/18. ■

Problem 3.11 Consider the same problem as Example 3.10, except that two consecutive positive tests are required to quarantine a student. In this case, what is the probability that a quarantined student truly has the disease? You may assume that the tests are conditionally independent, given the patient's disease status.

Problem 3.12 Suppose that you have a fair coin and a two-headed coin. You select one of the two coins randomly with equal probability and then toss it without knowing which coin you selected. The coin shows up as heads. What is the probability that you had selected the fair coin? How would your answer have changed if the coin had shown up as tails.

Problem 3.13 Jim either walks to work or drives to work. Jim walks to work with probability 0.8 if it is sunny, and he drives to work with probability 0.4 if it is not sunny. The probability of a randomly selected day being sunny is 0.7. You are told that Jim drove to work today but you are not told anything about the weather. What is the probability that it was sunny today?

3.6.1 The Observability Perspective: Posteriors versus Likelihoods

The notion of likelihood plays a central role in statistics and data science. Although both posterior probabilities and likelihoods are conditional probabilities, it is important not to confuse the two in terms of how they are interpreted from a statistical perspective. Stated simply, likelihood is a special class of conditional probabilities that quantify the probability¹ of an observable event, given that a particular (unobservable) scenario (or *hypothesis*) holds. Likelihoods are attached to observable events, whereas prior and posterior probabilities are (typically) attached to unobservable hypotheses. Much of machine learning is about using data (observations) to make predictions about (directly unobservable) hypotheses. The Bayes rule provides an early example of how our observation of an event changes our estimation of the probability of an unobservable event (hypothesis) from a prior value to a posterior value. We repeat the general form of the Bayes rule here:

$$P(B_i|A) = \frac{P(B_i)P(A|B_i)}{\sum_{j=1}^r P(B_j)P(A|B_j)}$$

¹As we will see later, likelihoods might also refer to probability *densities* for continuous random variables.

In this case, the alternatives B_1, B_2, \dots, B_r are the r different hypotheses. The probabilities of the hypotheses change from the prior value of $P(B_i)$ to the posterior value of $P(B_i|A)$ after we observe the “data,” which is essentially the occurrence of event A (e.g., a positive result of a disease test). The hypothesis is typically something that we are not certain about even after having observed the data (e.g., whether a person truly has a disease). The changed probability $P(B_i|A)$ is a posterior, whereas $P(A|B_i)$ is the likelihood. Although $P(A|B_i)$ and $P(B_i|A)$ seem to have similar mathematical structure, they treat observable and unobservable events differently; the posterior probability $P(B_i|A)$ conditions the unobservable hypothesis on the observed data, whereas the likelihood $P(A|B_i)$ conditions the observed data on the unobservable hypothesis.

There may be infinitely many hypotheses in many statistical scenarios, whereas the data are always finite. For example, the hypotheses may be all the (infinitely many) possible values of continuous parameters in a statistical model (e.g., the values of the mean and variance of a normal distribution), and the likelihood may correspond to the probability of the observed data to be generated, given specific values of the parameters (i.e., given the specific hypothesis being evaluated). Finding parameters that maximize either these likelihoods or the posterior probabilities of the parameters (given the data) helps us find statistical models that are most consistent with observed data. The approach of optimizing likelihoods is referred to as *frequentist statistics*, whereas the approach of optimizing posterior probabilities is referred to as *Bayesian statistics*. These two approaches form the holy grail of much of probabilistic machine learning, and they are introduced in Chapter 6.

3.7 The Basics of Probability Distributions

As discussed earlier in this chapter, random variables map sample spaces to values, which helps enable the notion of *probability distributions* in the form of probability mass functions. For example, consider the case in which we wish to model the random variable corresponding to the sum of the face values resulting from the throws of two dice. In this case, the sample space corresponds to the 36 possible outcomes, and each of these outcomes is mapped to an integer in $[2, 12]$. The probability of each of these outcomes can also be easily quantified by using the counting view of probability (see section 3.3). It is noteworthy that *this random process is a generating mechanism, which implicitly defines a generating distribution of the data*. The process of *sampling* from a distribution means that the generating mechanism of the distribution is executed in order to create an outcome from the sample space, which is then mapped to its observed random variable value. This general principle is also true in machine learning, wherein the observed data is assumed to be an outcome of a simplified generating distribution (cf. Figure 3.1). Much of the machine learning process (shown in red in the figure) is about creating this simplified generating distribution.

The random variables discussed thus far are discrete variables over a finite set of possibilities because they are obtained by mapping sample spaces of finite size to random variable values. As a result, the probability distribution of such random variables can be represented as a table of finite size, which is referred to as its probability mass function. However, this tabular approach to data representation becomes impractical (or even impossible) when the number of possible values of the random variable is very large (or even infinite). The case of an infinite number of possible values of the random variable naturally occurs in cases when it is continuous, and therefore the underlying sample spaces are infinitely large as well. In such cases, it becomes critical to represent probability distributions with the use of *closed-form functions*, where the probability density is an algebraic function of the value of

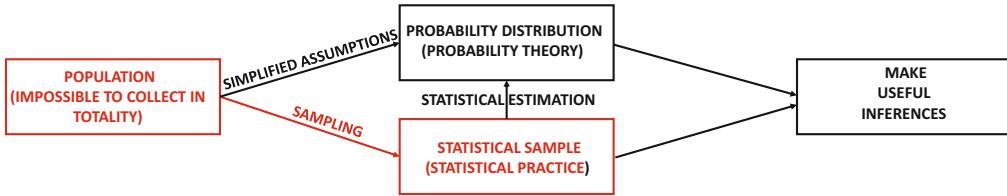


Figure 3.1: Revisiting Figure 1.6: How populations and samples interact with probability and statistics in machine learning — the portion in red corresponds to how data collection is implicitly viewed as the process of sampling from an unknown generating distribution of arbitrary complexity.

Table 3.2: Revisiting Table 1.3: Outcomes for number of successes in 10 die throws and their probabilities

Number of Successes	Probability
0	0.1615
1	0.3230
2	0.2907
3	0.1550
4	0.0543
5	0.0130
6	0.0022
7	0.0002
8	< 0.0001
9	$\ll 0.0001$
10	$\ll 0.0001$

the variable (rather than being represented in a table of value-probability pairs). The use of closed-form representations is not exclusive to continuous random variables, because they are also used in the discrete setting with probability mass functions.

3.7.1 Closed-Form View of Probability Distributions

Consider a discrete random variable denoted by X . A closed-form representation of this probability mass function maps a realized value x of this variable to the algebraic function $p_X(x)$ (instead of a tabular mapping function to probabilities). For example, consider the case where the random variable X corresponds to the number of times the face 2 shows up in 10 die rolls (i.e., number of “successes” in 10 die rolls). In such a case, the tabular form of the probability mass function is shown in Table 3.2. However, a more compact view of this table of 11 rows may be obtained by mapping the value k in the left column of the table to the probability value in the right column of the table with the following closed-form function:

$$\text{Probability of } k \text{ successes} = \frac{10!}{(10-k)!k!} \left(\frac{1}{6}\right)^k \left(\frac{5}{6}\right)^{(10-k)} \quad (3.3)$$

The rationale for the above expression is that there are $\binom{10}{k}$ ways of selecting the indices of the k die rolls (where the face value is 2), and each such option has a probability of $\left(\frac{1}{6}\right)^k \left(\frac{5}{6}\right)^{(10-k)}$. One can easily verify that the probabilities in Table 3.2 satisfy the above expression. This closed-form representation has the advantage of being much more compact

than the unwieldy table with 11 rows. Increasing the number of die rolls increases the size of the table. For example, an experiment with 1000 rolls would have required a table with 1001 rows but it does not change the size of the closed-form representation.

An even more important reason for the usefulness of closed-form functions in machine learning is that *it allows us to work with a parameterized distribution*, and the parameters can be learned from the observed data. This approach allows a certain level of flexibility when one only has a general idea of the shape of the data distribution, but not the specific details that are controlled by the underlying parameters. The general form of the data distribution is obtained by making a hypothesis about the generating mechanism of the observed data. In most cases, we only have access to the observations (like the data samples available in machine learning), but we do not have access to the precise generating mechanism of this data. One might make the simplified assumption that the frequency of a word in a document is generated by a process similar to the counts being generated in die rolls, but we may not know either the number of rolls or the probability of a success. In such cases, one might *model*² a particular word frequency in the following form with the parameters p (probability of success) and n (number of die rolls). Therefore, one may now assume that the instances of the attributes we observe are generated from the following *parameterized* probability distribution, referred to as the *binomial distribution*:

$$\text{Probability of } k \text{ successes} = \frac{n!}{(n-k)!k!} p^k (1-p)^{(n-k)} \quad (3.4)$$

The parameters of the distribution can be estimated by the process of *fitting* observations to distributions (cf. Chapter 6). The modeling process corresponds to the portion marked “simplified assumptions” and the estimation process corresponds to the portion marked “statistical estimation” in Figure 3.1. It is noteworthy that if the modeling process (such as choosing a specific distribution to represent word frequencies) is oversimplified or somehow grossly incorrect at an empirical level, it will be hard to make accurate inferences from the model. The process of making the correct types of simplified assumptions is a soft skill that is gained with experience in different machine learning settings. For example, the counts of lexicon words in a document collection are often modeled in a manner similar to the frequencies of die faces after repeated rolls of a biased die with as many faces as the lexicon size. While it is obvious that the counts of lexicon words are not actually generated by die rolls, this *assumption* provides an excellent way to perform *simplified modeling* of a data distribution whose generating process is unknown to us. In practice, real-world data is always generated from some complex distribution (i.e., the population) that cannot be represented in closed form — nevertheless, the distributions used in machine learning to *model* the population are invariably oversimplifications of this complex distribution. However, even with oversimplifications, a judicious choice of distributions leads to predictions of high quality in downstream applications.

An important property of all probability mass functions is that the probability values must be nonnegative and must sum to 1 over all possible outcomes. Therefore, all probability entries of the PMF table must sum to 1. We encourage the reader to add the probability values in the rows of Table 3.2 and confirm that the probability values of the table add up to almost 0.9999 (with the slight difference being a result of numerical rounding). Furthermore, one can use the known algebraic form of the binomial expansion to show that the closed-form expression for the probability mass function of the number of successes in n die rolls

²In fact, a generalized form of this modeling is used to capture the counts of each lexicon word in each document of a data set containing collections of documents. The underlying distribution is referred to as the *multinomial distribution*.

also sums to 1 over all possible outcomes:

$$\sum_{k=0}^n \text{Probability of successes} = \sum_{k=0}^n \frac{n!}{(n-k)!k!} p^k (1-p)^{(n-k)} = [p + (1-p)]^n = 1$$

It is possible for a discrete distribution to take on one of an infinite number of possible values. Examples include the *Poisson distribution* and the *geometric distribution*. In such cases, the closed-form representation is the only way to represent the PMF of the random variable (because a tabular representation would have an infinite number of rows).

Example 3.12 Consider a PMF of a discrete random variable defined over all natural numbers \mathbb{N} with probability $K/[n(n+1)]$ for some constant K and outcome $n \in \mathbb{N}$. What is the value of K that makes it a valid PMF?

Solution: Since the sum of all probabilities is 1, we have:

$$1 = \sum_{n \in \mathbb{N}} \frac{K}{n(n+1)} = K \sum_{n \in \mathbb{N}} \left[\frac{1}{n} - \frac{1}{n+1} \right] = K$$

In other words, the value of K should be 1. ■

Problem 3.14 Suppose you toss a coin repeatedly until it first turns up tails and then define a random variable that counts the total number of tosses until termination of the experiment (including the terminating toss of tails). Describe the sample space in terms of a sequence of ‘H’ (for heads) and ‘T’ (for tails). What are the sample-space outcomes, corresponding random variable values, and their probabilities? Show that the sum of the probabilities of all possible outcomes of the random variable is 1.

3.7.2 Continuous Distributions

In cases where the random variable X is continuous, a slightly different form of the probability distribution function is used, which is referred to as a *probability density function* $f_X(x)$, instead of a probability mass function. Since the number of possible outcomes of the random variable is infinite, the probability of any specific point is infinitesimally small, and it only makes sense to speak of ranges of values when quantifying probability values. For individual points, one typically talks about probability *densities* rather than probabilities. A probability density function is defined somewhat differently from a probability mass function in that it measures the “density” of probabilities at a given point rather than the actual probability. Just as physical densities are integrated over regions to obtain a physical mass, probability densities are integrated over regions to obtain the probability mass. Therefore, we define the notion of a probability density function as follows:

Definition 3.16 (Probability Density Function) A probability density function $f_X(x) : \mathbb{R} \rightarrow \mathbb{R}^+$ of a random variable X is a nonnegative function satisfying the following property for any real-valued range $[a, b]$:

$$P(a \leq X \leq b) = \int_{x=a}^b f_X(x) dx$$

Setting $a = -\infty$ leads to a special form of the above integral, referred to as the *cumulative distribution function* $F_X(b)$:

$$F_X(b) = P(X \leq b) = \int_{x=-\infty}^b f_X(x)dx$$

One can derive a similar expression for discrete, numeric random variables by replacing the integral with a summation:

$$F_X(b) = P(X \leq b) = \sum_{x=-\infty}^b p_X(x)dx$$

For discrete distributions, it is important to note that $F_X(b)$ includes the probability of the random variable taking on the probability of exactly b . The probability of the random variable taking on a specific value(such as b) is no longer infinitesimally small in discrete distributions. Since the probability of the random variable taking on at least one outcome in the range $(-\infty, \infty)$ is 1, it implies the following:

$$F_X(\infty) = \int_{x=-\infty}^{\infty} f_X(x)dx = 1$$

Note that the cumulative distribution function is monotonically non-decreasing, and takes on the values of 0 and 1 at $x = -\infty$ and $x = +\infty$, respectively. One can also derive the probability density function from the cumulative distribution function. The algebraic expression for the cumulative distribution function can be differentiated to obtain the expression for the probability density function:

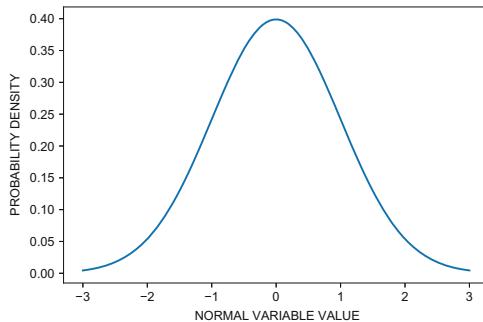
$$f_X(x) = \frac{d}{dx} F_X(x)$$

The relationship between the probability density function and the cumulative distribution function is very useful because certain probabilistic settings allow easy derivation of the expression for the cumulative distribution function, which can be interpreted directly as probability values (as compared to the somewhat less intuitive interpretation of density functions). Direct interpretation of the cumulative distribution function as a probability value allows natural use of all the machinery and laws of probabilities introduced in this chapter. On the other hand, the application of such laws to density functions may not be quite as intuitive, although many laws such as the Bayes rule continue to apply to them as well.

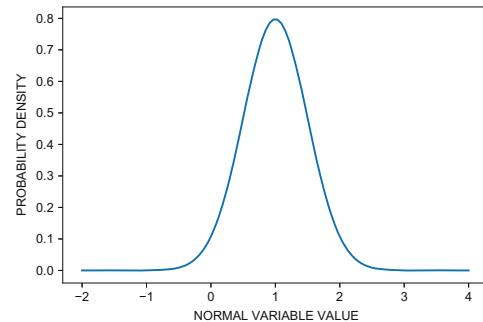
An example of the probability density function of a well-known continuous distribution is that of the normal distribution. It has two parameters that correspond to its mean and variance. The probability density function of the normal distribution with mean μ and variance σ^2 is defined as follows:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (3.5)$$

The probability density function of the normal distribution has the shape of a bell curve, which is centered at μ and has a standard deviation of σ . Although the basic shape of the distribution is similar across all values of μ and σ , the specific choice of the parameters defines the placement and the width of the bell curve. Two examples of the probability



(a) Mean of 0 and variance of 1



(b) Mean of 1 and variance of 0.25

Figure 3.2: Revisiting Figure 1.4: The probability density functions for two examples from the family of normal distributions

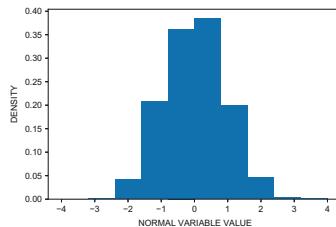
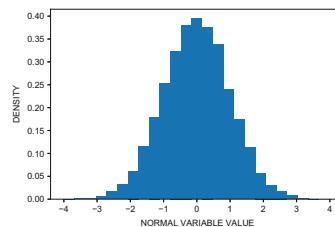
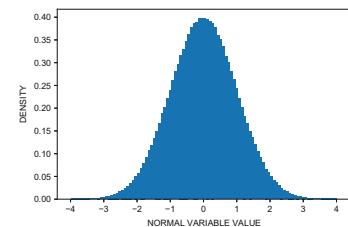
(a) 10 bins, 10^3 points(b) 25 bins, 10^4 points(c) 100 bins, 10^6 points

Figure 3.3: Histograms are empirical representations of probability density functions: An illustration with data generated from a normal distribution.

density function of the normal distribution with different means and variances are shown in Figure 3.2. It is noteworthy that the shapes of the two distributions of Figure 3.2(a) and 3.2(b) are very similar, except for the placement of the center of the distribution and its level of dispersion (spread). The distribution of Figure 3.2(a) with mean 0 and variance of 1 is special, and it is referred to as a *standard normal distribution*. Data are often preprocessed (or *normalized*) by shifting its center and scaling it in order to have a mean of 0 and standard deviation of 1.

It is noteworthy that histograms can be used to empirically approximate probability density functions, when a sufficient amount of data is available to robustly represent the frequencies in a large number of bins. In other words, if we generate a very large number of points from a particular distribution $f_X(x)$, its density histogram (using a sufficient number of bins) will have a smooth shape that closely approximates $f_X(x)$. In order to illustrate this point, histograms with increasing granularity (i.e., increasing numbers of bins and points) are generated from a standard normal distribution and shown in the different subplots of Figure 3.3. Increasing the number of points also allows the use of a larger number of bins, resulting in a more fine-grained and smooth histogram. It is evident that increasing the granularity of the histograms makes the shape of the histograms closer and closer to the true probability density function of a standard normal distribution of Figure 3.2(a). In fact, *histograms can be viewed as the empirical representations of probability density functions or probability mass functions*.

Example 3.13 Consider the continuous random variable defined over all $x \geq 1$ in which the cumulative distribution function is defined as follows:

$$F_X(x) = 1 - 1/x^2 \quad \text{for } x \geq 1$$

Compute the algebraic expression for the probability density function of this continuous random variable.

Solution: The density function is obtained by differentiating the cumulative distribution function as follows for $x \geq 1$:

$$\begin{aligned} f_X(x) &= \frac{d}{dx} F_X(x) \\ &= 2/x^3 \end{aligned}$$

The above density function is defined for $x \geq 1$, and the value is 0, otherwise. ■

Example 3.14 Consider the discrete random variable defined over all natural numbers in which the cumulative distribution function is as follows:

$$F_X(x) = 1 - 1/x^2 \quad \forall x \in \mathbb{N}$$

Compute the algebraic expression for the probability mass function of this discrete random variable.

Solution: For $x = 1$, we have $p_X(1) = F_X(1) = 1 - 1/1^2 = 0$. Therefore, the value of $p_X(x)$ is nonzero only for natural numbers greater than 1 as follows:

$$p_X(x) = F_X(x) - F_X(x-1) = \frac{1}{(x-1)^2} - \frac{1}{x^2} = \frac{2x-1}{x^2(x-1)^2}$$

■

Problem 3.15 Let $f_X(x)$ be a probability density function taking on the value Kx^3 in the range $x \in [0, 1]$, and 0, otherwise. Find the value of K so that $f_X(x)$ is a valid probability density function.

A hint for solving this problem is that the total probability over all possible outcomes of X must be 1.

3.7.3 Multivariate Probability Distributions

The probability distributions introduced thus far involve a single variable, and are therefore *univariate probability distributions*. However, most machine learning applications work with data sets involving multiple variables, which can be viewed as outcomes of sampling from multivariate distributions. Multivariate probability distributions are also referred to as *joint distributions*. Although the probability mass function of the joint distribution of a discrete random variable can also be expressed in tabular form, it becomes increasingly cumbersome to maintain such a table when the number of possible values of the random variable is large

— this is particularly true in the multivariate case because of the multiplicative effect of substituting different combinations of attribute values in the PMF table. Therefore, the probability mass functions and the probability density functions of multivariate distributions are almost always expressed in closed form.

The binomial distribution discussed earlier in the chapter can be generalized to the multivariate case. An example of a binomial distribution is one in which the number of times a face with a value of 2 shows up in n throws of the die. The corresponding multivariate generalization, which is the *multinomial distribution*, creates a 6-dimensional random variable vector \vec{X} containing the number of times the six faces of the die show up in n throws. Therefore, we want to compute the probability that the frequency of the six faces is given by $\vec{k} = [k_1, k_2, \dots, k_6]$ for a set of n throws. Furthermore, assume that the probabilities of the faces showing up are given by $p_1 \dots p_6$, so that the die is biased. First, note that this probability value is 0 in the event that $\sum_{i=1}^6 k_i \neq n$ or any k_i is negative. This is because the number of outcomes over the six faces must sum to n and a frequency can never be negative. In the event that the condition $\sum_{i=1}^6 k_i = n$ is satisfied, the joint probability mass function is given by the following expression:

$$p_{\vec{X}}(k_1, k_2, k_3, k_4, k_5, k_6) = \frac{n!}{k_1!k_2!k_3!k_4!k_5!k_6!} p_1^{k_1} p_2^{k_2} p_3^{k_3} p_4^{k_4} p_5^{k_5} p_6^{k_6} \quad [\text{When } \sum_{i=1}^6 k_i = n]$$

This probability mass function can be shown to add to 1 over all possible values of its argument. Interestingly, this distribution has an important application in machine learning — it is used to model the joint probabilities of various words in a given document (belonging to a particular topic) by using a biased topic-specific die (with as many faces as the number of words in the lexicon) to generate the frequencies of words in documents belonging to that topic. Although it is evident that the frequencies of words in documents are not actually generated by throwing dice, the probabilistic model provides excellent results in many predictive applications. This is an example of a simplified generating process used to explain observed data in machine learning models. It is precisely this type of simplified modeling skill that is useful for the statistician to develop in order to solve machine learning problems.

Just like the multivariate probability mass function, one can also define a multivariate normal density function (also referred to as the *Gaussian distribution*) as follows:

$$f_{\vec{X}}(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\sum_{i=1}^2 \frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right) \quad (3.6)$$

Here, μ_i and σ_i^2 are respectively assumed to be the mean and variance along the i th dimension. It can be shown that the joint probability sums to 1 over all possible values of the random variable vector \vec{X} . Since the random variable is drawn from a continuous distribution, this summation is expressed as an integral as follows:

$$\int_{x_1=-\infty}^{\infty} \int_{x_2=-\infty}^{\infty} f_{\vec{X}}(x_1, x_2) dx_1 dx_2 = 1$$

The probability density function of a standard 2-dimensional normal (Gaussian) distribution is shown in Figure 3.4. As in all standard normal distributions, each mean μ_i is set to 0 and each variance σ_i^2 is set to 1 (cf. Equation 3.6).

Given a probability density function (or probability mass function), it is possible to derive the *marginal distributions* along individual variables by integrating (or summing) over

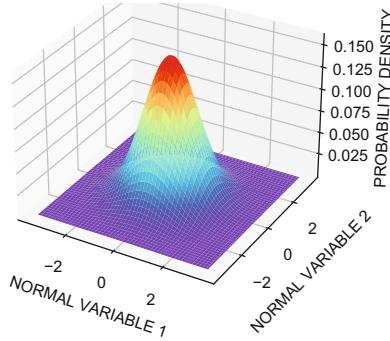


Figure 3.4: The probability density function for a 2-dimensional Gaussian distribution with mean of 0 and variance of 1 along each direction

all outcomes of the other variables. In the two-dimensional case, the marginal distributions along the first variable X_1 can be computed as follows:

$$p_{X_1}(x_1) = \sum_{x_2} p_{\vec{X}}(x_1, x_2)$$

$$f_{X_1}(x_1) = \int_{x_2} f_{\vec{X}}(x_1, x_2) dx_2$$

In the case of the Gaussian distribution above, each 1-dimensional marginal distribution can be shown to be the normal distribution. The above 2-dimensional example can be easily generalized to the d -dimensional case by summing (or integrating) over the remaining $(d - 1)$ dimensions. In the case of the multinomial distribution above, the marginal distribution along x_1 may be obtained by simultaneously summing along x_2, x_3, \dots, x_6 . The corresponding 1-dimensional marginal distribution can be shown to be the same as the binomial distribution, because it shows the distribution for a single face rather than all six faces.

Example 3.15 Let $f_{\vec{X}}(x_1, x_2) = K(x_1 + x_2)$ be a joint probability density function for $x_1, x_2 \in (0, 1)$ (and 0 otherwise). For what value of K is this a valid density function for the random variable vector $\vec{X} = [X_1, X_2]$? Derive the 1-dimensional marginal density function in x_1 .

Solution: For this density function to be valid, it must integrate to 1 over its defined range. Therefore, we have the following:

$$\int_{x_1=0}^1 \int_{x_2=0}^1 K(x_1 + x_2) dx_1 dx_2 = 1$$

The integral expression above on the left-hand side evaluates to K . Therefore, the value of K must be 1.

The expression for the marginal density function is as follows:

$$f_{X_1}(x_1) = \int_{x_2=0}^1 (x_1 + x_2) dx_2$$

The integral expression above evaluates to $(x_1 x_2 + x_2^2/2)$, when evaluated as an indefinite integral. On substituting the bounds of the definite integral, the expression evaluates to $(x_1 + 0.5)$. Therefore, the marginal density function is defined over $(0, 1)$, and is expressed as follows:

$$f_{X_1}(x_1) = x_1 + 0.5$$

■

Problem 3.16 Let $f_{\vec{X}}(x_1, x_2) = Kx_1x_2$ be a joint probability density function for $x_1, x_2 \in (0, 1)$ (and 0 otherwise). For what value of K is this a valid density function for the random variable vector $\vec{X} = [X_1, X_2]$? Derive the 1-dimensional marginal density function in x_1 .

A hint for solving the above problem is that the total probability of the density function over the entire sample space must always be 1.

Problem 3.17 Let $f_{\vec{X}}(x_1, x_2) = K(x_1^2 + x_2^2)$ be a probability density function for $x_1, x_2 \in (0, 1)$ (and 0 otherwise). For what value of K is this a valid density function for the random variable vector $\vec{X} = [X_1, X_2]$? Derive the 1-dimensional marginal density function in x_1 . What is the probability that $X_1 \leq 0.5$?

3.8 Distribution Independence and Conditionals

The discussion of independence and conditional probabilities has so far been restricted to events and sample spaces. This section generalizes the concept of independence to random variables.

3.8.1 Independence of Distributions

When two events are independent, the probability that both occur is a product of their probabilities. Since random variables are mappings from sample spaces to values, a specific value of a random variable is an event — therefore, the product rule of independence naturally generalizes from event spaces to random variables. In other words, if X_1 and X_2 are independent random variables, the following must hold:

$$P([X_1 = x_1] \cap [X_2 = x_2]) = P(X_1 = x_1) \cdot P(X_2 = x_2)$$

One can impose the product rule of independence even in cases where the random variables are defined using distributions. The independence result for random variable distributions is stated below:

Definition 3.17 (Independence of Random Variable Distributions) Let $\vec{X} = [X_1, \dots, X_d]$ be a d -dimensional random variable with joint PMF $p_{\vec{X}}(\vec{x})$ (or probability density function $f_{\vec{X}}(\vec{x})$ if it is continuous). Then, the random variables X_1, X_2, \dots, X_d are independent if and only if the joint distribution can be expressed as a product of the marginal distributions:

$$p_{\vec{X}}(\vec{x}) = \prod_{i=1}^d p_{X_i}(x_i) \quad [\text{Discrete Random Variables}]$$

$$f_{\vec{X}}(\vec{x}) = \prod_{i=1}^d f_{X_i}(x_i) \quad [\text{Continuous Random Variables}]$$

It is also possible to combine a probability density function and a probability mass function in order to model mixed-attribute data by using the product of the probability mass function and the density function. In such cases, the joint distribution has the properties of both a probability density function and a probability mass function, depending on which attribute is being considered.

The notion of independence is extremely useful in order to simplify the representation of a random variable. When the joint distribution can be expressed as a product of marginal distributions, the number of parameters required to represent the distribution is fewer, which allows for a more compact representation. Such compact representations simplify the learning process in machine learning applications.

Problem 3.18 Are the random variables X_1 and X_2 in each of Problems 3.16 and 3.17 independent? Give a justification for each answer.

3.8.2 Conditional Distributions

Distributions are often conditioned on specific events. For example, if A is the event that a particular salmon belongs to the freshwater category, the probability distribution of the length of salmon (conditional on its being a freshwater salmon) is denoted by $f_{X|A}(x)$. This conditional distribution is different from the unconditional distribution $f_X(x)$ of the length of all salmon. In most cases, $f_{X|A}(x)$ will have a much simpler distribution than $f_X(x)$, because $f_{X|A}(x)$ represents a homogeneous class of fish. This is an important consideration in data-driven modeling. For many real-world data sets, it is inconvenient to directly formulate the unconditional distribution $f_X(x)$. On the other hand, it is easier to view the distribution as an amalgamation of different simpler distributions, each of which has a conditional density function. The unconditional distribution can be mathematically derived in terms of conditional distributions using the total probability rule. One can also define conditional probability mass functions of discrete data and conditional density functions of continuous data in a similar manner as follows:

Definition 3.18 (Conditional Distribution) The conditional probability mass function $p_{X|A}(x)$ of a discrete distribution is given by the following:

$$p_{X|A}(x) = \frac{P((X = x) \cap A)}{P(A)}$$

For continuous distributions, the conditional cumulative distribution function $F_{X|A}(x)$ is defined as follows:

$$F_{X|A}(x) = \frac{P((X \leq x) \cap A)}{P(A)}$$

The conditional probability density function $f_{X|A}(x)$ is given by the derivative of the cumulative distribution function $F_{X|A}(x)$ with respect to x .

It is helpful to understand the concept of a conditional distribution with the help of an example. Consider a situation where the salmon sold in a particular market are either freshwater fish (70%) or seawater fish (30%). Since the two types of fish develop in completely different environments (i.e., different generating processes), it is reasonable to assume that their lengths are drawn from different probability distributions as well. For example, freshwater salmon might have a mean length of 27 inches and standard deviation of 1 inch, whereas seawater salmon may have a mean length of 30 inches and a standard deviation of 1 inch.

Let A denote the event that a sampled salmon from the market is a freshwater fish. Since 70% of the salmon in the market are freshwater fish, we have $P(A) = 0.7$. A *modeling assumption* is that the conditional probability density $f_{X|A}(x)$ of the length of fresh-water and the conditional probability density $f_{X|A^c}(x)$ of the length of sea-water salmon are drawn from a normal distribution:

$$\begin{aligned} f_{X|A}(x) &= \frac{1}{\sqrt{2\pi}} \exp(-(x - 27)^2/2) \quad [\text{Freshwater}] \\ f_{X|A^c}(x) &= \frac{1}{\sqrt{2\pi}} \exp(-(x - 30)^2/2) \quad [\text{Seawater}] \end{aligned}$$

How does one characterize the unconditional distribution of all salmon? It turns out that the total probability rule can be generalized to distributions by treating each value of the variable as an event:

Lemma 3.6 (Total Probability Rule for Distributions) *Let $f_{X|A}(x)$ and $f_{X|A^c}(x)$ be the conditional density functions with respect to an event A and its complement. Then, the unconditional density function $f_X(x)$ is given by the following:*

$$f_X(x) = P(A) \cdot f_{X|A}(x) + P(A^c) \cdot f_{X|A^c}(x)$$

The result can also be generalized to discrete random variables by substituting the probability density functions in the above equation with probability mass functions:

$$p_X(x) = P(A) \cdot p_{X|A}(x) + P(A^c) \cdot p_{X|A^c}(x)$$

In the case of the salmon example discussed above, the length of all fish will be given by the following distribution:

$$f_X(x) = 0.7 * \frac{\exp(-(x - 27)^2/2)}{\sqrt{2\pi}} + 0.3 * \frac{\exp(-(x - 30)^2/2)}{\sqrt{2\pi}} \quad [\text{All fish}]$$

The distributions of freshwater fish, seawater fish, and all fish are shown in Figure 3.5. This particular form of the unconditional distribution (as a weighted combination of conditional distributions) is referred to as a *mixture model*, and it is ubiquitous for modeling observed data in probability, statistics, and machine learning. A significant amount of focus will be placed on such models in subsequent chapters.

Just as the total probability rule can be generalized to distributions, the Bayes rule can also be generalized to probability distributions:

Lemma 3.7 (Bayes Rule for Distributions) *Let $f_{X|A}(x)$ and $f_{X|A^c}(x)$ be the conditional density functions with respect to an event A and its complement. Then, the probability of event A , given the observed value of the variable $X = x$ is given by the following:*

$$P(A|X = x) = \frac{P(A) \cdot f_{X|A}(x)}{P(A) \cdot f_{X|A}(x) + P(A^c) \cdot f_{X|A^c}(x)}$$

The result can also be generalized to discrete random variables by substituting the conditional probability density functions in the above equation with conditional probability mass functions:

$$P(A|X = x) = \frac{P(A) \cdot p_{X|A}(x)}{P(A) \cdot p_{X|A}(x) + P(A^c) \cdot p_{X|A^c}(x)}$$

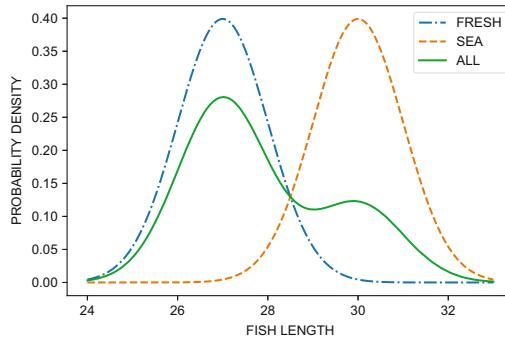


Figure 3.5: The conditional and unconditional distributions for fish length

Armed with this distribution-centric version of the Bayes rule, we are now ready to set up a simple knowledge-based Bayes classifier.

The results shown for the total probability rule and the Bayes rule are presented using 1-dimensional distributions. However, they can also be used for multidimensional data by replacing the 1-dimensional distributions with joint distributions. However, it is rare to directly use joint distributions without some form of simplification — the most common approach is to apply *conditional independence* during modeling. A joint distribution exhibits conditional independence with respect to event A , if the conditional joint distribution is expressed as the product of conditional univariate distributions:

$$p_{\vec{X}|A}(\vec{x}) = \prod_{i=1}^d p_{X_i|A}(x_i) \quad [\text{Discrete Random Variables}]$$

$$f_{\vec{X}|A}(\vec{x}) = \prod_{i=1}^d f_{X_i|A}(x_i) \quad [\text{Continuous Random Variables}]$$

It is also possible to use the product of probability mass functions and probability density functions for mixed-attribute data containing both discrete and continuous variables.

The intuition behind conditional independence is to assume that different parts of the data are created by different generating processes, depending on the outcome of event A . Each of these generating processes is relatively simplified, but combining them creates a realistic distribution. For example, although the attributes of the data set in Figure 3.6 are clearly related on a *global* basis, they are independent on a *local* basis (i.e., within each cluster of the data set). Each of these clusters is generated by a joint distribution of the type shown above. As we will see in the next chapter, this approach is used in *mixture models*, which are among the most widely used distributions in machine learning.

3.8.3 Example: A Simple 1-Dimensional Knowledge-Based Bayes Classifier

The Bayes classifier is a popular method in machine learning that first learns the parameters of a mixture model from a labeled data set and then uses it for predictions into classes. For example, the feature variable might be the length of a salmon and the class label might be a tag indicating whether that observation is a freshwater salmon or a seawater salmon. It is assumed that the lengths of each of these different types of fish are drawn from a normal

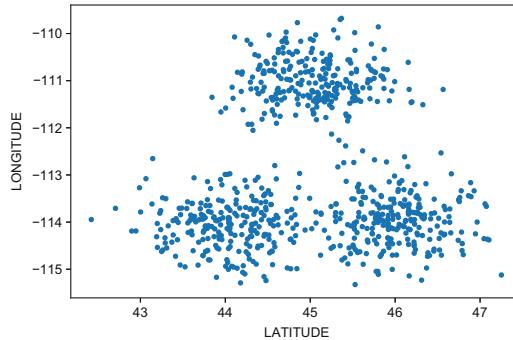


Figure 3.6: The two attributes are conditionally independent in local regions (clusters) of the data set.

distribution whose parameters are estimated from the observed data. Since this type of statistical estimation has not yet been introduced (cf. Chapter 6), we will work with a *knowledge-based simplification* of this setting, where a domain expert is available to tell us the mean and variance of the two distributions.

Example 3.16 (Bayes Classification with Domain Knowledge) *The salmon available in a given market are 70% freshwater and 30% seawater fish. You decide to model their lengths using different normal distributions. The local fish expert, Mr. Know-It-All, tells you that the lengths of freshwater salmon are normally distributed (cf. Equation 1.3 of Chapter 1) with a mean of 27 inches and standard deviation of 1 inch, whereas the lengths of seawater salmon are normally distributed with a mean of 30 inches and standard deviation of 1 inch. You randomly select a salmon from the market and find it to be 29 inches long. What is the probability that the selected salmon is a freshwater fish?*

Solution: Let A be the event that the selected salmon is a freshwater fish. We would like to find $P(A|X = 29)$. Using the Bayes rule, we obtain the following:

$$P(A|X = 29) = \frac{P(A) \cdot f_{X|A}(29)}{P(A) \cdot f_{X|A}(29) + P(A^c) \cdot f_{X|A^c}(29)}$$

on substituting the expressions for the two Gaussian distributions (while ignoring constant terms like $1/\sqrt{2\pi}$ in the numerator and denominator), one obtains the following:

$$P(A|X = 29) = \frac{0.7 \cdot \exp(-2)}{0.7 \cdot \exp(-2) + 0.3 \cdot \exp(-0.5)} \approx 0.34$$

In other words, the probability that it is a freshwater fish is 0.34. Note that freshwater fish have a higher prior probability of 0.7 to be selected; however, since the selected fish is somewhat on the longer side at 29 inches, its posterior probability of being a freshwater fish (given its length) is only 0.34. ■

The methodology discussed in the aforementioned example is the prediction portion of a Bayes classifier, which is popularly used in machine learning. However, it is not assumed in machine learning that a domain expert is available to tell us about the parameters (e.g.,

means and standard deviations) of the normal distributions of freshwater fish and seawater fish. These parameters are estimated in a data-driven manner using a process referred to as *maximum likelihood estimation* (cf. Chapter 6). The analyst does have the burden of deciding the family of distributions (e.g., normal distribution) that is used for modeling.

Problem 3.19 *Aside from the length distribution information already provided by your local fish expert (see Example 3.16), the precise color of the (light pink or dark pink) salmon provides additional clues about its origin. Mr. Know-It-All tells you that freshwater salmon are dark pink in color with probability 0.3, whereas seawater salmon are dark pink with probability 0.8. Furthermore, for any particular type of salmon (i.e., freshwater or seawater fish), the color and length are independent attributes. You find that your selected salmon (of length 29 inches) is dark pink in color. What is the probability that it is a freshwater salmon?*

The above problem is exactly similar to Example 3.16), except that joint distributions must be used by the Bayes classifier.

Problem 3.20 *The time in hours to the next bus at stop A follows the probability density function $f_X(x) = \exp(-x)$ and the time in hours to the next bus at stop B follows the probability density function $f_X(x) = 2\exp(-2x)$. Mary selects one of the two stops randomly, but she goes to stop A twice as often as she goes to stop B. Compute the distribution of Mary's waiting time on a randomly selected day. If Mary waited for an hour on a particular day, what is the probability that she was waiting at stop A.*

3.9 Summarizing Distributions

Section 2.2 of Chapter 2 discusses the computation of summary statistics from *observed data samples*. Examples of such summary statistics include the sample mean and the sample variance. This section discusses the computation of exactly the same types of summary statistics from *probability distributions*. How are the two related? If probability distributions are used to generate a very large number of data samples, the summary statistics of the data samples will converge to the corresponding expected values for the probability distributions. For example, the sample mean of a large number of generated points will become closer and closer to the distribution mean, as more and more points are generating. The two are the same in the limiting case, as the number of points goes to infinity. This is the reason that the summary statistics of probability distributions are referred to as *expected values*; they reflect the sample statistics in the limiting case.

The results of this section apply only to numeric attributes (rather than categorical attributes). Furthermore, the general assumption is that the numeric attributes are continuous (with probability density functions) rather than discrete (with probability mass functions). Throughout this section, the results will be presented for continuous random variables rather than discrete (numeric) random variables for greater generality. *The results can be generalized easily to discrete (numeric) random variables by replacing integrals with summations (and probability density functions with probability mass functions).*

3.9.1 Expectation and Variance

As in the case of the sample statistics discussed in section 2.2, the two most common forms of univariate summarization include the generation of measures of central tendency and measures of dispersion. The distribution-centric analog to the sample mean is the expected

value (or distribution mean) $E[X]$ of random variable X . It is also common to denote $E[X]$ by μ_X (without the circumflex on top because it is an expected value rather than an estimated sample mean). The circumflex is generally indicative of estimated (or predicted) values.

Definition 3.19 (Distribution Mean or Expected Value) *Given a continuous random variable X with probability density function $f_X(x)$, its expected value $\mu_X = E[X]$ is defined as follows:*

$$E[X] = \int_{x=-\infty}^{\infty} x \cdot f_X(x) dx$$

One can also rewrite the above expression for discrete (numeric) random variables by combining probability mass functions with summations:

$$E[X] = \sum_x x \cdot p_X(x)$$

Note that if one samples x from the distribution $f_X(x)$ a very large number of times, the sample mean $\hat{\mu}_X^{(n)}$ of n points will converge to the distribution mean μ_X with an increasing number of sampled points (which is what makes the distribution mean an expected value):

$$\mu_X = \lim_{n \rightarrow \infty} \hat{\mu}_X^{(n)}$$

One can define the expected value of any function $G(X)$ of the random variable in an analogous way, because it defines the mean value of the function when applied to a large number of samples drawn from the distribution:

Definition 3.20 (Expected Value of Function) *Given a continuous random variable X with probability density function $f_X(x)$, the expected value of the function $G(X)$ is defined as follows:*

$$E[G(X)] = \int_{x=-\infty}^{\infty} G(x) \cdot f_X(x) dx$$

For discrete random variables with probability mass function $p_X(x)$, the expected value of the function is defined as follows:

$$E[G(X)] = \sum_x G(x) \cdot p_X(x)$$

An important observation is that the expectation of a function of a random variable is *not* (always) equal to that function being applied to the variable expectation:

$$E[G(X)] \neq G(E[X])$$

As an example, consider the case where $G(X) = X^2$, and the random variable X is uniformly distributed in $[0, 1]$ with $E[X] = (1/2)$. In such a case we have the following:

$$\begin{aligned} E[G(x)] &= \int_{x=0}^1 x^2 dx = 1/3 \\ G(E[X]) &= (1/2)^2 = 1/4 \end{aligned}$$

As shown in Chapter 11, Jensen's inequality implies that $E[G(X)] \geq G(E[X])$ for convex functions, and $E[G(x)] \leq G(E[X])$ for concave functions.

A notable case where $E[G(X)] = G(E[X])$ does indeed hold is one in which the function $G(\cdot)$ is³ affine (i.e., applies scaling and translation to X):

³It is noteworthy that affine functions are both convex and concave.

Lemma 3.8 (Expectation under Scaling and Translation) *Let $Y = a \cdot X + b$ be an affine transformation of the random variable X for constants a and b . Then, the expected values of X and Y are related by the same affine transformation:*

$$E[Y] = a \cdot E[X] + b$$

The above result is easy to show by expressing the expectation in terms of the integral and then expanding the result.

How does one compute the expectation of individual random variables from joint distributions? In such cases, one can first compute the marginal density function on the individual random variable and then use the above approach for computing the expectation. Indeed, all the univariate summaries discussed in this section can be generated by first computing the marginal density function. However, it is a different matter when one wants to compute the expectation of a function of multiple variables. In such cases, computing the marginal distribution first is not helpful. If a function of multiple random variables is given, one needs to integrate the function over all outcomes of the joint distribution in order to compute its expectation. For example, the expectation of the function $H(\vec{X})$ with joint distribution $f_{\vec{X}}(\vec{x})$ over all outcomes $\vec{x} = [x_1, \dots, x_d]$ can be computed as follows:

$$E[H(\vec{X})] = \int_{x_1} \int_{x_2} \dots \int_{x_d} H(\vec{x}) \cdot f_{\vec{X}}(\vec{x}) dx_1 dx_2 \dots dx_d$$

In general, the computation of the joint integral may not be possible in closed form and may require computational tricks for evaluation. Some multivariate functions have special characteristics that enable the computation of the expectation easily. In particular, the expectation of the sum of two random variables is given by the sum of their expectations:

Lemma 3.9 *Let X and Y be two random variables. Then, we have $E[X+Y] = E[X]+E[Y]$.*

The above lemma can be easily shown by using the definition of expectation.

Problem 3.21 *Let $\vec{X} = [X_1, X_2]$ have the joint density function $f_{\vec{X}}(x_1, x_2) = (x_1 + x_2)$, which takes on this functional form only for $x_1, x_2 \in (0, 1)$ (and 0 otherwise). Show that $E[X_1 X_2] \neq E[X_1]E[X_2]$.*

The above result can be easily shown from first principles by computing the expectations on both the left-hand side and the right-hand side as integrals.

The *conditional expectation* of a random variable is computed in the same way as the expectation, except that the joint distribution is used in the computation process:

Definition 3.21 (Conditional Expectation) *Given a continuous random variable X with conditional probability density function $f_{X|A}(x)$, its conditional expectation (or conditionally expected value) $E[X|A]$ is defined as follows:*

$$E[X|A] = \int_{x=-\infty}^{\infty} x \cdot f_{X|A}(x) dx$$

One can also compute the conditional expectation over a discrete, numerical random variable by rewriting the above expression with the probability mass function (instead of the probability density function) and using a summation instead of an integral:

$$E[X|A] = \sum_{x=-\infty}^{\infty} x \cdot p_{X|A}(x) dx$$

Example 3.17 Suppose a die is thrown and X is the value of the face. Let A be the event that the outcome is an odd face value. Compute $E[X]$ and $E[X|A]$.

Solution: The expected value of the face is obtained by multiplying each face value by $1/6$ and adding. Therefore, we have $E[X] = (1+2+3+4+5+6)/6 = 21/6 = 3.5$. In the case of conditioning on event A , only the faces 1, 3, and 5 can occur with equal probability of $1/3$. Therefore, the conditional expectation is $E[X|A] = (1+3+5)/3 = 3$. ■

Problem 3.22 Gambler A pays \$5 to play the following game. He tosses a fair die 10 times and is paid an amount in dollars equal to the square of the number of times the face 2 shows up. See Table 3.2 for the probability mass function of this distribution (approximating the last three entries to be 0). What are the expected winnings of gambler A (including the cost of playing)? Should he expect to make a profit by gambling?

Some measures of central tendency are more easily defined with the use of the *cumulative distribution function* rather than the probability density function. Recall that the cumulative distribution function $F_X(x)$ of the random variable X is equal to the fraction of the distribution area that is less than or equal to x :

$$F_X(x) = \int_{z=-\infty}^x f_X(z)dz$$

One can derive a similar expression for discrete numeric random variables by replacing the integral with the summation of the probability mass function:

$$F_X(x) = \sum_{z=-\infty}^x p_X(z)$$

Note that the upper-limit x is included in the summation, which is consequential in the context of discrete functions.

Next, we focus on the concept of distribution median in which the goal is to identify a value that is expected to lie in the middle of a large number of generated points from the distribution.

Definition 3.22 (Continuous Distribution Median) The median of a continuous random variable X with cumulative distribution function $F_X(x)$ is the value a at which $F_X(a) = 0.5$.

In other words $P(X \leq a) = 0.5$, and therefore half the samples of X are expected to be less than a . As in the case of the mean, the sample median converges to the distribution median with an increasing number of sampled points.

The notion of a median can be generalized to that of percentiles in the data, because the median is simply the 50th percentile:

Definition 3.23 (Continuous Distribution Percentile) The p th percentile of a continuous random variable X with cumulative distribution function $F_X(x)$ is the value a at which $F_X(a) = p/100$.

The aforementioned definitions are for continuous random variables. What about discrete random variables? For example, consider the case in which a random variable takes on the

value of 1 with probability 0.2, a value of 2 with probability 0.5, and a value of 3 with probability 0.3. In such a case, the median is 2, but there is no discrete value of a for which $F_X(a) = 0.5$. Furthermore, consider the case of a die roll in which the cumulative distribution function value $F_X(3)$ is 0.5 for $x = 3$. However, a die face of 4 is symmetrically related to 3 in the sample space — in other words, the median is the average of 3 and 4, which is 3.5. Therefore, the median of a discrete distribution needs to be defined more carefully while taking into account the case where $F_X(a)$ is exactly 0.5.

Definition 3.24 (Discrete Distribution Median) *Let X be a discrete random variable with cumulative distribution function $F_X(x)$. If a value of a exists so that $F_X(a)$ is exactly equal to 0.5, then the average of a and its next higher discrete value with nonzero probability is the distribution median. Otherwise, the median is the smallest value of a so that $F_X(a) > 0.5$.*

The percentiles can also be defined in a similar manner. As in the case of sample quartiles, *first quartile* corresponds to a percentile value of 25%, whereas the *third quartile* corresponds to a percentile value of 75%. The difference between the third quartile and the first quartile is a nonnegative value, referred to as the *interquartile range*:

Definition 3.25 (Distribution Interquartile Range) *The interquartile range of a distribution is a nonnegative value that is the difference between the third quartile and the first quartile of the distribution.*

This interquartile range is, therefore, is the length of the interval containing the middle 50% of the distribution area.

Example 3.18 Consider the density function $f_X(x) = 1/(2\sqrt{x})$, which is defined over $(0, 1)$. The density is 0 otherwise. Compute the distribution median, quartiles, and interquartile range.

Solution: The cumulative distribution function is as follows for $z \in (0, 1)$:

$$F_X(x) = \int_{z=0}^x f_X(z)dz = \frac{1}{2} \int_{z=0}^x \frac{1}{\sqrt{z}}dz = \sqrt{x}$$

One can confirm that the cumulative distribution function takes on the value of 1 at $x = 1$. Therefore, the distribution median is obtained by setting $\sqrt{x} = 0.5$, which yields a median of $0.5^2 = 0.25$. The two distribution quartiles are similarly $0.25^2 = 0.0625$ and $0.75^2 = 0.5625$. The interquartile range is $0.5625 - 0.0625 = 0.5$. ■

Example 3.19 Let X be a discrete random variable with $p_X(x) = 2/[(x+1)(x+2)]$ for all natural numbers $x \geq 1$. Find the median, quartiles, and interquartile range of X .

Solution: The PMF can be expressed as a difference of fraction in the form $[2/(x+1)] - [2/(x+2)]$. This form allows the cancelation of adjacent terms when computing the cumulative distribution function. The corresponding cumulative distribution function is therefore as follows:

$$F_X(a) = \sum_{x=1}^a ([2/(x+1)] - [2/(x+2)]) = 1 - 2/(a+2)$$

On substituting $F_X(a) = 0.5$, one obtains $a = 2$. Therefore the median is the average of 2 and 3, which is 2.5.

For the first quartile, one obtains $a = 2/3$ by setting $F_X(a) = 0.25$. Since a is a fraction lying in $(0, 1)$ it follows that the first quartile is 1. For the third quartile, one obtains $a = 6$ by setting $F_X(a) = 0.75$. Therefore, the third quartile lies between 6 and 7 and is interpolated to 6.5. The interquartile range is $(6.5 - 1) = 5.5$. ■

As in the case of samples, one can define the distribution variance as the expected squared deviation from the distribution mean:

Definition 3.26 (Distribution Variance) *Let X be a random variable with expected value $\mu_X = E[X]$. Then, the distribution variance σ_X^2 is given by the expected value of $(X - \mu_X)^2$:*

$$\sigma_X^2 = E[(X - \mu_X)^2]$$

A second method exists to compute the variance in terms of the expectation of the *second moment* $E[X^2]$ and the *first moment* $\mu_X = E[X]$ of the random variable:

Lemma 3.10 *The variance of a random variable X with expected value μ_X can be expressed as follows:*

$$\sigma_X^2 = E[X^2] - \mu_X^2$$

Proof: By expanding the expression for the variance, one obtains the following:

$$\sigma_X^2 = E[(X - \mu_X)^2] = E[X^2 - 2X\mu_X + \mu_X^2]$$

By using the property of expectation under scaling and sums, one obtains the following:

$$\sigma_X^2 = E[X^2] - 2E[X]\mu_X + \mu_X^2$$

by substituting $E[X] = \mu_X$ in the above expression and simplifying, one obtains the desired result. ■

A natural question arises as to why Lemma 3.10 is important. It turns out that many distributions have special structure, because of which their moments are easy to compute. In such cases, many useful statistical properties such as the mean and the variance can be conveniently derived from the moments. A detailed discussion of moments and their properties is provided in section 4.14 of Chapter 4.

The distribution standard deviation is the square-root of the distribution variance:

Definition 3.27 (Distribution Standard Deviation) *The distribution standard deviation is the square-root of the distribution variance.*

The variance of a distribution is not affected by translation but it scales with the square of the scaling factor:

Lemma 3.11 (Variance under Scaling and Translation) *Let $Y = a \cdot X + b$ be an affine transformation of the random variable X for constants a and b . Then, the variances of X and Y are related as follows:*

$$\sigma_Y^2 = a^2 \cdot \sigma_X^2$$

This lemma can be easily shown by observing that $(Y - \mu_Y)^2 = a^2 \cdot (X - \mu_X)^2$, as translation does not affect either $(X - \mu_X)$ or $(Y - \mu_Y)$ and the only effect is that of scaling. Taking the expectation of both sides yields the result.

The variance of a sum of random variables is *not* equal to the sum of their variances, unless the random variables are independent. This is an issue that will be discussed in a later section.

Definition 3.28 (Distribution Mean Absolute Deviation) *The mean absolute deviation MAD_X of the random variable X is the expected value of $|X - \mu_X|$.*

$$MAD_X = E[|X - \mu_X|]$$

Example 3.20 Let $f_X(x) = 2x$ be the density function for random variable X , which is defined only on the domain $(0, 1)$. The probability density is 0 outside $(0, 1)$. Compute the expected value, median, first quartile, and variance of random variable X .

Solution: The expected value $E[X]$ and squared expected value is given by the following:

$$\begin{aligned} E[X] &= \int_0^1 xf_X(x)dx = [2x^3/3]_0^1 = 2/3 \\ E[X^2] &= \int_0^1 x^2 f_X(x)dx = [2x^4/4]_0^1 = 1/2 \end{aligned}$$

The variance is given by $E[X^2] - E[X]^2 = (1/2) - (4/9) = 1/18$. The cumulative distribution is given by x^2 in $(0, 1)$ by integration. The first quartile is therefore obtained by finding a so that $a^2 = 1/4$ and the median is obtained by finding b so that $b^2 = 1/2$. The first quartile and median are therefore $1/2$ and $1/\sqrt{2}$, respectively. ■

Example 3.21 Suppose that you have a random variable X that lies in $[a, b]$. Show that the distribution variance is bounded above by $(b-a)^2/4$. How is this result related to the bound in Example 2.5? What accounts for the difference in the bound in the two cases?

Solution: The solution is similar to that of Example 2.5, except that the Bessel correction is not used. It can be shown by method of contradiction that the variance is maximized for distributions in which all probabilities are concentrated at either a or b by modifying any intermediate probability mass to the extremes. Then, it can be shown that if the probability of a is f , the variance is $f(1-f)(b-a)^2$. This variance is maximized when f is 0.5, which yields the above expression. ■

Problem 3.23 Let X be a random variable in $[a, b]$. Show that for any even integer n , the following result is true:

$$E[(X - E[X])^n] \leq \left(\frac{b-a}{2}\right)^n$$

The above problem is a generalization of Example 3.21. Therefore, the hint given for Example 3.21 is applicable to this case as well.

Problem 3.24 Let X_1 and X_2 be any pair of independent and identically distributed (i.i.d.) variables. Show that for any odd n , the following result is true:

$$E[(X_1 - X_2)^n] = 0$$

A hint for solving the previous problem is to use the binomial expansion of $(X_1 - X_2)^n$. Show that the terms in the binomial expansion can be paired so that each pair adds up to 0.

3.9.2 Distribution Covariance

As in the case of multivariate summary statistics over samples, the most popular measure of multivariate association is the distribution covariance:

Definition 3.29 (Distribution Covariance) Let X and Y be two random variables drawn from a joint distribution $f_{(X,Y)}(x,y)$ with expected values μ_X and μ_Y , respectively. Then, the covariance σ_{XY} between X and Y is defined as follows:

$$\sigma_{XY} = E[(X - \mu_X)(Y - \mu_Y)]$$

A positive value of the covariance means that pairwise samples drawn from the joint distribution will be positively correlated. As a larger and larger number of samples is drawn, the sample covariance will converge to the distribution covariance. The distribution-centric covariance can also be expressed in the following alternative form:

Lemma 3.12 Let X and Y be two random variables drawn from a joint distribution $f_{(X,Y)}(x,y)$ with expected values μ_X and μ_Y , respectively. Then, the covariance σ_{XY} between X and Y can be expressed as follows:

$$\sigma_{XY} = E[XY] - \mu_X\mu_Y$$

Proof: By expanding the expression for the variance, one obtains the following:

$$\sigma_{XY} = E[(X - \mu_X)(Y - \mu_Y)] = E[XY - X\mu_Y - \mu_X Y + \mu_X\mu_Y]$$

By using the property of expectation under scaling and sums, one obtains the following:

$$\sigma_{XY} = E[XY] - E[X]\mu_Y - \mu_X E[Y] + \mu_X\mu_Y$$

by substituting $E[X] = \mu_X$ and $E[Y] = \mu_Y$ in the above expression and simplifying, one obtains the desired result. ■

It is noteworthy that the notion of covariance is a generalization of the concept of variance, since the variance of a variable is the covariance of the variable with itself. Correspondingly, this lemma is a generalization of Lemma 3.10 on variances. The proof technique used is also very similar. Furthermore, this lemma is also the distribution-centric version of the sample-centric Lemma 2.1 in section 2.2 of Chapter 2. We mentioned earlier that the variance of the sum of two random variables is *not* equal to the sum of their variances. This is because of the covariance between the two variables, which can either increase or decrease the variance of the sum of variables:

Lemma 3.13 Let X and Y be two random variables drawn from a joint distribution $f_{(X,Y)}(x,y)$ with expected values μ_X and μ_Y , respectively. Then, the variance σ_{X+Y}^2 of the sum of the random variables can be expressed in term of the individual variances σ_X^2 , σ_Y^2 and covariance σ_{XY} as follows:

$$\sigma_{X+Y}^2 = \sigma_X^2 + \sigma_Y^2 + 2\sigma_{XY}$$

Proof: The variance σ_{X+Y}^2 can be expressed as follows:

$$\begin{aligned}\sigma_{X+Y}^2 &= E[((X - \mu_X) + (Y - \mu_Y))^2] \\ &= E[(X - \mu_X)^2] + E[(Y - \mu_Y)^2] + 2E[(X - \mu_X)(Y - \mu_Y)] \\ &= \sigma_X^2 + \sigma_Y^2 + 2\sigma_{XY}\end{aligned}$$

The result follows. ■

Note that a negative covariance between two random variables will reduce the variance of the sum of the variables. In the extreme case, where $Y = -X + a$, the sum of the two variables is a constant whose variance is 0.

Example 3.22 Let X and Y be two positively correlated random variables so that $Y = 2X + 5$. Show that the variance of the sum of X and Y is $(9/5)$ times the sum of the variances of X and Y .

Solution: Note that adding constants such as 5 do not affect the variance whereas multiplying by 2 causes the variance to be multiplied by 2^2 . Furthermore the sum of X and Y is $3X + 5$.

$$\begin{aligned}\sigma_X^2 + \sigma_Y^2 &= \sigma_X^2 + \sigma_{2X}^2 = \sigma_2^X + 2^2\sigma_X^2 = 5\sigma_X^2 \\ \sigma_{X+Y}^2 &= \sigma_{3X}^2 = 9\sigma_X^2\end{aligned}$$

Dividing the two quantities above, one obtains $9/5$. ■

Example 3.23 Consider the probability density function $f_{(X,Y)}(x,y) = x+y$, which is defined only over $x, y \in (0, 1)$ (and taking a value of 0 elsewhere). Compute $E[XY]$, μ_X , and μ_Y by setting up the appropriate integrals. Use these quantities to compute σ_{XY} .

Solution: One can compute the marginal distribution $f_X(x)$ as follows:

$$f_X(x) = \int_{y=0}^1 (x+y) dy = x + 0.5$$

Similarly, the marginal distribution $f_Y(y)$ is $(y+0.5)$. The expected values $\mu_X = E[X]$ can be shown to be the following:

$$\mu_X = \int_{x=0}^1 x(x+0.5) dx = (1/3) + (1/4) = (7/12)$$

Similarly, μ_Y can be shown to be $(7/12)$. The expected value $E[XY]$ is as follows:

$$\begin{aligned} E[XY] &= \int_{x=0}^1 \int_{y=0}^1 xy(x+y) dx dy = \int_{x=0}^1 [(x^2/2) + (x/3)] dx \\ &= [(x^3/6) + (x^2/6)]_0^1 = (1/3) \end{aligned}$$

The covariance is given by the following:

$$\sigma_{XY} = E[XY] - E[X]E[Y] = (1/3) - (7/12)^2 = -(1/144)$$

■

Problem 3.25 For any pair of random variables X and Y drawn from a joint distribution, derive an expression for the variance of $(X - Y)$ in terms of the variances of the individual variables X , Y and the covariance σ_{XY} .

Problem 3.26 Let $Y = \sum_{i=1}^d X_i$, where $\vec{X} = [X_1, \dots, X_d]$ is drawn from a d -dimensional joint distribution. Let σ_i^2 denote the variance of random variable X_i and let σ_{ij} be the covariance between X_i and X_j . Show the following result relating the variance σ_Y^2 of Y to the variances and covariances of X_i :

$$\sigma_Y^2 = \sum_{i=1}^d \sigma_i^2 + 2 \sum_{i=1}^d \sum_{j=i+1}^d \sigma_{ij}$$

The proof technique for the above problem is very similar to that of Lemma 3.13, except that one must expand $E[(\sum_{i=1}^d (X_i - \mu_{X_i}))^2]$ in order to derive the expression for σ_Y^2 .

Problem 3.27 Consider the probability density function $f_{(X,Y)}(x,y) = 4xy$, which is defined only over $x, y \in (0, 1)$ (and taking a value of 0 elsewhere). Compute $E[XY]$, μ_X , and μ_Y by setting up the appropriate integrals. Use these quantities to compute σ_{XY} . Explain qualitatively using the properties of the distribution why you got the covariance that you did in terms of its sign (positive, zero, or negative).

3.9.3 Useful Multivariate Properties Under Independence

Machine learning applications often assume independence among attributes, which allow a number of modeling simplifications. These modeling simplifications sometimes use a number of properties of the expectation and the variance, which are discussed in this section. First, note that when two variables are independent, their joint distribution can be expressed as a product of their marginal distributions. It turns out that this property also applies to the expected values:

Lemma 3.14 (Expectation of Products of Random Variables) Let X and Y be two independent random variables drawn from a joint distribution $f_{(X,Y)}(x,y) = f_X(x)f_Y(y)$ with expected values μ_X and μ_Y , respectively. Then, we have the following result:

$$E[XY] = E[X]E[Y] = \mu_X\mu_Y$$

Proof:

$$\begin{aligned} E[XY] &= \int_x \int_y xy f_X(x) f_Y(y) dx dy \\ &= \left[\int_x x f_X(x) dx \right] \left[\int_y y f_Y(y) dy \right] \\ &= E[X] E[Y] = \mu_X \mu_Y \end{aligned}$$

The above proof is possible only because of the factored nature of the joint distribution. The factored nature of the joint distribution is how independence is defined. ■

The factored nature of the expectation of products can be used to show that the covariance of independent variables is 0.

Lemma 3.15 (Zero Covariance of Independent Variables) *Let X and Y be independent variables with expected values μ_X and μ_Y . In such a case, the distribution covariance σ_{XY} is 0.*

Proof: By using the form of the distribution covariance in Lemma 3.12, we obtain the following:

$$\sigma_{XY} = E[XY] - \mu_X \mu_Y$$

However, since the variables are independent, we already know that the expectation $E[XY]$ of products is the product of expectations. In other words, we have $E[XY] = \mu_X \mu_Y$. The result follows. ■

Finally, the fact that the covariances are zero for independent variables helps establish the fact that the variance of the sum of independent variables is the sum of the variances. We already know from Lemma 3.13 that the variance of the sum of random variables satisfies the following relationship:

$$\sigma_{X+Y}^2 = \sigma_X^2 + \sigma_Y^2 + 2\sigma_{XY}$$

Setting the covariance to zero for independent variables yields the following result on the sum of variances of independent variables:

Lemma 3.16 (Variance of Sum of Independent Variables) *Let X and Y be two independent random variables. Then, the variance σ_{X+Y}^2 of the sum of the random variables is the sum of the variances σ_X^2 and σ_Y^2 :*

$$\sigma_{X+Y}^2 = \sigma_X^2 + \sigma_Y^2$$

One can combine the above result with the scaling law of variances (cf. Lemma 3.11) in order to show the following:

Theorem 3.3 (Variance of Average of Random Variables) *Let X_1 and X_2 be two independent and identically distributed (i.i.d.) random variables and Z be the random variable denoting their average. Then, the variance σ_Z^2 is half the variance of that of each X_i . In general, averaging n i.i.d. variables reduces their variance by a factor of n .*

The above result is important because it is used in all sorts of sampling-based statistical settings such as *hypothesis testing* or building *confidence intervals* of averaged statistics. A detailed discussion of hypothesis testing is provided in Chapter 5.

Example 3.24 Compute the mean and variance of the outcome of a single die throw of a fair die. Compute the mean and variance of the average outcome of 100 die throws. Comment on the result.

Solution: Let X be the outcome of a die throw. Then, one can compute $E[X] = \sum_{i=1}^6 i/6 = 3.5$ and $E[X^2] = \sum_{i=1}^6 i^2/6 = 91/6$. The variance of a single throw is therefore given by the following:

$$\sigma_X^2 = (91/6) - 3.5^2 \approx 2.917$$

The expected mean of the average of 100 die throws is still 3.5. However, the variance of the average of 100 throws is obtained by dividing 2.917 by 100, which is 0.02917. The low variance is caused by the averaging process, which reduces the variability of the outcome. ■

Problem 3.28 Let $Z = b + \sum_{i=1}^d a_i X_i$ be a random variable that is an affine function of independent random variables $X_1 \dots X_d$. Show the following result:

$$\sigma_Z^2 = \sum_{i=1}^d a_i^2 \sigma_{X_i}^2$$

3.10 Compound Distributions

A compound distribution is defined by using a sequence of random draws in which the draw of a random variable uses a distribution that is chosen based on the outcome of earlier draws of other random variables. Section 3.8.2 already introduces the notion of compound distributions in an indirect way by defining the conditional distribution $f_{X|A_i}(x)$, where the conditioning is done over the occurrence of one of events A_1, A_2, \dots, A_r . A conditional distribution such as $f_{X|A_i}(x)$ naturally enables the type of dynamic for generating X — the i th distribution $f_{X|A_i}(x)$ is selected by first drawing A_i (from the choices A_1, \dots, A_r) and generating X from this distribution. The different $f_{X|A_i}(x)$ often represent different parameter choices from the same family of distributions in practical applications. Recall that this approach was used to determine the posterior probabilities $P(A_i|X = x)$ of the $r = 2$ possibilities for parameter choices in the example involving the lengths of freshwater and seawater salmon (see Example 3.16).

A further generalization of the aforementioned idea (of selecting one of a fixed set of r parameters) is to first generate the parameter Θ from a prior distribution (which might be continuous) and then generate X using the conditional distribution $f_{X|\Theta=\theta}(x)$. This distribution is conditional in the sense that it is parameterized by Θ , which is itself a random variable with prior distribution $f_\Theta(\theta)$. The unconditional distribution $f_X(x)$ is referred to as a *compound distribution*, because it compounds the effect of parameter randomness and data generation (given a chosen parameter). The posterior distribution $f_{\Theta|X=x}(\theta)$ is also very useful in *Bayesian statistics* for estimating the optimal choice of the parameters Θ based on observed data using a concept known as *maximum a posteriori estimation* (cf. section 6.6.2 of Chapter 6). It turns out that variants of the total probability rule and the Bayes rule can be developed in these cases. The following section will introduce these variants.

3.10.1 Total Probability Rule in Continuous Hypothesis Spaces

In order to understand the notion of compounding, we will explain the total probability rule with the help of an example. Consider the case where fishes were exposed to mercury poisoning for a short period, and the amount of mercury poisoning $\Theta \in (0, 1)$ in fish is a random variable that follows the following distribution:

$$f_\Theta(\theta) = 2\theta \quad \forall \theta \in (0, 1) \text{ and } 0, \text{ otherwise}$$

Furthermore, the residual life-time X of a fish in years (just after mercury poisoning) depends on the amount of mercury poisoning based on the following conditional distribution:

$$f_{X|\Theta=\theta}(x) = \theta \exp(-\theta x) \quad \forall x > 0 \text{ and } 0, \text{ otherwise}$$

Note that θ is a parameter of the above conditional distribution. This conditional distribution of the residual lifetime of a fish is an exponential distribution with parameter θ (cf. Chapter 4), which has the property that larger values of θ (poisoning) result in lower expected values of X (residual lifetimes). Intuitively, heavily poisoned fish are expected to have shorter lifetimes. One would like to find the unconditional probability density function $f_X(x)$ of the lifetime of all fish irrespective of their level of mercury poisoning. The resulting distribution $f_X(x)$ is referred to as the compound distribution of the two aforementioned distributions $f_\Theta(\theta)$ and $f_{X|\Theta=\theta}(x)$. The parameter Θ represents the continuous hypothesis space for the conditional distribution $f_{X|\Theta=\theta}(x)$.

It turns out that the following continuous version of the total probability rule is useful for computing compound distributions:

Lemma 3.17 (Total Probability Rule in Continuous Hypothesis Spaces) *Let $f_\Theta(\theta)$ be the distribution of the random variable Θ , where the realized values θ of Θ parameterize the conditional distribution $f_{X|\Theta=\theta}(x)$. Then, the unconditional density function $f_X(x)$ is given by the following:*

$$f_X(x) = \int_{\theta=-\infty}^{\infty} f_\Theta(\theta) f_{X|\Theta=\theta}(x) d\theta$$

The result can also be generalized to discrete random variable X with conditional distribution $p_{X|\Theta=\theta}(x)$ by substituting the conditional probability density function in the above equation with a conditional probability mass function:

$$p_X(x) = \int_{\theta=-\infty}^{\infty} f_\Theta(\theta) p_{X|\Theta=\theta}(x) d\theta$$

The main difference from the total probability rule in Lemma 3.6 is that summations have been replaced by integrals, since the prior event has a continuous distribution. This is a natural generalization, given that integrals are continuous versions of summations (based on the notion of *Riemann integral*).

Based on this result, let us compute the unconditional probability distribution of fish lifetimes:

$$\begin{aligned} f_X(x) &= \int_{\theta=-\infty}^{\infty} f_\Theta(\theta) f_{X|\Theta=\theta}(x) d\theta \\ &= \int_{\theta=0}^1 2\theta^2 \exp(-x \cdot \theta) d\theta \end{aligned}$$

It is noteworthy that the above integral is being computed for fixed x , which should therefore be treated as a constant in the integration process. One can use integration by parts to show that the unconditional density function evaluates to the following:

$$\begin{aligned} f_X(x) &= \left[-\frac{2\theta^2 \exp(-\theta x)}{x} - \frac{4\theta \exp(-\theta x)}{x^2} - \frac{4 \exp(-\theta x)}{x^3} \right]_{\theta=0}^{\theta=1} \quad \forall x > 0 \\ &= \frac{4 - 4 \exp(-x)}{x^3} - \frac{4 \exp(-x)}{x^2} - \frac{2 \exp(-x)}{x} \quad \forall x > 0 \end{aligned}$$

The above density function is rather clumsy in spite of the fact that the prior density function $f_\Theta(\theta)$ and the conditional density function $f_{X|\theta=\theta}(x)$ have very simple forms. Such a situation is extremely common, and in many cases, the aforementioned integral cannot even be computed in closed form (in which case it is evaluated numerically). It is instructive to examine the conditional density function of X with respect to various parameter values and examine it in relation to the unconditional density function.

Figure 3.7(a) shows the conditional distributions of the residual life of fish at mercury poisoning parameter $\theta = 0.5$ and $\theta = 1$, respectively. The unconditional lifetime $f_X(x)$ that was generated using the total probability rule is also shown in the same figure. It is evident that fishes with high poisoning ($\theta = 1$) tend to have lower lifetimes on average than fishes with low poisoning (based on the two conditional distributions). Furthermore, the unconditional distribution lies somewhere between the two distributions for the most part (although this is not true for all values of the lifetime x). Note that the average poisoning level $E[\Theta]$ over all fish can be shown to be $2/3$ by simple integration using the prior (poisoning) distribution function $f_\Theta(\theta)$. This is consistent with the fact that the unconditional density function lies somewhere between the two conditional densities at $\theta = 0.5$ and $\theta = 1$. However, the unconditional density function has a much more complex distribution, which is the result of merging an infinite number of conditional density functions over different values of θ . This type of merging is similar to the example of section 3.8.3, which merges the length of freshwater fish and seawater fish into a single distribution. The main difference is that the merging is now done over the *infinite* and *continuous* hypothesis space of poisoning levels Θ rather than a finite hypothesis space of two types of fish. The unconditional distribution has a heavier inverse polynomial tail than the exponentially decaying tail of the conditional distribution for fixed Θ . Therefore, the behavior of compound distributions is not always easy to completely predict; they can often have artifacts that are different from the corresponding conditional distributions.

Example 3.25 Let Θ be a positive random variable distributed according to the density function $f_\Theta(\theta) = \exp(-\theta)$ for $\theta > 0$. Let X be a positive random variable with the conditional distribution $f_{X|\Theta=\theta}(x) = \theta \exp(-\theta x)$ for $x > 0$. Show that the expression for the density function of the compound distribution $f_X(x)$ (i.e., the unconditional distribution) is as follows:

$$f_X(x) = \frac{1}{(x+1)^2} \quad \forall x > 0$$

Solution: One can use the total probability rule for continuous hypothesis spaces to derive the following:

$$f_X(x) = \int_{\theta=0}^{\infty} f_\Theta(\theta) f_{X|\Theta=\theta}(x) d\theta = \int_{\theta=0}^{\infty} \theta \exp(-\theta[x+1]) d\theta$$

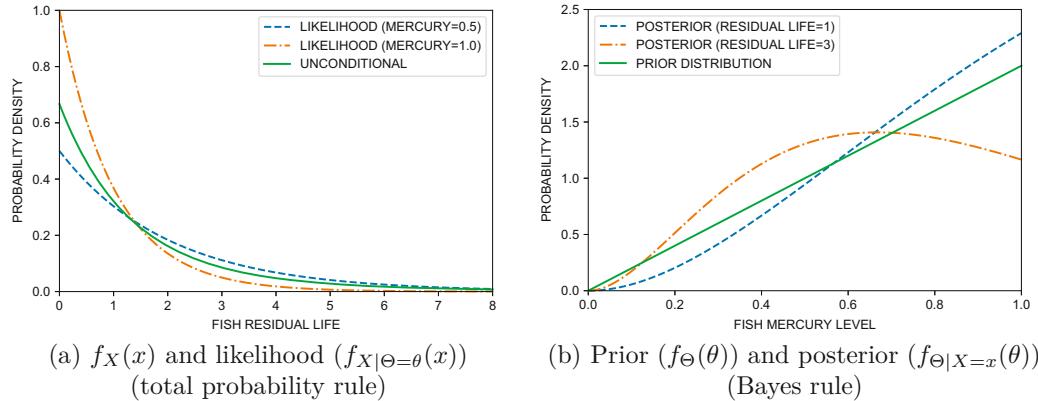


Figure 3.7: Illustration of continuous version of the total probability rule and Bayes rule

In the aforementioned integral with respect to θ , x is treated as a constant. Integrating by parts, one can derive the following:

$$f_X(x) = \left[-\frac{\theta \exp(-\theta[x+1])}{x+1} - \frac{\exp(-\theta[x+1])}{(x+1)^2} \right]_{\theta=0}^{\infty} = \frac{1}{(x+1)^2}$$

Note that the substitution of limits in the above (improper) integral uses the fact that the first term tends to 0 at both the upper and lower bounds for positive values of x . ■

3.10.2 Bayes Rule in Continuous Hypothesis Spaces

Consider a real-world situation where it is impossible or impractical to measure the poisoning levels of fish exactly, but one can (eventually) measure the lifetime of each fish. In other words, the random variable X corresponding to the lifetime can be observed, whereas the random variable Θ for the poisoning level can only be hypothesized (but not observed). This is an ideal setting for the application of the Bayes rule in a continuous hypothesis space.

In this scenario, the distribution of Θ needs to be inferred for a particular fish based on its observed lifetime $X = x$ (i.e., lifetime corresponding to a particular number of years). In this scenario, the density $f_\Theta(\theta)$ of the poisoning levels of fishes is the *prior distribution*, whereas the density $f_{\Theta|X=x}(\theta)$ is the posterior distribution after observing the lifetime of the fish. In other words, we want the posterior distribution of the entire hypothesis space, given the observed lifetimes. Obviously, fish that live longer than average will have a different posterior distribution of poisoning levels (biased towards less poisoning) than all fish. An important assumption is that the conditional distribution $f_{X|\Theta=\theta}(x)$ of the lifetime X for poisoning level $\Theta = \theta$ is available (and is typically the result of domain-specific modeling). This distribution is referred to as the likelihood in Bayesian statistics parlance.

The Bayes rule uses the conditional distributions of the observed lifetimes (likelihoods) in order to compute the posterior distribution of the poisoning levels. The continuous version of Bayes rule is as follows:

Lemma 3.18 (Bayes Rule in Continuous Hypothesis Spaces) Let $f_\Theta(\theta)$ be the prior density function of the parameter Θ , where the realized values θ of Θ parameterize the conditional distribution $f_{X|\Theta=\theta}(x)$. Then, the posterior density function $f_{\Theta|X=x}(\theta)$ is given by the following:

$$f_{\Theta|X=x}(\theta) = \frac{f_\Theta(\theta) \cdot f_{X|\Theta=\theta}(x)}{\int_{\theta=-\infty}^{\infty} f_\Theta(\theta) f_{X|\Theta=\theta}(x) d\theta}$$

The result can also be generalized to discrete random variable X with conditional distribution $p_{X|\Theta=\theta}(x)$ by substituting the conditional probability density function in the above equation with a conditional probability mass function:

$$f_{\Theta|X=x}(\theta) = \frac{f_\Theta(\theta) \cdot p_{X|\Theta=\theta}(x)}{\int_{\theta=-\infty}^{\infty} f_\Theta(\theta) p_{X|\Theta=\theta}(x) d\theta}$$

Note that the denominator in the above equation is inherited from the total probability rule.

As an example, we will compute the posterior distribution of the poisoning levels for fishes with lifetime $x = 1$ and $x = 3$ years, respectively. The expected lifetime $E[X]$ over all fishes can be shown to be 2 by using the unconditional distribution $f_X(x)$ derived in the previous section; it would be instructive to compare the posterior distributions of the poisoning levels of relatively short-lived fish ($x = 1$) and long-lived fish ($x = 3$), and compare them to the prior distribution $f_\Theta(\theta)$. The denominator of the Bayes rule can be substituted using the total probability rule discussed in the previous section to obtain the following posterior distribution:

$$\begin{aligned} f_{\Theta|X=x}(\theta) &= \frac{f_\Theta(\theta) \cdot p_{X|\Theta=\theta}(x)}{\int_{\theta=-\infty}^{\infty} f_\Theta(\theta) p_{X|\Theta=\theta}(x) d\theta} \\ &= \frac{2\theta^2 \exp(-\theta x)}{\left[\frac{4 - 4 \exp(-x)}{x^3} - \frac{4 \exp(-x)}{x^2} - \frac{2 \exp(-x)}{x} \right]} \end{aligned}$$

By instantiating $x = 1$ and $x = 3$, we obtain the following:

$$\begin{aligned} f_{\Theta|X=1}(\theta) &= \frac{2\theta^2 \exp(-\theta)}{0.3212} \\ f_{\Theta|X=3}(\theta) &= \frac{2\theta^2 \exp(-3\theta)}{0.08545} \end{aligned}$$

The posterior distribution of Θ at $X = 1$ and $X = 3$ are shown in Figure 3.7(b). The prior distribution $f_\Theta(\theta) = 2\theta$ is also shown in the figure for reference. Note that for the case of short-lived fish (i.e., $X = 1$), the posterior distribution of mercury poisoning level tends to show a shift towards higher densities of larger mercury values (relative to the prior distribution). This is because a life-time of 1 is less than the (unconditional) expected lifetime $E[X] = 2$, which strongly biases the posterior probability distribution towards greater poisoning levels. On the other hand, for longer-lived fish with $X = 3$, the posterior distribution first increases and peaks at a certain point before dropping off. In other words, high mercury levels are rare for long-lived fish.

The posterior density $f_{\Theta|X=3}(\theta)$ peaks at $\theta = 0.6667$, which suggests that it is the most likely value of the random parameter Θ for long-lived fish satisfying $X = 3$. Fixing

$\theta = 0.6667$ helps us reconstruct a conditional distribution $f_{X|\Theta=0.6667}(x)$, which is a better choice than $f_X(x)$ for modeling the lives of fishes with similar poisoning characteristics as a long-lived fish with a residual lifetime $X = 3$ years. An important point is that one can find the optimum value θ^* of Θ not just for a specific value of the lifetime X but an entire data set of fishes with various lifetimes, and then create an optimized distribution $f_{X|\Theta=\theta^*}(x)$ of residual fish lifetimes. This distribution will be a better representative of fish lifetimes under the assumption that the data set reflects poisoning levels of interest. The use of lifetime information from the specific data set at hand therefore allows us to choose a “well-informed” hypothesis θ^* , which enables more robust analysis in downstream applications (like classification). This approach is referred to as *maximum a posteriori estimation*, although the above example uses only a single observation to do so. The generalization of this approach to a data set of observations is discussed in section 6.6.2 of Chapter 6.

Example 3.26 Consider the scenario of Example 3.25 in which Θ is a positive random variable with density function $f_\Theta(\theta) = \exp(-\theta)$ and X has conditional distribution $f_{X|\Theta=\theta}(x) = \theta \exp(-\theta x)$ for $x > 0$. Show that the posterior density function $f_{\Theta|X=x}(\theta)$ is as follows:

$$f_{\Theta|X=x}(\theta) = (x+1)^2 \theta \exp(-[x+1]\theta)$$

Suppose that the value $X = x_0$ is observed, but you do not know the realized value of Θ . Show that the posterior density of Θ given $X = x_0$ is maximized at the following value of $\theta = \theta^*$:

$$\theta^* = \frac{1}{x_0 + 1}$$

Solution: We use the Bayes rule for continuous hypothesis spaces to derive the posterior density as follows:

$$f_{\Theta|X=x}(\theta) = \frac{f_\Theta(\theta) f_{X|\Theta=\theta}(x)}{\int_{\theta=0}^{\infty} f_\Theta(\theta) f_{X|\Theta=\theta}(x) d\theta}$$

The denominator has already been evaluated in the solution to Example 3.25, whereas the terms in the numerator are available in the problem statement. Substituting the different terms in the above expression, the following is obtained:

$$f_{\Theta|X=x}(\theta) = \frac{\exp(-\theta) \theta \exp(-\theta x)}{\left[\frac{1}{(x+1)^2} \right]}$$

It is easy to simplify the above expression to arrive at the following result:

$$f_{\Theta|X=x}(\theta) = (x+1)^2 \theta \exp(-[x+1]\theta)$$

If a single value of $x = x_0$ is observed, one can differentiate the above expression and set it to 0 to derive the mode of the above distribution at $x = x_0$:

$$(x_0 + 1)^2 \exp(-[x_0 + 1]\theta) - (x_0 + 1)^3 \theta \exp(-[x_0 + 1]\theta) = 0$$

It is easy to show that the only solution to the above problem is $\theta = 1/(x_0 + 1)$. It can also be shown that the second derivative of the above expression is negative at $\theta = 1/(x_0 + 1)$. This type of optimization can be viewed as *maximum a posteriori estimation with a single observation*. ■

Problem 3.29 For the previous problem, compute the conditional expectation $E[X|\Theta = \theta]$. Does this expectation increase or decrease when θ is larger? Use this fact to give an intuitive explanation of why the expression for θ^* (i.e., most “likely” value of θ for observed $X = x_0$) is smaller when larger values of $X = x_0$ are observed.

3.11 Functions of Random Variables (*)

In some applications, one is faced with scenarios where it is desired to find the distributions of functions of random variables. The results in this section are required only for advanced applications of machine learning, and do not appear frequently. Given the advanced nature of the material in this section, it has been designated as an asterisked section, which the reader may choose to skip without loss of continuity.

We will first discuss how the distribution of a function of a single random variable can be constructed. Then, we will focus on the distribution of the sums of independent random variables. As it turns out, the cumulative distribution is extremely helpful in both these scenarios. Therefore, the approach is referred to as the *cumulative distribution function method*. For simplicity, we will focus only on invertible one-to-one functions of the random variable, although generalizations of the approach to other types of functions are possible with modifications. Note that a continuous one-to-one function is always monotonically increasing or decreasing, a fact that will be made use of below.

3.11.1 Distribution of the Function of a Single Random Variable

Consider a continuous random variable X with known cumulative distribution function $F_X(x) = P(X \leq x)$. A new random variable Y is defined as $Y = G(X)$, where $G(\cdot)$ is an invertible function. We would like to find the cumulative distribution function $F_Y(y)$ and the density function $f_Y(y)$ of the random variable Y .

The first step is to find the cumulative distribution of Y . The reason that the cumulative distribution is preferred over the probability density is that it works with a concrete probability $F_Y(y) = P(Y \leq y)$ that lends itself to easy transformation of variables. Since $Y = G(X)$, one can immediately infer the following:

$$F_Y(y) = P(G(X) \leq y)$$

It is here that the monotonicity of $G(X)$ becomes useful. For increasing functions, $G(X) \leq y$ is the same as imposing the condition $X \leq G^{-1}(y)$; for decreasing functions, the condition is $X \geq G^{-1}(y)$. In other words, we have the following:

$$F_Y(y) = \begin{cases} P(X \leq G^{-1}(y)) = F_X(G^{-1}(y)) & \text{if } G(\cdot) \text{ is increasing} \\ P(X \geq G^{-1}(y)) = 1 - F_X(G^{-1}(y)) & \text{if } G(\cdot) \text{ is decreasing} \end{cases}$$

The probability density function of Y can be derived by differentiating the cumulative distribution function. Let $[G^{-1}(y)]'$ denote the derivative of $G^{-1}(y)$ with respect to its argument. The resulting probability density function of Y is as follows:

$$f_Y(y) = \begin{cases} f_X(G^{-1}(y))[G^{-1}(y)]' & \text{if } G(\cdot) \text{ is increasing} \\ -f_X(G^{-1}(y))[G^{-1}(y)]' & \text{if } G(\cdot) \text{ is decreasing} \end{cases} \quad (3.7)$$

An important observation here is that the derivative of $G(\cdot)$ is positive for increasing functions and negative for decreasing functions. Therefore, in both cases above the density will

be a nonnegative value Furthermore, one can use this observation to consolidate the density function into a single simplified form:

$$f_Y(y) = f_X(G^{-1}(y))|G^{-1}(y)|'$$

Not that the domain of the new random variable will change. In particular, if $X \in (a, b)$, then $Y \in (G(a), G(b))$ for increasing functions and $Y \in (G(b), G(a))$ for decreasing functions. In order to illustrate the use of this approach, we will use an example:

Example 3.27 Let $f_X(x) = 3x^2$ be the probability density function of a random variable defined only over $(0, 1)$. Define the new random variable using the monotonically increasing function $Y = G(X) = 2X^2 + 1$ in the domain of the random variable. What is the domain of the random variable Y ? Find the cumulative distribution function and the density function of Y .

Solution: It is easy to see that the cumulative distribution function of X is simply $F_X(x) = x^3$ in the domain $(0, 1)$. The function $G(X)$ is increasing, and the domain of Y is $[1, 3]$, which is obtained by plugging in the end points $(0, 1)$ of X into $G(X)$. Furthermore, the function $G^{-1}(y)$ may be computed as follows:

$$G^{-1}(y) = \sqrt{(y - 1)/2}$$

While taking the square-root, the positive sign is chosen because the domain of X is positive as well. Since the function $G(\cdot)$ is increasing in the domain $(0, 1)$, the cumulative distribution function is obtained using the increasing-function case of Equation 3.11.1:

$$F_Y(y) = F_X(G^{-1}(y)) = [G^{-1}(y)]^3 = [(y - 1)/2]^{3/2}$$

By differentiating the cumulative distribution function, one can also derive the density function:

$$f_Y(y) = \frac{3}{4\sqrt{2}} \sqrt{y - 1}$$

■

An important point is that we used the fact that $G(X)$ is increasing in the domain of X . Furthermore, care must be taken about the nature of the domain of $G(X)$ while inverting it. Otherwise, it is possible to end up with a negative or otherwise invalid density function.

Example 3.28 Let X be a random variable in $(-\infty, +\infty)$, and $Y = G(X) = aX + b$ be an affine function of X for $a \neq 0$. Show that the density function of Y is defined as follows:

$$f_Y(y) = \frac{f_X([y - b]/a)}{|a|}$$

Solution: We first compute $G^{-1}(y) = (y - b)/a$. Note that if $G(Y)$ is increasing for $a > 0$ and decreasing for $a < 0$. Therefore, one can use Equation 3.7 to derive the

following:

$$f_Y(y) = \begin{cases} f_X(G^{-1}(y))[G^{-1}(y)]' & \text{if } G(\cdot) \text{ is increasing} \\ -f_X(G^{-1}(y))[G^{-1}(y)]' & \text{if } G(\cdot) \text{ is decreasing} \end{cases}$$

$$= \begin{cases} f_X([y-b]/a)[1/a] & \text{if } a > 0 \\ -f_X([y-b]/a)[1/a] & \text{if } a < 0 \end{cases}$$

The above two cases can be consolidated to a single case using the modulus:

$$f_Y(y) = \frac{f_X([y-b]/a)}{|a|}$$

■

Example 3.29 (Generating Samples from a Continuous Distribution) Let X be a continuous random variable for which you know the cumulative distribution to be the invertible function $F(\cdot)$ but you do not have the code to generate it. However, you are given the code to generate a random variable that is uniformly distributed in $[0, 1]$. Let y be a sample generated by the code for the uniform distribution. Show that $F^{-1}(y)$ is a sample of the random variable X .

Solution: Let Y be a uniform random variable. We want to find the distribution of $F^{-1}(Y)$. Since the function $F(\cdot)$ is invertible, it is monotonic. Additionally, it is increasing, since it is a cumulative distribution function. Then, for any monotonically increasing function $F(\cdot)$, we have the following for any x :

$$P(F^{-1}(Y) \leq x) = P(Y \leq F(x))$$

Since Y is uniformly distributed in $[0, 1]$ and $F(x) \in (0, 1)$, it follows that $P(Y \leq F(x)) = F(x)$. In other words, the cumulative distribution function of $F^{-1}(Y)$ is $F(\cdot)$. Therefore, generating samples from a uniform distribution and applying the inverse cumulative function $F^{-1}(\cdot)$ of X to them generates samples from X . Note that cumulative distribution functions of continuous random variables are generally invertible as long as they have non-zero densities in their entire domain. See Exercise 24 for a related problem. ■

Problem 3.30 (Discrete Numeric Distribution) Discuss how you can modify the approach in the above example to generate samples from a discrete numeric distribution. The cumulative distribution function $F(x)$ of a discrete numeric random variable is generally not invertible.

The key hint for solving the above problem is to decide what to do when you generate a uniform random variable value which lies between the cumulative distribution values $F(x_1)$ and $F(x_2)$, where x_1 and x_2 are two consecutive discrete random variable values of X in its domain.

Problem 3.31 Let $f_X(x) = 3x^2$ be the probability density function of a random variable defined only over $(-1, 0)$. Define the new random variable using the function $Y = G(X) =$

$2X^2 - 3$. Is the function $G(\cdot)$ increasing or decreasing over the domain of X ? What is the domain of the random variable Y ? Find the cumulative distribution function and the density function of Y .

A particularly useful case is one of that of a linear function of a random variable.

Problem 3.32 (Powers of Random Variables) Let X be a random variable and $Y = G(X) = X^k$ be the k th power of X . For simplicity, assume that X is defined over a domain that is either $[-\infty, 0]$ or $[0, \infty]$ (so that $G(\cdot)$ is monotonic and one-to-one). Show that the density function of Y is defined as follows:

$$f_Y(y) = \begin{cases} \frac{f_X(y^{1/k})y^{-1+1/k}}{\frac{k}{k}} & \text{if } X \text{ is positive or } k \text{ is odd.} \\ \frac{f_X(-y^{1/k})y^{-1+1/k}}{\frac{k}{k}} & \text{if } X \text{ is negative and } k \text{ is even.} \end{cases}$$

Does the domain of the random variable Y depend on the odd/even nature of k ?

3.11.2 Distribution of the Sum of Random Variables

Next, we will discuss how to find the probability distribution of the sum of two independent random variables X and Y with given probability distributions. First, we consider the simple case in which X and Y are discrete random variables with PMFs $p_X(x)$ and $p_Y(y)$, respectively. We want to find the distribution of the random variable Z obtained by adding the two independent random variables X and Y :

$$Z = G(X, Y) = X + Y$$

We want to find $p_Z(z)$. In order for Z to take on the value of z , the value of Y must be $(z - x)$ for every possible value of $X = x$. For any specific value of z and x , this probability is $p_X(x)p_Y(z - x)$, since X and Y are independent. This probability needs to be summed up over all mutually exclusive events corresponding to different values of x to obtain the probability that $Z = z$:

$$p_Z(z) = \sum_{x=-\infty}^{\infty} p_X(x)p_Y(z - x)$$

One can naturally generalize this idea to density functions:

$$f_Z(z) = \int_{x=-\infty}^{\infty} f_X(x)f_Y(z - x)dx$$

The expression on the right-hand side is referred to as the *convolution operator*. Furthermore, a general formula exists for computing the density function of the sum of two random variables, whether they are independent or not:

$$f_Z(z) = \int_{x=-\infty}^{\infty} f_{X,Y}(x, z - x)dx$$

If X and Y are nonnegative random variables, then the above formula is changed only in terms of the lower and upper limits of the integral:

$$f_Z(z) = \int_{x=0}^z f_{X,Y}(x, z - x)dx$$

The reason that the lower and upper limits change is that both x and $(z - x)$ need to be nonnegative. Furthermore, $f_Z(z)$ takes on nonzero values only for $z > 0$, and therefore the above formula is valid only for this case. This example shows that one has to be careful about how the convolution operation is implemented when dealing with bounded random variables. The choice of bounds becomes particularly tricky when the two random variables X and Y have different bounds. In such a case, one must check x against both bounds for X , and also check $(z - x)$ against both bounds for Y in order to select the appropriate bounds of the integral.

Example 3.30 Let X and Y be nonnegative and independent (discrete) random variables with probability mass functions $p_X(x) = \lambda_1^x \exp(-\lambda_1)/x!$ and $p_Y(y) = \lambda_2^y \exp(-\lambda_2)/y!$. These types of random variables are referred to as Poisson random variables. Let $Z = X + Y$ be the sum of these variables. Show that the PMF of Z is as follows:

$$p_Z(z) = \frac{(\lambda_1 + \lambda_2)^z \exp(-[\lambda_1 + \lambda_2])}{z!}$$

Solution: First note that since the random variables are nonnegative, the convolution operator will be summed from 0 to z while evaluating $p_Z(z)$. Using the discrete convolution operator, one obtains the following:

$$\begin{aligned} p_Z(z) &= \sum_{x=0}^z p_X(x)p_Y(z-x) = \sum_{x=0}^z \frac{\lambda_1^x \lambda_2^{z-x} \exp(-[\lambda_1 + \lambda_2])}{x!(z-x)!} \\ &= \frac{\exp(-[\lambda_1 + \lambda_2])}{z!} \sum_{x=0}^z \frac{z! \lambda_1^x \lambda_2^{z-x}}{x!(z-x)!} \end{aligned}$$

One can note that the term within the summation on the right-hand side is simply the binomial expansion of $(\lambda_1 + \lambda_2)^z$. Substituting this term in place of the summation, the result is obtained. ■

Example 3.31 Let X be a nonnegative random variable with the probability density function $f_X(x) = \lambda \exp(-\lambda x)$. This variable is referred to as the exponential random variable with decay parameter λ . Show that the probability density function of the sum Z of two independent and identically distributed (i.i.d.) exponential variables with decay parameter λ is the following:

$$f_Z(z) = \lambda^2 z \exp(-\lambda z)$$

Solution: One can use the convolution operator to derive the density function as follows:

$$f_Z(z) = \int_{x=-\infty}^{\infty} f_X(x)f_Y(z-x)dx = \int_{x=0}^z [\lambda \exp(-\lambda x)][\lambda \exp(-\lambda(z-x))]dx$$

It is noteworthy that the bounds of the integrals in the above expression have been restricted to ensure that both X and $Y = Z - X$ are nonnegative. By observing

that the variable z should be treated as a constant with respect to the integral, the following expression is obtained:

$$f_Z(z) = \lambda^2 \exp(-\lambda z) \int_0^z dx = \lambda^2 z \exp(-\lambda z)$$

■

Problem 3.33 (Difference of Random Variables) Let X and Y be independent random variables with probability density functions $f_X(x)$ and $f_Y(y)$, respectively. Let $Z = X - Y$ be the difference between the random variables. Show that the density function $f_Z(z)$ is given by the following modified convolution:

$$f_Z(z) = \int_{x=-\infty}^{\infty} f_X(x) f_Y(x - z) dx$$

A hint for solving the above problem is to reduce it to the case of summing two random variables by first finding the distribution of $-Y$.

3.11.3 Geometric Derivation of Distributions of Functions

In general, given an arbitrary function of multiple random variables, such as $Z = G(X, Y)$, there is no guaranteed way to compute the distribution of Z from the distributions of X and Y . However, exploiting the properties of the density function within the context of the geometry of the sample space may sometimes provide some solutions. This approach can sometimes work in two or three dimensions, where visualization of the geometry is simpler. We emphasize that the approach is ad hoc and the geometry of the sample space cannot always be exploited to set up the integrals for computing the cumulative distribution function.

This section will use examples to elucidate the geometric approach. Consider the case where the joint distribution of X and Y is as follows:

$$f_{X,Y}(x, y) = 2 \exp(-x - 2y) \quad \forall x, y \geq 0$$

This density function has nonzero values only in the first quadrant. Suppose that we wish to find the probability distribution of the random variable Z , where Z is defined as follows:

$$Z = G(X, Y) = X^2 + Y^2$$

How does one find the density function of Z ? The key is to observe that the equation of Z is that of a circle, which immediately provides a path towards the cumulative distribution approach by enabling the setup of the integral bounds. In other words, we want to compute the mass between $x, y \geq 0$ and $x^2 + y^2 \leq z$. This is the mass of the distribution of a quarter circle in the first quadrant. The value of x ranges from 0 to z , and that of y from 0 to $\sqrt{z^2 - x^2}$ for the first quadrant. The corresponding visualization of the relevant region of the sample space is shown in Figure 3.8(a). This geometric visualization leads to the following integral for the cumulative distribution by using the fact that $F_Z(z)$ is the density

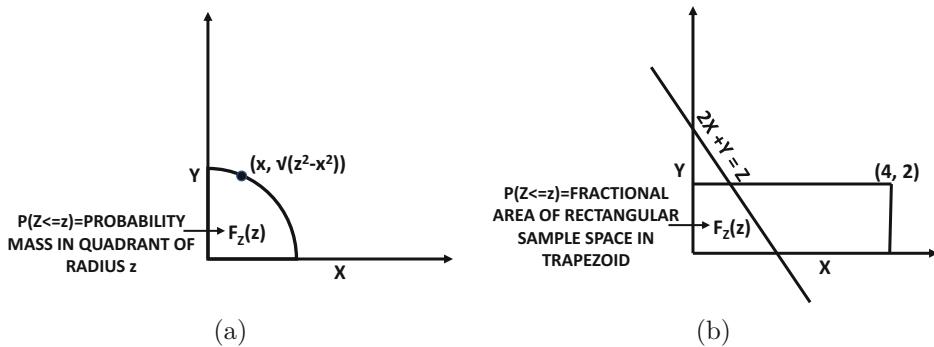


Figure 3.8: Using the geometry of the sample space to derive cumulative distributions

integral of the region $X^2 + Y^2 \leq z$:

$$\begin{aligned} F_Z(z) &= \int_{x=0}^z \left[\int_{y=0}^{\sqrt{z^2-x^2}} f_{X,Y}(x,y) dy \right] dx = \int_{x=0}^z \exp(-x) \left[\int_{y=0}^{\sqrt{z^2-x^2}} 2\exp(-2y) dy \right] dx \\ &= \int_{x=0}^z \exp(-x)(1 - \exp(-2\sqrt{z^2 - x^2})) dx \end{aligned}$$

A closed-form solution does not exist for this cumulative distribution function. Nevertheless, it can be numerically integrated to obtain a value at each $Z = z$. This example shows that using the geometry of sample spaces can sometimes help compute the density functions of random variables. In general, the computation of the density function of a function $G(\cdot)$ of random variables often requires one to understand the geometry of the region $G(\cdot) \leq z$. This geometry is not always easy to express in terms of z , which makes computation difficult.

In order to further understand the geometric approach, we will consider a second example where the sample space is in the form of a rectangle.

Example 3.32 Consider the following 2-dimensional density function:

$$f_{X,Y}(x,y) = \begin{cases} 1/8 & \text{if } x \in (0,4) \text{ and } y \in (0,2) \\ 0 & \text{otherwise} \end{cases}$$

Find the density function for the random variable $Z = G(X, Y) = 2X + Y$.

Solution: It is possible to solve this problem by first computing the density function of $2X$ and then applying a convolution. However, a simpler approach presents itself. One can draw the rectangular sample space and then compute the fraction of the area of the rectangular region bounded by the axes and the line $2x + y \leq z$ for fixed z . This area is shown as a trapezoid in Figure 3.8(b). This (fractional) area provides the cumulative distribution function. In the event that the density is not uniform, integrals need to be set up to calculate the probability mass within the appropriate region of the rectangle.

A key point is that there are three possible cases depending on the value of z , because the line $2x + y = z$ may intersect different sides of the rectangle depending on the value of z . For example, even though the region marked $F_Z(z)$ is a trapezoid in

Figure 3.8(b), it could very easily be a triangle by reducing the value of z to the range $(0, 2)$ and moving down the line $3x + y = z$. For values of $z \in (6, 8)$, the line will move up, so that the shape of the relevant region for $F_Z(z)$ will be a pentagon (in the form of the rectangular sample space with a corner cut off). For values of z less than 0 or greater than 10, the line does not intersect the sample space at all. These different cases need to be considered when setting up the cumulative distribution function. In particular, the value of the cumulative distribution function can be shown to be the following (based on this area calculation):

$$F_Z(z) = \begin{cases} 0 & \text{if } z < 0 \\ z^2/32 & \text{if } 0 \leq z \leq 2 \\ (z-1)/8 & \text{if } 2 \leq z \leq 8 \\ 1 - (10-z)^2/32 & \text{if } 8 \leq z \leq 10 \\ 1 & \text{otherwise} \end{cases}$$

One can differentiate this cumulative distribution function to obtain a piecewise linear density function:

$$f_Z(z) = \begin{cases} z/16 & \text{if } 0 \leq z \leq 2 \\ 1/8 & \text{if } 2 \leq z \leq 8 \\ (10-z)/16 & \text{if } 8 \leq z \leq 10 \\ 0 & \text{otherwise} \end{cases}$$

Deriving the above cumulative distribution requires case-wise analysis for the various ways in which the line $2x + y = z$ intersects the rectangular sample space (or doesn't intersect it at all). ■

Problem 3.34 This problem repeats Example 3.32, except that the joint density function of (X, Y) is $f_{X,Y}(x, y) = xy/16$. In other words, derive the density function for $Z = G(X, Y) = 2X + Y$ in a rectangular sample space $[0, 4] \times [0, 2]$ with the probability density function $f_{X,Y}(x, y) = xy/16$.

3.12 Summary

This chapter introduces fundamental concepts of probability like sample spaces and event spaces. The mapping of sample spaces to random variables is discussed along with the concept of probability distributions. The notion of conditional probability is introduced along with the total probability rule and the Bayes theorem. Furthermore, the total probability rule and the Bayes rule can be extended to continuous hypothesis spaces, which leads to the notion of compound distributions and maximum a posteriori estimation. The properties of random variables, such as mean, variance, and covariance, are discussed. This chapter also discusses methods to compute the distributions of some simple functions of random variables.

3.13 Further Reading

The basics of probability are introduced in [9, 19]. The exposition [19] provides the simplest discussion of random variables and probability distributions along with discussions of advanced methods like moment generating functions for computing functions of random variables. The book is also rich with instructive problems. A classical book [58] discusses the algebra of random variables in terms of how derivative distributions may be obtained for functions of variables with known distributions. A statistical view of samples drawn from probability distributions is given in [63]. The Bayesian view of statistics is given in [14, 27, 47].

3.14 Exercises

1. Suppose that you toss a coin and roll a die simultaneously, and observe the pair of values as a single outcome of an experiment. What is the sample space of this experiment? How would you represent this sample space as a 2-dimensional discrete random variable vector? What is the size of the sample space in this problem?
2. Consider the setting of Exercise 1. How would you define the event that the coin shows up as heads as a set of outcomes from the sample space of Exercise 1?
3. Suppose that you roll two fair dice together. Find the probability that the sum of their face values is odd.
4. Assuming that each outcome in the sample space of Exercise 1 is equally likely (i.e., the die and coin are fair), construct the PMF table of the random variable vector. Make sure that the probability entries of the table sum to 1.
5. You roll a blue die and red die together. You are paid the face value of the blue die and penalized the face value of the red die. Construct the PMF table of the random variable representing the payoff.
6. The probability that the Chicago Bulls win both of their next two games is $1/4$ and the probability that they win at least one of their next two games is $1/3$. If the probability that they win the next game is the same as that of winning the next one after that, what is this probability? What is the probability that they win exactly one game of their next two games?
7. You throw a die 5 times. What is the probability that you obtain the sequence 2, 3, 2, 3, 2, 3? What is the probability that you obtain three 2s and three 3s in any order? What is the probability that you obtain only 2s and 3s in six throws? What is the probability that you obtain at least one throw out of six throws in which the outcome is either a 2 or a 3?
8. Suppose that you roll two fair dice together. Find the probability that the sum of their face values is odd given that it is known to be at most 6. Are the events of the sum of the face values being odd and being at most 6 independent?
9. The probability that it rains tomorrow is 0.5. If it rains, the probability that an accident occurs on street A and street B tomorrow is each 0.1 (and independent).

Otherwise, there is no chance that an accident occurs on either street. Create notations for events in order to show how the mathematical relationship for conditional independence holds in this case. Also show that the occurrences of accidents on the two streets are not (unconditionally) independent.

10. You have two biased coins, one of which shows heads with probability 0.6 and the other shows heads with probability 0.2. You select one of the two coins at random with equal probability and toss it. You observe a heads. What is the probability that you selected the coin that shows heads with probability 0.6?
11. Michael Jordan was injured in the last basketball game, and the probability that he will play for the Chicago Bulls today is 0.4. If Michael Jordan plays, the probability that the Chicago Bulls will win today is 0.8. If Michael Jordan does not play, the probability that the Chicago Bulls will win today drops to 0.4. After the game, you are told that the Chicago Bulls won. What is the probability that Michael Jordan played today, given the outcome of the game?
12. Two people A and B each throw a fair die and the person throwing the larger face is declared the winner. If they throw the face with the same value, it is a draw. After a throw, person A is declared the winner. Given this result, what is the probability that person A threw a 6?
13. When you go to the market and buy a widget you are equally likely to buy one from either factory A or factory B. Widgets made by factory A have a length following a normal distribution with a mean of 100 cm and a standard deviation of 3 cm. The lengths of widgets made by factory B also follow a normal distribution, but with a mean of 101 cm and a standard deviation of 1 cm. You buy a widget and find it to be 102 cm long. What is the probability that it was manufactured in factory B?
14. Consider the probability density function $f_X(x) = \exp(-x)$, where X has nonzero density only for nonnegative values of x . Verify that $f_X(x)$ satisfies the properties of a probability density function. Find the mean, median, and variance of X .
15. Consider the probability density function $f_{(X,Y)}(x,y) = \exp(-x-y)$, which is nonzero only for $x, y \geq 0$. Compute $E[XY]$, μ_X , and μ_Y by setting up the appropriate integrals. Compute σ_{XY} and explain the sign of the covariance qualitatively using the properties of the joint distribution.
16. Show that the covariance of the random variables X and Y can never be larger than the arithmetic mean of the variances of X and Y .
17. Show that the expected value of the random variable described in Example 3.12 is infinity.
18. Let $f_X(x) = \exp(-x)$ be the density function of a non-negative random variable X . Find the density function of the minimum of two independent draws of this random variable. Find the density function of the maximum of two independent draws of this random variable. [Hint: Think in terms of cumulative distribution functions.]
19. **[Expectation of strictly convex functions]:** The text states that $E[G(X)] \neq G(E[X])$ for general functions (although equality holds for affine functions). A strictly convex function $G(x)$ is one that satisfies the following for all x_1, x_2 and $\lambda \in (0, 1)$:

$$G(\lambda x_1 + (1 - \lambda)x_2) < \lambda G(x_1) + (1 - \lambda)G(x_2)$$

Show that if $G(\cdot)$ is strictly convex, then $E[G(X)] > G(E[X])$. For which strictly convex function is the difference between the two equal to the variance of X ?

20. Let X be a nonnegative random variable with the probability density function $f_X(x) = \lambda \exp(-\lambda x)$. This variable is referred to as the exponential random variable with decay parameter λ . Compute the probability density function of (i) its translation $X + 2$; (ii) its scaling $3X$; and (iii) its reflection $-X$. State the bounds of the new random variable in each case.
21. A standard normal distribution with zero mean and variance 1 is defined by the following density function:

$$f_X(x) = \frac{\exp(-x^2/2)}{\sqrt{2\pi}}$$

Show that the sum Z of two independent variables drawn from a standard normal distribution has the following density function, which corresponds to a normal distribution with zero mean and variance 2:

$$f_Z(z) = \frac{\exp(-z^2/4)}{2\sqrt{\pi}}$$

You will find it helpful to show that the convolution of two standard normal variables is equal to the product of the above expression and the integration of a density function over $[-\infty, +\infty]$ (which evaluates to 1).

Challenge: Show that the sum of independent variables drawn from two normal distributions with different means (μ_1, μ_2) and variances (σ_1^2, σ_2^2) can be shown to follow a normal distribution with mean $(\mu_1 + \mu_2)$ and variance $(\sigma_1^2 + \sigma_2^2)$. You can find the density function for the normal distribution in Equation 1.3 of Chapter 1.

22. [Mathematical Games (1959)]: Three prisoners A, B, and C are condemned to be executed tomorrow. The governor has pardoned one of the three prisoners by drawing lots with equal probability. The jailer knows who is to be pardoned but is not allowed to inform individual prisoners of their fate. Prisoner A goes to the jailer and says, “We both know that at least one of the other two will be executed. Please at least tell me the name of one of the other prisoners to be executed without telling me about my fate.” The jailer tells him that B is to be executed. Hearing this, A feels very happy and assumes that his probability of being pardoned has now increased to 0.5. Show using Bayes theorem that A’s probability of being pardoned is still $1/3$, whereas C’s probability of being pardoned has increased to $2/3$. Assume that if both B and C were to be executed, the jailer picks one of them with equal probability while informing B.
23. [Tricky Bayes]: Consider the same scenario as Exercise 22, except that the governor dislikes prisoner B and rigs the draw of lots to stack it against prisoner B. The probabilities of A, B, and C being pardoned are now 0.495, 0.01, and 0.495, respectively. On being queried by the anxious prisoner A, the jailer says that B is to be executed (assuming that the jailer picks a prisoner with equal probability when both B and C are to be executed). Show that the probability of prisoner A being pardoned has now dropped to $1/3$ and that of C being pardoned has increased to $2/3$. [To understand the intuition, compare the frequency of B being reported by the jailer in cases where A is to be pardoned or not.]

- 24.** Suppose that you want to generate samples from a positive random variable X following the density function $f_X(x) = 4 \exp(-4x)$ for $x > 0$. You have available to you a piece of code that generates samples from the uniform distribution with density of 1 in $[0, 1]$ (and 0, otherwise). How can you use the uniformly generated sample to transform them to samples of X ? [Hint: See Example 3.29.]
- 25.** Let Θ be a random variable generated according to the distribution $f_\Theta(\theta) = 1$ for $\theta \in (0, 1)$. Subsequently, the random variable $X > 0$ is generated using the conditional distribution $f_{X|\Theta=\theta}(x) = \theta \exp(-\theta x)$. Find the unconditional distribution $f_X(x)$ of X .
- 26.** The random variable X is generated using the procedure of Exercise 25. The value of X is found to be 2.1. Find the posterior density function $f_{\Theta|X=2.1}(\theta)$. Given that you know that X was observed to be 2.1, what value of θ yields the maximum posterior density?
- 27.** Let Θ be a random variable generated according to the distribution $f_\Theta(\theta) = 1/\theta^2$ for $\theta \in (1, \infty)$. Subsequently, the random variable $X > 0$ is generated using the conditional distribution $f_{X|\Theta=\theta}(x) = \theta^2 x \exp(-\theta x)$. Find the unconditional distribution $f_X(x)$ of X .
- 28.** The random variable X is generated using the procedure of Exercise 27. The value of X is found to be 0.5. Find the posterior density function $f_{\Theta|X=0.5}(\theta)$. Given that you know that X was observed to be 0.5, what value of θ yields the maximum posterior density?
- 29.** Suppose that you have the code to generate the uniformly distributed random variable with the density $1/a$ in $[-a, a]$ and 0, otherwise. How would you use this code to generate the random variable X with density function $f_X(x) = 3x^2$ for $x \in [0, 1]$ and 0, otherwise?
- 30.** Suppose that you want to generate samples from a 2-dimensional random variable vector \vec{X} of positive values following the density function $f_{\vec{X}}(x_1, x_2) = 2 \exp(-2x_1 - x_2)$ for $x_1, x_2 > 0$. You have available to you a piece of code that generates independent and identically distributed (i.i.d.) samples from the uniform distribution with density of 1 in $[0, 1]$ (and 0, otherwise). How can you use the uniformly generated samples to transform them to samples of \vec{X} ? [Hint: This problem is similar to Exercise 24 in terms of using the result in Example 3.29. However, a key point is to investigate the independence of the components of \vec{X} along the two dimensions.]
- 31.** Let X be a random variable with the density function $f_X(x) = 5 \exp(-5x)$ for $x > 0$ (and 0, otherwise). What is the density function of the random variable $Y = -2X + 3$?
- 32.** You have two urns, each containing 5 balls. The first urn contains 2 green balls and 3 blue balls, whereas the second contains 3 green balls and 2 blue balls. You throw a fair die and select the first urn if the outcome is divisible by 3. Otherwise, you select the second urn. Then, you draw one ball out of the selected urn and find it to be blue. What is the probability that the first urn was selected?
- 33.** Repeat Exercise 32 for the case where the outcome of the experiment is a green ball.
- 34.** The probability density of the decay of radioactive element A at time $x > 0$ is $f_X(x) = 2 \exp(-2x)$. The probability density of the decay of radioactive element B at time

$y > 0$ is $f_Y(y) = 3 \exp(-3y)$. For a specific atom of element A and one of element B, what is the probability that the decay of element A occurs first?

35. Consider a box containing n atoms of element A and n atoms of element B. These elements are radioactive, and their radioactivity is defined in Exercise 34. Their decays are independent of one another. What is the probability that the first decay in the box is that of element A. [Hint: Derive the density function of the first decay of each of elements A and B.]
36. A physicist is trying to learn the decay behavior of an element. She knows that the time of decay $x > 0$ (in seconds) of an atom of A depends on the unknown decay parameter Θ . If the decay parameter were known to be λ , then the time of decay has the density function $f_{X|\Theta=\lambda}(x) = \lambda \exp(-\lambda x)$, where Θ is the parameter that she is trying to learn. Based on her past experience, she feels that the real-valued parameter Θ is likely to lie anywhere between 2 and 3 with equal probability. Find the unconditional probability density of the decay time an atom A by incorporating the physicist's past experience in the probability model.
37. Consider the setting in Exercise 36. The physicist starts observing a single atom and finds that it decays in 2.3 seconds. This observation changes the physicist's estimation of the density function of Θ from a uniform distribution in $(2, 3)$. Find this posterior distribution $f_{\Theta|X=2.3}(\lambda)$.
38. **Random Incidence:** Mad hatter runs a 24-hour bus company in which he decides the departure time of the next bus as follows. Just after each bus departs, he flips a fair coin. If the coin turns up heads, the next bus departs in 32 minutes. Otherwise, it departs in 64 minutes. What is the expected interval between a pair of consecutive departures? Suppose a customer arrives at a time that is uniformly distributed over the entire 24-hour period. Compute their expected waiting time and explain intuitively why it is greater than half the expected interval between consecutive bus departures.
39. Let $X_1 \dots X_n$ be n independent and identically distributed (i.i.d.) numeric random variables. Show that the expected squared distance between any pair is twice the variance of each X_i . In other words, show that $E[(X_i - X_j)^2] = 2\sigma_{X_i}^2$. How is this problem related to Exercise 19 of the previous chapter?
40. The position of an electron with quantum number n is modeled to lie on the X -axis with probability density $f_X(x) = 2 \sin^2(n\pi x/L)/L$ for $x \in (0, L)$, and 0, otherwise. Find the probability that an electron with quantum number $n = 2$ lies in $(0, L/3)$.
41. Let X and Y be independent random variables defined over $(0, 1)$ with density functions $f_X(x) = 3x^2$ and $f_Y(y) = 4y^3$. Find the probability $P(X^2 < Y)$.



Chapter 4

Probability Distributions

“All models are wrong but some are useful.” — George E. P. Box

4.1 Introduction

Several families of probability distributions arise repeatedly in various machine learning settings. We refer to these probability distributions as *families*, because they are defined in terms of parameters. In other words, a family of probability distributions is defined using a specific functional form of the probability density or mass function but with some parameters appearing as free (unspecified) variables. Specifying particular numeric values for the parameters defines an instantiation from this family. For example, the probability density might be uniformly distributed in $[a, b]$ with density $1/(b-a)$ in that range and 0, otherwise. The values a and b are parameters defining many properties of the distribution (such as its mean and dispersion), although leaving a and b unspecified results in the *family* of uniform distributions. Irrespective of the choice of parameters, there are also many properties (such as the basic shape of the distribution) that are specific to the distribution family at hand. Many of these families of probability distributions are created by generating processes that might seem simplistic at first glance but form the building blocks of models that can approximate the generating processes of real-world data quite well. Therefore, it is important to explore the properties of these important distributions.

The data observations that are collected in real-world applications are assumed to be samples of some unknown generating process, which is usually assumed to be too complex to model. The resulting distribution is sometimes thought of in terms of the (possibly infinite number of) underlying points, referred to as the *population*. Although the generating distribution of real-world data is typically too complex to model exactly, it is often sufficient to create simplified variants of these distributions. This goal can be achieved with the use of some key distribution families. The focus of this chapter is, therefore, on understanding the nature of these important distributions. These simplified distributions are used to model the

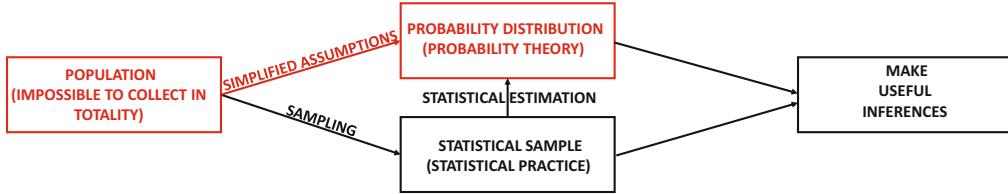


Figure 4.1: Revisiting Figure 1.6: How populations and samples interact with probability and statistics in machine learning — the portion in red corresponds to the modeling of real-world observed data with the types of simplified distributions discussed in this chapter.

real-world data while leaving some of the free parameters of the distributions unspecified initially. These parameters are then estimated to be as consistent with the observed data as possible. This *fitting* process is shown in Figure 4.1, and the highlighted part (in red) corresponds to the modeling of real-world observed data with simplified distributions.

4.1.1 Chapter Organization

This chapter is organized as follows. The uniform distribution is introduced in section 4.2. Section 4.3 introduces the Bernoulli distribution. The geometric distribution is described in section 4.5. The binomial distribution is introduced in section 4.6. A multidimensional generalization of the binomial distribution is the multinomial distribution, which is discussed in section 4.7. The exponential distribution is introduced in section 4.8. A closely related distribution to the exponential distribution is the Poisson distribution, which is discussed in section 4.9. The normal distribution is discussed in section 4.10. The Student's *t*-distribution is discussed in section 4.11. The χ^2 -distribution is discussed in section 4.12. Mixture distributions are introduced in section 4.13. An introduction to the notion of the moments of a distribution is provided in section 4.14. A summary is given in section 4.15.

4.2 The Uniform Distribution

The uniform distribution is the simplest of all continuous distributions. It samples a point uniformly at random from the real number line in the range $[a, b]$. The parameters a and b define the lower and upper bounds of the random variable drawn from the uniform distribution. The probability density function for a random variable drawn from the uniform distribution is as follows:

$$f_X(x) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

It is easy to verify that the density function integrates to an area of 1 over all values in its range:

$$\int_{x=a}^b f_X(x) dx = 1$$

The cumulative distribution function $F_X(x)$ is 0 for $x < a$ and 1 for $x > b$. The cumulative distribution function $F_X(x)$ for $x \in [a, b]$ is as follows:

$$F_X(x) = \int_{z=a}^x f_X(z) dz = \frac{x - a}{b - a}$$

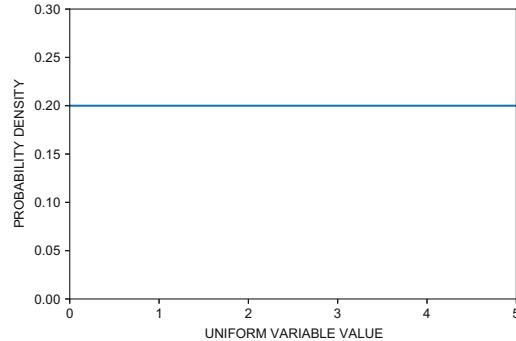


Figure 4.2: Example of the probability density a uniform distribution in the range [0, 5]

Therefore, the overall cumulative distribution function is defined as follows:

$$F_X(x) = \begin{cases} 0 & \text{if } x < a \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ 1 & \text{if } x > b \end{cases}$$

When a random variable X is drawn from the uniform distribution in the range $[a, b]$, it is denoted as follows:

$$X \sim U(a, b)$$

An example of the probability density of the uniform distribution in the range [0, 5] is shown in Figure 4.2. It is evident that the probability density is 0.2 at all points in [0, 5] so that the area under the probability density curve is exactly one unit.

Next, we derive expressions for the expected value and variance of random variables drawn from the uniform distribution. The expected value of a uniform random variable is as follows:

$$\begin{aligned} \mu_X = E[X] &= \int_{x=a}^b x f_X(x) dx = \frac{b^2 - a^2}{2(b-a)} \\ &= \frac{(b-a)(b+a)}{2(b-a)} = \frac{a+b}{2} \end{aligned}$$

Therefore, the mean is the mid-point of the range of the uniform random variable, which is not particularly surprising, since it is a symmetric distribution about the mid-point of its range. Note that the median and mean of the symmetric uniform distribution are both the same as the mean. In order to compute the variance, we first compute the second moment of the uniform random variable:

$$\begin{aligned} E[X^2] &= \int_{x=a}^b x^2 f_X(x) dx = \frac{b^3 - a^3}{3(b-a)} \\ &= \frac{(b-a)(b^2 + ab + a^2)}{3(b-a)} = \frac{a^2 + ab + b^2}{3} \end{aligned}$$

The variance of the uniform random variable can then be computed as a function of its first and second moments as follows:

$$\begin{aligned}\sigma_X^2 &= E[X^2] - (E[X])^2 = \frac{a^2 + ab + b^2}{3} - \frac{a^2 + 2ab + b^2}{4} \\ &= \frac{a^2 - 2ab + b^2}{12} = \frac{(b-a)^2}{12}\end{aligned}$$

Note that the standard deviation σ_X can be obtained by dividing the width of the range with $\sqrt{12}$:

$$\sigma_X = \frac{(b-a)}{\sqrt{12}}$$

Example 4.1 Let X be a uniformly distributed random variable in $[0, 1]$, and $Y = X^3$ be a derived random variable. Find the value of $P(Y \leq y)$ for $y \in (0, 1)$. Use this result to compute the density function $f_Y(y)$ of Y for $y \in (0, 1)$.

Solution: One can compute $P(Y \leq y)$ by expressing it in terms of X as follows:

$$P(Y \leq y) = P(X^3 \leq y) = P(X \leq y^{1/3})$$

The expression on the right is simply the cumulative distribution of the uniform random variable X evaluated at $y^{1/3}$. Using the formula for the cumulative distribution of a uniform random variable, it follows that $P(Y \leq y)$ is equal to $y^{1/3}$. Differentiating the cumulative distribution function, one can obtain the following density function for $y \in (0, 1)$:

$$f_Y(y) = \frac{1}{3y^{2/3}}$$

The density is 0 outside $(0, 1)$. ■

Example 4.2 Consider a random variable $Z \in (0, d)$ that is the sum of d i.i.d. uniform random variables X_1, \dots, X_d in $(0, 1)$. Show that the density function of Z in the lower-tail of the distribution in the region $(0, 1)$ is a smooth and increasing function $f_Z(z) = dz^{d-1}/2$ for $z \leq 1$. What happens for $d = 2$?

Solution: The joint distribution of (X_1, \dots, X_d) lies inside a unit cube of d dimensions. The cumulative distribution $F_Z(z)$ is a part of this larger cube that lies below $\sum_{i=1}^d x_i \leq z$ — this region turns out to be half of the d -dimensional cube with side z with triangular right-angled sides stuck to the origin (only for $z \leq 1$). The volume of this region is $F_Z(z) = z^d/2$. Differentiating the cumulative distribution function, one obtains the density function $f_Z(z) = dz^{d-1}/2$.

For $d = 2$, the density function is $f_Z(z) = z$ for $z \leq 1$. Using a similar argument to the above for $(1 - F_Z(z))$, one can also show that the density function is $(2-z)$ for $z \in (1, 2)$. Therefore, the density function for $d = 2$ is piecewise linear and takes on the form of a triangle. As d increases, the density function can be shown to become increasingly smooth, eventually resembling a bell-shaped curve. ■

Problem 4.1 Suppose that you have two independent uniform random variables in the range $[1, 4]$. What is the probability that the sum of the two variables is less than 3? You will find it helpful to draw a 2-dimensional sample space defined by the values of the two variables,

and geometrically examine what fraction of the sample space satisfies the aforementioned condition.

4.3 The Bernoulli Distribution

The Bernoulli distribution is one of the simplest discrete distributions with only two outcomes that are either 0 or 1. The value of 1 is sometimes designated as a “success” and a value of 0 is designated as a “failure.” The generating process for a Bernoulli random variable assumes that a biased coin is available for which one is the sides (the “success” side) has outcome probability p and the failure side has outcome probability $(1 - p)$. Since the coin is biased, the value of p may be different from 0.5. The success side maps to a value of 1 and the other side maps to a value of 0. Therefore, the sample space, Ω , for this experiment is $\{0, 1\}$, and the probability mass function for the Bernoulli distribution may be defined as follows:

$$p_X(x) = \begin{cases} p & \text{if } x = 1 \\ (1 - p) & \text{if } x = 0 \end{cases}$$

A compact way to write the probability mass function of the Bernoulli distribution for $x \in \{0, 1\}$ is as follows:

$$p_X(x) = p^x(1 - p)^{(1-x)}$$

When a random variable X is drawn from a Bernoulli distribution, it is denoted as follows:

$$X \sim \text{Bernoulli}(p)$$

Next, we examine important summary statistics of the Bernoulli distribution, such as the expected value and the variance. The expected value $E[X]$ of the Bernoulli random variable is as follows:

$$\mu_X = E[X] = p(1) + (1 - p)(0) = p$$

Interestingly, since $X^k = X$ for binary X and any integer $k > 1$, it can be shown that $E[X^k] = p$ irrespective of the value of k . In other words, the k th moment of the Bernoulli random variable is always p ; this result is helpful in computing the variance of the Bernoulli random variable. The variance σ_X^2 can be computed as follows:

$$\begin{aligned} \sigma_X^2 &= E[X^2] - (E[X])^2 \\ &= p - p^2 = p(1 - p) \end{aligned}$$

The Bernoulli process is one of the fundamental building blocks of other types of generating processes like those of the geometric distribution and the binomial distribution.

Example 4.3 Show that the variance of the Bernoulli distribution is maximized at $p = 0.5$. Give an intuitive explanation of this result.

Solution: The variance of the Bernoulli distribution is maximized when $p(1 - p)$ is maximized. This is a quadratic expression with a negative second-order coefficient and a negative second derivative. Therefore, the maximum value may be obtained by setting the derivative of the expression to 0:

$$1 - 2p = 0$$

The optimal value of p is 0.5.

When the value of p is close to 1, most trials are successes and there is little variability/dispersion in the results. A similar argument applies when p is close to 0, and most trials are not successes. The greatest variability occurs when p is 0.5 and there is an even mixture of both outcomes. This is the reason that the greatest variance is achieved at $p = 0.5$. ■

Example 4.4 You sample the Bernoulli success parameter $p \in (0, 1)$ from a distribution defined by the density function $f_\Theta(p) = 3p^2$. Subsequently, you perform a Bernoulli trial using this sampled parameter as the success probability. What is the probability that the trial is a failure? Now suppose you actually perform the trial and find the outcome X to be a failure. Use the Bayes rule in continuous hypothesis spaces to find the posterior density function $f_{\Theta|X=failure}(p)$. What is the mode of this density function and its significance?

Solution: The probability of failure can be found using the total probability rule in continuous hypothesis spaces:

$$p_X(\text{failure}) = \int_{p=0}^1 3p^2 \underbrace{p_X(\text{failure}|\Theta = p)}_{1-p} dp = \int_{p=0}^1 (3p^2 - 3p^3) dp = \frac{1}{4}$$

Next, using the Bayes rule in continuous hypothesis spaces, the posterior density function of the Bernoulli success parameter is as follows:

$$f_{\Theta|X=failure}(p) = \frac{f_\Theta(p)P(X = \text{failure}|\Theta = p)}{p_X(\text{failure})} = 12p^2(1-p)$$

In order to find the mode of the above density function, one can differentiate it with respect to p and set it to 0:

$$24p - 36p^2 = 0$$

Solving for p yields $p = 0$ and $p = 2/3$. However, since we want the maximum (mode), the second derivative needs to be negative and the values at the end points of the interval $[0, 1]$ also need to be checked. This occurs only at $p = 2/3$, which is the mode of the posterior distribution. This mode is the maximum a posteriori estimate of the Bernoulli success parameter with a single observation. ■

Problem 4.2 Consider a fair die that pays \$5 for each time the number 6 shows up and charges \$1, otherwise. Find the expected value and variance of the payoff.

Problem 4.3 Suppose that you have a simulator that generates uniformly distributed random variables in $[a, b]$. Discuss how you can use this simulator to generate samples of Bernoulli random variables with success probability p .

4.4 The Categorical Distribution

The categorical distribution is a generalization of the Bernoulli distribution. While the Bernoulli distribution generates only binary values, the categorical distribution generates

one of d categorical values. The categorical values are denoted by $1, 2, \dots, d$, although no ordering is assumed among these values. For example, the categorical values could represent colors painted on the face of a die. The probability of generating the category i is p_i , where $p_1 \dots p_d$ satisfy the following:

$$\sum_{i=1}^d p_i = 1$$

The categorical distribution is discrete since it is defined over a finite sample space of unordered outcomes indexed by $\{1, \dots, d\}$. The probability mass function of the categorical distribution for outcome $x \in \{1, \dots, d\}$ is as follows:

$$p_X(x) = p_x \quad \forall x \in \{1, 2, \dots, d\}$$

The categorical distribution with parameters p_1, p_2, \dots, p_d is denoted by the following:

$$X \sim \text{Categorical}(p_1, p_2, \dots, p_d)$$

Strictly speaking, there is a redundancy among the parameters since the probabilities of different categorical values sum to 1.

A categorical random variable X with d categories can be used to create d related Bernoulli random variables X_1, \dots, X_d :

$$X_i = \begin{cases} 1 & \text{if } X = i \\ 0 & \text{otherwise} \end{cases}$$

The mean and variance of the i th such Bernoulli distribution is as follows:

$$\begin{aligned} \mu_{X,i} &= p_i \\ \sigma_{X,i}^2 &= p_i(1 - p_i) \end{aligned}$$

It is not difficult to see that the categorical distribution is a direct generalization of the Bernoulli distribution. In fact, a categorical distribution with $d = 2$ can be shown to be equivalent to the Bernoulli distribution by treating one of the two outcomes as a success. Because of its relationship with the Bernoulli distribution, the categorical distribution is sometimes referred to as the generalized Bernoulli or the multinoulli distribution.

Example 4.5 Before going to work, John decides the color of his blazer as follows. He tosses a coin twice. He wears a blue blazer if he doesn't get any heads, a green blazer if he gets one head, and a pink blazer if he gets two heads. Write the PMF of the categorical distribution representing John's blazer color.

Solution: The sample space of coin tosses is $\Omega = \{HH, HT, TH, TT\}$. Therefore the probabilities of 0, 1, and 2 heads are 0.25, 0.5, and 0.25, respectively. The corresponding probability mass function is as follows:

$$p_X(x) = \begin{cases} 0.25 & \text{if } x = \text{Blue} \\ 0.5 & \text{if } x = \text{Green} \\ 0.25 & \text{if } x = \text{Pink} \end{cases}$$



Example 4.6 Before going to work, John creates probabilities p_1 and p_2 by sampling from the joint density function $f_{\vec{\Theta}}(p_1, p_2) = 192p_1p_2^2$ for $p_1, p_2 \in (0, 0.5)$. The density function is 0 outside the range $[0, 0.5] \times [0, 0.5]$. Then, he selects a blue, green, or pink blazer using a sample from the 3-dimensional categorical distribution with probability vector $[p_1, p_2, 1-p_1-p_2]$. What is the probability that John wears a pink blazer today?

Solution: Let X be a random variable denoting the color of John's blazer. One can compute the probability that John wears a pink blazer (i.e., $X = \text{Pink}$) using the total probability rule in continuous hypothesis spaces. In this case, one needs to integrate the probability density of a pink blazer over all valid probability vectors $[p_1, p_2, 1-p_1-p_2]$ defined by p_1 and p_2 . Therefore, we have the following:

$$\begin{aligned} P(X = \text{Pink}) &= \int_{p_1=0}^{0.5} \int_{p_2=0}^{0.5} f_{\vec{\Theta}}(p_1, p_2) \underbrace{p_{X|\vec{\Theta}=[p_1,p_2]}(\text{Pink})}_{1-p_1-p_2} dp_1 dp_2 \\ &= \int_{p_1=0}^{0.5} \int_{p_2=0}^{0.5} [192p_1p_2^2 - 192p_1^2p_2^2 - 192p_1p_2^3] dp_1 dp_2 \end{aligned}$$

On evaluating the above integral, one obtains the following:

$$P(X = \text{Pink}) = 1 - \frac{1}{3} - \frac{3}{8} = \frac{7}{24}$$

Furthermore, by using similar arguments, one can show that the probability of John wearing blue and green blazers are $\frac{1}{3}$ and $\frac{3}{8}$, respectively. ■

Example 4.7 John chooses his blazer color from blue, green, and pink using the probabilistic procedure described in Example 4.6. John was observed wearing a pink blazer today. Find the posterior density function $f_{\vec{\Theta}|X=\text{Pink}}(p_1, p_2)$. For what values of the parameters of the categorical distribution is this posterior density maximized?

Solution: One can use the Bayes rule in continuous hypothesis spaces to compute the posterior density function as follows:

$$f_{\vec{\Theta}|X=\text{Pink}}(p_1, p_2) = \frac{f_{\vec{\Theta}}(p_1, p_2)p_{X|\vec{\Theta}=[p_1,p_2]}(\text{Pink})}{P(X = \text{Pink})} = \frac{24 \times 192p_1p_2^2(1-p_1-p_2)}{7}$$

In order to find the optimal values of p_1 and p_2 , one can compute the partial derivatives of the posterior density w.r.t. p_1 and p_2 and set them to zero (to obtain two constraints in two variables). This yields several solutions, including $p_1 = 0$ and $p_2 = 0$. These critical points yield a density of 0, which is a minimum rather than a maximum. Another set of constraints obtained from the partial derivatives is as follows:

$$\begin{aligned} 1 - 2p_1 - p_2 &= 0 \\ 2 - 2p_1 - 3p_2 &= 0 \end{aligned}$$

The above system of equations yield $p_1 = 0.25$, $p_2 = 0.5$, and $1-p_1-p_2 = 0.25$. This solution provides the maximum density. ■

Problem 4.4 Let X_i be the 0-1 Bernoulli outcome for the i th outcome of a categorical distribution with parameters $p_1 \dots p_d$. We already know that the variance of X_i is $p_i(1-p_i)$. What is the covariance between X_i and X_j for $i \neq j$? What is the correlation? Why are these values negative?

4.5 The Geometric Distribution

The generating process of the geometric distribution uses repeated independent Bernoulli trials (with success probability p) until a success is achieved. Such trials are referred to as *independent and identically distributed (i.i.d.)* trials. The geometric random variable counts the number of trials to termination (including the final successful trial). How can one define this random variable in terms of an underlying sample space? One can treat the outcome of each experiment as a sequence of 0s (failures) followed by a 1 (success). It is noteworthy that the sample space Ω for this experiment is the infinite set $\{1, 01, 001, 0001, \dots\}$. The corresponding mapping to the geometric random variable is $\{1, 2, 3, 4, \dots, \infty\}$. Therefore, this is an example of a discrete numerical distribution that takes on one of an infinite number of possible values. This type of sample space is referred to as a *countably infinite sample space*, because a one-to-one correspondence can be found between the elements of the sample space and the natural numbers. On the other hand, continuous distributions, such as the uniform distribution, represent examples of *uncountably infinite sample spaces*.

In order for the geometric variable to take on the (integer) value of x , one must have $(x - 1)$ failures followed by a success. Since all trials are independent, the probability of this occurring is obtained by multiplying the probability of $(x - 1)$ failures followed by the multiplication with the probability of a success. In other words, the probability mass function of the geometric random variable may be written as follows:

$$p_X(x) = (1 - p)^{(x-1)} p \quad \forall x \in \{1, 2, \dots, \infty\}$$

The fact that a random variable is drawn from the geometric distribution with parameter p is expressed as follows:

$$X \sim G(p)$$

The sum of the probabilities of all possible outcomes of the geometric random variable can be shown to be 1 by using the algebraic formula for the summation of a geometric series:

$$\sum_{x=1}^{\infty} p_X(x) = \sum_{x=1}^{\infty} (1 - p)^{(x-1)} p = \frac{p}{1 - (1 - p)} = 1$$

Next, we explore the computation of the mean and variance of the geometric random variable. The expected value of a geometric random variable X may be computed as follows:

$$\mu_X = E[X] = \sum_{x=1}^{\infty} x \cdot p_X(x) = \sum_{x=1}^{\infty} x \cdot (1 - p)^{(x-1)} p = \sum_{x=1}^{\infty} x \cdot [(1 - p)^{(x-1)} - (1 - p)^x]$$

It turns out that the alternating terms of different signs in the infinite series above can be grouped differently. Instead of grouping $x(1 - p)^{x-1}$ with the negative term $-x(1 - p)^x$, we can group it with the previous negative term $-(x - 1)(1 - p)^{(x-1)}$. Therefore, we have:

$$\mu_X = E[X] = \sum_{x=1}^{\infty} (1 - p)^{(x-1)} [x - (x - 1)] = \sum_{x=1}^{\infty} (1 - p)^{(x-1)} = 1 / (1 - (1 - p)) = 1/p$$

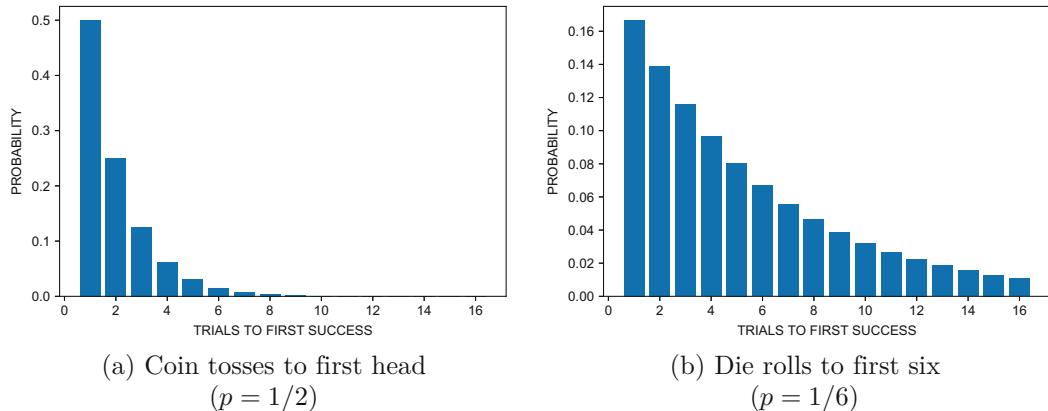


Figure 4.3: PMFs of different geometric distributions at varying p . The probabilities of only the first 16 outcomes are shown.

In order to compute the variance, one can first compute $E[X^2]$ as follows:

$$\begin{aligned}
 E[X^2] &= \sum_{x=1}^{\infty} x^2 \cdot p_X(x) = \sum_{x=1}^{\infty} x^2 \cdot (1-p)^{(x-1)} p = \sum_{x=1}^{\infty} x^2 \cdot [(1-p)^{(x-1)} - (1-p)^x] \\
 &= \sum_{x=1}^{\infty} (1-p)^{(x-1)} [x^2 - (x-1)^2] = (2x-1)(1-p)^{(x-1)} \quad [\text{Grouping terms differently}] \\
 &= 2 \left[\frac{d}{dp} \right] \left(\sum_{x=1}^{\infty} -(1-p)^x \right) - \sum_{x=1}^{\infty} (1-p)^{x-1} = 2 \left[\frac{d}{dp} \right] \left(-(1-p)/p \right) - 1/p \\
 &= 2/p^2 - 1/p
 \end{aligned}$$

The variance is then computed as follows:

$$\sigma_X^2 = E[X^2] - (E[X])^2 = 2/p^2 - 1/p - 1/p^2 = (1-p)/p^2$$

The geometric distribution arises in settings where one is trying to count the number of attempts required to achieve a particular goal. Examples of the probability values of the first 16 values of the PMF are shown in Figure 4.3 for two different geometric distributions — the first is for a fair coin with success probability $p = 1/2$ (where a success is defined as a head), and the second is for a die roll with success probability $p = 1/6$ (where a success is defined as an outcome of six). It is evident that the probability values for increasing x decay rapidly as each toss has a relatively high probability of termination. On the other hand, the probability decay in the case of die rolls is much slower, as termination (i.e., face value of six) is less likely for each roll.

A random variable from the geometric distribution can also be viewed from a temporal perspective by assuming that the Bernoulli events occur at discrete clock ticks spaced at intervals of length 1. The geometric random variable can then be interpreted as the time to the first success. From this perspective, the geometric distribution has a very useful property, referred to as the *memoryless property*.

Observation 4.1 (Memoryless Property of Geometric Distribution) *The geometric distribution can be interpreted as the arrival time of a “success” event on performing*

repeated Bernoulli trials at successive clock ticks. If a success event is known to not have occurred at the stroke of clock tick t_0 , the conditional distribution of the remaining time to arrival (from clock tick t_0) is the same as the unconditional PMF of the time to arrival (at clock tick 0).

The reason that the memoryless property holds for geometric distributions is that successive Bernoulli trials are independent. Therefore, the outcomes of previous events have no bearing on the remaining number of trials to success.

Example 4.8 You throw a pair of dice together until you hit a double-six. You count your total number of throws, including the final, successful one. This number is a discrete random variable. What is the mean and variance of this random variable?

Solution: This random variables has a geometric distribution with success probability $p = 1/36$. The mean of this distribution is $1/p = 36$ and the variance is $(1 - p)/p^2 = 1260$. ■

Example 4.9 (Alternative Derivation of Expectation) Let Z_i for $i \in \mathbb{N}$ be a 0-1 random variable, which is 1 if a geometric process with success probability p has not terminated sometime strictly before the i th trial and 0, otherwise. What is the PMF of Z_i ? Express the underlying geometric random variable X as a function of the different values of Z_i and use it to compute $E[X]$.

Solution: The random variable Z_i has a Bernoulli distribution with probability $(1 - p)^{i-1}$ for all $i \geq 2$. For $i = 1$, the probability is 0. The geometric variable X can be expressed in terms of the different values of Z_i as follows:

$$X = 1 + \sum_{i=2}^{\infty} Z_i$$

The above relationship holds because Z_i is 1 for $i \geq 2$ if and only if the i th trial is needed in the geometric process. A value of 1 is added for the first trial. All other values of Z_i are 0. Taking the expectation of both sides and using $E[Z_i] = (1 - p)^{i-1}$, one obtains the following:

$$X = 1 + \sum_{i=2}^{\infty} (1 - p)^{i-1} = 1/p$$

Example 4.10 You sample the geometric success parameter $p \in (0, 1)$ from the density function $f_{\Theta}(p) = 3p^2$. Subsequently, you generate a random variable X from the geometric distribution using this sampled parameter as the success probability. What is the probability that $X = 3$? Now suppose you actually sample the geometric random variable and find the outcome X to be 3. Find the posterior density function $f_{\Theta|X=3}(p)$. At what value of p is the posterior density function maximized?

Solution: The probability of $X = 3$ can be found using the total probability rule in continuous hypothesis spaces:

$$p_X(3) = \int_{p=0}^1 3p^2 \underbrace{p_{X|\Theta=p}(3)}_{(1-p)^2 p} dp = \int_{p=0}^1 3p^3(1-p)^2 dp = 3 \frac{3!2!}{6!} = \frac{1}{20}$$

The above definite integral is a standard result for *beta functions* (which can also be shown by integrating by parts). Next, using the Bayes rule in continuous hypothesis spaces, the posterior density function of the geometric success parameter is as follows:

$$f_{\Theta|X=3}(p) = \frac{f_{\Theta}(p)p_{X|\Theta=p}(3)}{p_X(3)} = 60p^3(1-p)^2$$

In order to find the mode of the above density function, one can differentiate it with respect to p and set it to 0:

$$60p^2(1-p)[3(1-p) - 2p]$$

Solving for p yields $p = 0$, $p = 1$, and $p = 3/5$. However, since we want the maximum (mode), the second derivative needs to be negative. This occurs only at $p = 3/5$, which is the mode of the posterior distribution. This mode is the maximum a posteriori estimate of the geometric success parameter, given the fact that $X = 3$ was observed. ■

Problem 4.5 (Cumulative Distribution and Median) Compute an expression for the cumulative distribution function of the geometric process with success probability p . Show that the median of the geometric distribution is $\lceil -\log(2)/\log(1-p) \rceil$ in cases when the expression inside the ceiling operator is not an integer. Otherwise, the median is $\log(2)/\log(R) + 0.5$ via interpolation tie-breaking. Try different values of p to check whether the mean or the median is greater. Is a geometric distribution left skewed or right skewed?

Problem 4.6 Jane will score 1600 on the SAT with probability 0.5. Her outcome in each exam is independent of the others. What is the expected number of attempts she will require to score 1600? Obviously, it is not possible to make an unlimited number of attempts to reach 1600. Therefore, Jane uses the mean and standard deviation of the underlying geometric distribution as guidelines to limit her attempts. She decides to terminate the moment she exceeds one standard-deviation beyond the expected number of attempts to reach 1600 (unless she reaches 1600 earlier). What is the expected number of attempts to termination?

4.6 The Binomial Distribution

The binomial distribution is another well-known derivative of the Bernoulli process. While the geometric random variable counts the number of trials till the first success, the binomial random variable counts the number of successes in a fixed number n of trials. As in the case of the Bernoulli random variable, the probability of a success is assumed to be p . Therefore, the parameters of the binomial distribution are n and p . Both the binomial and geometric processes use repeated trials but use different termination criteria, which results in a different sample space. The sample space Ω of the underlying generating process contains all 2^n bitstrings of length n . The i th bit in this bitstring is indicative of a success or failure in the i th trial, where a value of 1 indicates a success. For example, the sample space of size

2^3 for $n = 3$ is as follows:

$$\Omega = \{000, 001, 010, 011, 100, 101, 110, 111\}$$

The binomial random variable maps outcomes from the sample space to numerical values by counting the total number of successes in the n trials (i.e., number of 1s in the bitstring) and reporting it as the realized value.

How can one compute the PMF of the binomial distribution? This is achieved by adding the probabilities of all the outcomes with a particular number of successes in the sample space. The number of outcomes in the sample space with x successes is given by $\binom{n}{x}$ and each such outcome has probability $p^x(1-p)^{(n-x)}$; each outcome probability is calculated by using the fact that the different trials are independent and the probabilities of x successes need to be multiplied with the probabilities of $(n - x)$ failures. Therefore, the probability of x successes is as follows:

$$p_X(x) = \binom{n}{x} p^x (1-p)^{(n-x)}$$

The fact that a random variable X is drawn from a binomial distribution with parameters n and p is denoted as follows:

$$X \sim \text{B}(n, p)$$

The sum of the probabilities of all possible outcomes of a binomial random variable can be shown to be 1 by using the algebraic properties of the binomial expansion:

$$\sum_{x=0}^n p_X(x) = \sum_{x=0}^n \binom{n}{x} p^x (1-p)^{(n-x)} = (p + (1-p))^n = 1$$

The binomial variable can be expressed as a sum of n independent and identically distributed (i.i.d.) Bernoulli trials. If Z_i is the 0-1 random variable indicating the outcome of the i th Bernoulli trial with success probability p , the binomial random variable $X \sim \text{B}(n, p)$ can be expressed in terms of the different Z_i as follows:

$$X = \sum_{i=1}^n Z_i$$

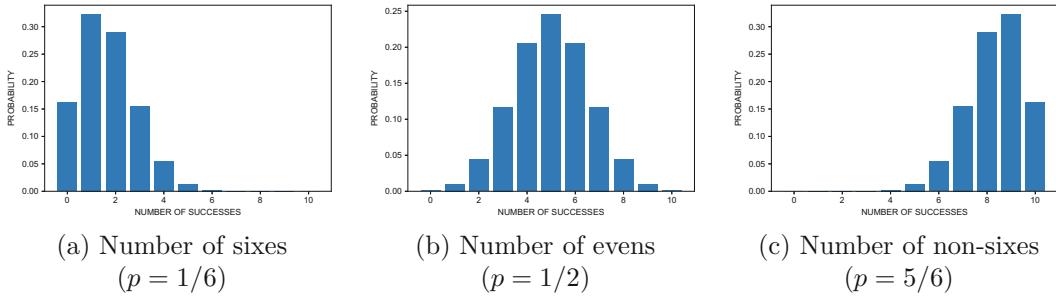
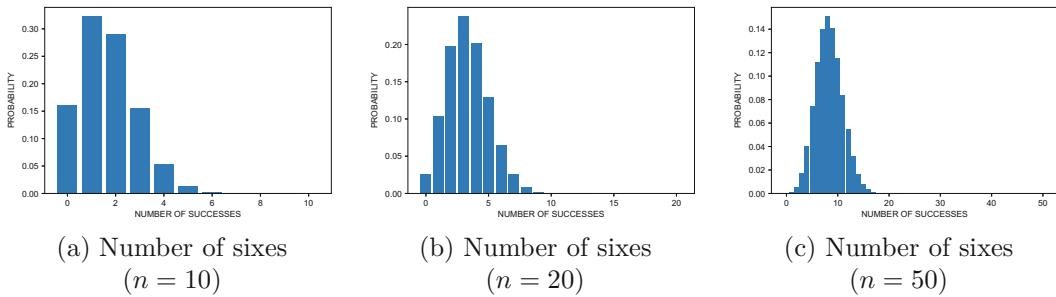
This relationship is useful for computing the expectation and variance of the binomial random variable:

$$E[X] = \sum_{i=1}^n E[Z_i] = n \cdot p$$

Furthermore, since the different values of Z_i are independent, the variance of X is the sum of the variances of the different values of Z_i :

$$\sigma_X^2 = \sum_{i=1}^n \sigma_{Z_i}^2 = n \cdot p(1-p)$$

Examples of the probability mass function for the binomial distribution for $n = 10$ and different values of p are shown in Figure 4.4. The leftmost distribution in Figure 4.4(a) corresponds to the probability mass function of the number of times a six is obtained in 10 die throws and the rightmost distribution in Figure 4.4(c) corresponds to the probability

Figure 4.4: PMFs of different binomial distributions at fixed $n = 10$ and varying p Figure 4.5: PMFs of different binomial distributions at fixed $p = 1/6$ and varying n

mass function of the number of times something other than a six is obtained in 10 die throws. The middle distribution in Figure 4.4(b) corresponds to the probability mass function of the number of times an even numbered face shows up. It is clear that values of p different from 0.5 lead to skewed distributions.

An interesting pattern is observed as one increases the value of n while keeping the value of p fixed. As the value of n increases, the distribution becomes increasingly symmetric (even for values of $p \neq 0.5$) and the distribution starts resembling a bell-shaped curve. Examples of the probability mass function for $p = 1/6$ and different values of n are shown in Figure 4.5. As we will see later, the increasing level of symmetry in the distribution with increasing n is a consequence of one of the most fundamental laws in statistics, referred to as the *central limit theorem* (cf. section 5.2 of Chapter 5).

Example 4.11 Suppose you throw a pair of dice together a total of 20 times. How many double sixes do you expect? Compute the probability of at most two double sixes in these 20 throws.

Solution: The distribution of the number of doubles sixes is binomial with parameters $n = 20$ and $p = 1/36$. The expected number of double is $20/36 = 5/9$. The probability of 0, 1, or 2 double sixes is as follows:

$$\begin{aligned} P(X \leq 2) &= \left(\frac{35}{36}\right)^{20} + \binom{20}{1} \left(\frac{1}{36}\right)^1 \left(\frac{35}{36}\right)^{19} + \binom{20}{2} \left(\frac{1}{36}\right)^2 \left(\frac{35}{36}\right)^{18} \\ &= 0.56926 + 0.32529 + 0.08829 \approx 0.983 \end{aligned}$$

The above result implies that it is very rare to get more than two double sixes in 20 throws. ■

Example 4.12 You throw a fair die 10 times. What is the probability of an odd number of sixes? Now repeat this calculation for a loaded die in which the probability of a six is 0.1.

Solution: One possibility would be to compute the probabilities of 1, 3, 5, 7, 9 sixes and then add them. However, this would be a cumbersome calculation. A key observation is that these binomial probabilities are the terms containing odd powers of p in the expansion of $((1-p) + p)^n$ where $p = 1/6$ and $n = 10$. These terms can be teased out by observing that the expansion of $((1-p) - p)^n$ is different from that of $((1-p) + p)^n$ only in the sense that the former includes the terms containing odd powers of p with a negative sign. Therefore, the required probability for a fair die is as follows:

$$\begin{aligned} P(\text{Odd number of sixes}) &= \frac{((1-p) + p)^n - ((1-p) - p)^n}{2} \\ &= \frac{1 - (1 - 2p)^n}{2} = \frac{1 - (2/3)^{10}}{2} \approx 0.4913 \end{aligned}$$

For a loaded die with $p = 0.1$, this calculation yields a probability of approximately 0.4463. ■

Problem 4.7 Compute the expected first and second moment of the binomial random variable via the summations $\sum_{x=0}^n x p_X(x)$ and $\sum_{x=0}^n x^2 p_X(x)$. Along with the standard algebraic form of the binomial expansion of $[p + (1-p)]^n$, two useful results for completing this exercise are $x(n)_x = n(n-1)_x$ and $x(x-1)(n)_x = n(n-1)(n-2)_x$.

Problem 4.8 Use the computed first and second moments in the previous exercise to derive the expected value and variance of the binomial random variable.

Problem 4.9 If you throw a fair die 12 times, what is the expected number of sixes? What is the probability of obtaining at least two sixes? What is the probability of obtaining at least 3 sixes? Is the mean or the median greater in the distribution of the number of sixes? Is the distribution left-skewed or right-skewed?

4.7 The Multinomial Distribution

The multinomial distribution is an example of a *discrete multivariate distribution* of numeric values. Just as the binomial distribution is a generalization of the Bernoulli distribution, the multinomial distribution is a generalization of the categorical/multinoulli distribution. A coin toss allows two outcomes, and a binomial distribution captures the probability of the number of times x that a particular coin face shows up in n tosses. Similarly, the multinomial distribution captures the probability of a vector of d frequencies, where d is the number of categories. These d numeric outcomes are the number of occurrences of each category in n trials — therefore, the counts of the d -dimensional vector must sum to n . The probabilities

of the d different outcomes are denoted by $p_1 \dots p_d$. The sample space Ω of the underlying generating process contains all d^n strings of length n . The i th symbol in this string is the outcome of the i th trial, and the symbol indicates the value of the corresponding outcome. For example, the sample space corresponding to three throws of a six-sided die is as follows:

$$\Omega = \{ijk : i, j, k \in \{1 \dots 6\}\}$$

How are the string outcomes from this sample space of length- n strings mapped to 6-dimensional numeric counts? The generating process of the multinomial distribution counts the total number of times each of the $d = 6$ faces occurs over $n = 3$ trials, and reports a 6-dimensional vector of counts. For example, the outcome string 334 corresponds to the face 3 showing up twice and the face 4 showing up once. The corresponding vector of counts is therefore $[0, 0, 2, 1, 0, 0]$. Note that this vector of counts must always sum to $n = 3$, since one is counting faces over n trials. Since the elements of the d -dimensional vector must add to n , the joint probability distribution will not exhibit attribute independence. A multinomial distribution over $d = 2$ dimensions simplifies to the binomial distribution in which the frequencies of both coin sides are tracked simultaneously.

Consider the d -dimensional numeric vector $[x_1, x_2, \dots, x_d]$ denoting the vector outcome from the d -dimensional multinomial distribution. Since the i th outcome must occur x_i times, the different ways of arranging the different outcomes over the n trials are obtained by first selecting x_1 positions from n positions, then x_2 positions from the remaining $(n - x_1)$ positions, and so on. Therefore, the total number of ways of selecting the different positions is given by the following:

$$\begin{aligned} \text{Number of ways of obtaining } [x_1 \dots x_d] &= \prod_{i=1}^d \binom{n - \sum_{j=1}^{i-1} x_j}{x_i} \\ &= \frac{n!}{\prod_{i=1}^d x_i!} \end{aligned}$$

The probability of each of these specific outcomes is given by $\prod_{i=1}^d p_i^{x_i}$. Since all the outcomes in the sample spaces are mutually exclusive, their probabilities can be added to obtain the probability of the vector outcome $[x_1, \dots, x_d]$ for the multinomial random variable \vec{X} :

$$p_{\vec{X}}(x_1, x_2, \dots, x_d) = \frac{n!}{\prod_{i=1}^d x_i!} \prod_{i=1}^d p_i^{x_i}$$

The above expression for the PMF is slightly incomplete as it fails to specify that the sum of the counts over the d possibilities must be n in order for the above condition to hold. A more complete expression for the PMF is as follows:

$$p_{\vec{X}}(x_1, x_2, \dots, x_d) = \begin{cases} \frac{n!}{\prod_{i=1}^d x_i!} \prod_{i=1}^d p_i^{x_i} & \text{if } \sum_{i=1}^d x_i = n \\ 0 & \text{otherwise} \end{cases}$$

The multinomial distribution has $(d + 1)$ parameters, which are the number of trials n and the probabilities $p_1 \dots p_d$ of the parameters. There is some redundancy in parameter specification because the sum of the different values of p_i is always 1. The fact that a random variable \vec{X} is drawn from a multinomial distribution with parameters n and p_1, p_2, \dots, p_d is denoted as follows:

$$\vec{X} \sim \text{Multinomial}(n, p_1, p_2, \dots, p_d)$$

Setting $n = 1$ for arbitrary d results in the categorical (i.e., multinoulli) distribution. Setting $d = 2$ for arbitrary n results in the binomial distribution, except that the distribution is presented as a 2-dimensional joint distribution corresponding to frequencies of both outcomes (that are redundant with respect to each other since they sum to n). Another connection of the multinomial distribution with the binomial distribution is that the 1-dimensional marginal probability mass function of the multinomial distribution on the i th attribute is the binomial distribution with parameters n and p_i . It is easy to show this fact by simply observing that the attribute of interest (i.e., i th outcome of underlying categorical process) can be considered the “success” outcome and the other outcomes can be aggregated together into a single outcome referred to as the “failure” outcome. In such a case, the 1-dimensional probability mass function of the i th attribute is the same as the probability mass function of the number of successes of the binomial distribution with parameters n and p_i because we are only counting the i th categorical outcome as the success outcome.

The fact that the marginal probability mass function of the i th attribute of the multinomial distribution is a binomial distribution with parameters n and p_i implies that the expected value of the i th attribute is $n \cdot p_i$ and the variance of the i th attribute is $n \cdot p_i(1 - p_i)$:

$$\begin{aligned} E[\vec{X}] &= n [p_1, p_2, \dots, p_d] \\ \sigma^2_{\vec{X}} &= n [p_1(1 - p_1), p_2(1 - p_2), \dots, p_d(1 - p_d)] \end{aligned}$$

Since the attributes are not independent, the variance of the sum of the attributes is not equal to the sum of the variances of the attributes. In fact, since the components of \vec{X} always sum to the fixed value of n , the variance of the sum of attributes is 0.

The multinomial distribution plays an important role in machine learning, because it is often used to model the counts of words in documents drawn from a particular topic in text modeling. It is assumed that each word in the document is generated with the throw of a d -sided die, where d is the number of words in the lexicon. Even though the generating process of word counts is obviously not based on die rolls, this simple model often provides excellent results in many applications.

Example 4.13 Suppose that you simulate a three-sided die from a six-sided die by remapping outcomes from a six sided die by mapping any outcome $x \in \{1, \dots, 6\}$ of the six-sided die to the integer $\lfloor (x + 1)/2 \rfloor \in \{1, 2, 3\}$. Such a three-sided die is thrown four times. A joint distribution is created on the frequencies of the three faces in four throws. Compute the probabilities of all outcomes for the face frequencies in four throws of the three-sided die in which the (redefined) face 1 occurs exactly one out of four times. Compute the sum of these probabilities. Justify how you can arrive at this sum more simply by using the binomial distribution.

Solution: The four die throws define a 3-dimensional multinomial distribution in which the value of n is 4, and each of the three probabilities p_1, p_2, p_3 is $1/3$. This because exactly two faces of the six-sided map to one face of the three-sided die. The frequency outcomes from four throws are 3-dimensional tuples like $(1, 2, 1)$ in which the i th element is the frequency of the i th face — these faces must sum to 4. Furthermore, we need to find the probabilities of only those tuples in which the first

element is 1. Using the multinomial distribution formula, we obtain the following:

$$\begin{aligned} P(1, 0, 3) &= \frac{4!}{1!0!3!} \left(\frac{1}{3}\right)^4 & P(1, 1, 2) &= \frac{4!}{1!1!2!} \left(\frac{1}{3}\right)^4 \\ P(1, 2, 1) &= \frac{4!}{1!2!1!} \left(\frac{1}{3}\right)^4 & P(1, 3, 0) &= \frac{4!}{1!3!0!} \left(\frac{1}{3}\right)^4 \end{aligned}$$

These probability values in order are $4/81$, $12/81$, $12/81$, and $4/81$. The sum of these values is $32/81$.

One can also model this problem using a binomial distribution in which a success (face value of 1) has probability $1/3$ and a failure (face value of 2 or 3) has probability of $2/3$. The aforementioned sum is equal to the probability that there is one success and three failures in four throws. The corresponding binomial probability is as follows:

$$P(\text{one success from four throws}) = \binom{4}{1} \left(\frac{1}{3}\right)^1 \left(\frac{2}{3}\right)^3 = \frac{32}{81}$$

■

Example 4.14 A gambling casino devises a game in which three fair dice are thrown simultaneously. The gambler is paid a dollar amount equal to the product of the face values if it is greater than 130. Otherwise, nothing is paid. What is the minimum amount the casino should charge for an attempt in order to not run at a loss (in expectation)?

Solution: A value greater than 130 can be arrived as $6 \times 6 \times 6$, $6 \times 6 \times 5$, $6 \times 6 \times 4$, or $6 \times 5 \times 5$. We need to find the probability of each possibility and compute its expected value. Let R be the random variable indicating the gambler's reward. Then, we have the following:

$$\begin{aligned} E[R] &= \frac{3!}{3!(0!)^5} \left(\frac{1}{6}\right)^3 6^3 + \frac{3!}{2!1!(0!)^4} \left(\frac{1}{6}\right)^3 6^2 \cdot 5 + \frac{3!}{2!1!(0!)^4} \left(\frac{1}{6}\right)^3 6^2 \cdot 4 + \\ &\quad + \frac{3!}{2!1!(0!)^4} \left(\frac{1}{6}\right)^3 6 \cdot 5^2 \\ &= 1 + 3 \left(\frac{5}{6}\right) + 3 \left(\frac{2}{3}\right) + 3 \left(\frac{5}{6}\right)^2 \approx 7.583 \end{aligned}$$

Therefore, the casino needs to price each play at a minimum of \$7.59 in order to break even.

■

Problem 4.10 Suppose you roll a fair die 5 times. What is the probability that you obtain two fours, two fives, and one outcome of something other than a four or five?

Problem 4.11 Let $\vec{X} = [X_1, \dots, X_d]$ be a multinomial random variable with parameters n and p_1, p_2, \dots, p_d for $d > 2$. Discuss why the random variable $(X_i + X_j)$ has a binomial distribution (by defining a related generating process to that of the multinomial distribution). What are the parameters of this binomial distribution? Find the variance of the random variables X_i , X_j , and $(X_i + X_j)$. Use these variances to compute the covariance

between X_i and X_j . What is the sign of the covariance? Give a qualitative explanation for the sign of the covariance.

4.8 The Exponential Distribution

It is easiest to think of the exponential random variable as a time variable, and therefore the random variable will be denoted by T — its instantiated value will be denoted by t . The exponential distribution can be viewed as a variable denoting the time for an event of interest to occur (e.g., a bus arriving at the stop you have been waiting). The exponential distribution uses a parameter $\lambda > 0$, which is referred to as the *arrival rate*, and the value of the random variable corresponds to the time to arrival from the beginning of the wait.

The generating process of the random variable is defined assuming that repeated Bernoulli trials with success probability $\delta_p = \lambda \cdot \delta_t$ occur over successive time intervals of infinitesimal length δ_t — the value of the exponential random variable is then set to the time elapsed to the first success. Therefore, the exponential random variable is related to the discrete geometric distribution that also performs repeated Bernoulli process to termination at discrete intervals of length 1. In other words, *the exponential distribution can be thought of as a continuous variant of the geometric distribution* with t/δ_t trials of infinitesimally small success probability $\lambda\delta_t$ occurring in a finite period t (until a success is reached).

In order to derive an expression for the probability density of the exponential random variable, we will first compute the cumulative probability that the first success occurs before time t . The number of independent trials over a period t by breaking it up into intervals of length δ_t is given by t/δ_t . The probability of no success in each such interval is given by $(1 - \lambda \cdot \delta_t)$. Therefore, the probability of no success before time t over all independent t/δ_t trials is given by the following:

$$P(\text{No success in } t) = \lim_{\delta_t \rightarrow 0^+} (1 - \lambda \cdot \delta_t)^{t/\delta_t}$$

One can also write the above probability in terms of the number of trials $m = t/\delta_t$ as follows:

$$P(\text{No success in } t) = \lim_{m \rightarrow \infty} (1 - \lambda t/m)^m = \exp(-\lambda t)$$

The above derivation uses the well-known simplification from mathematical analysis that the expression $(1 - a/m)^m$ converges to $\exp(-a)$ as m goes to ∞ . The cumulative distribution function $F_T(t) = P(T \leq t)$ is equal to the complement of the probability of no success:

$$F_T(t) = 1 - P(\text{No success in } t) = 1 - \exp(-\lambda t)$$

The probability density function is obtained by differentiating the cumulative distribution function with respect to t to obtain the following:

$$f_T(t) = \lambda \exp(-\lambda t) \quad [\text{for } t \geq 0]$$

The exponential distribution is illustrated in Figure 4.6 at two different arrival rates of $\lambda = 1$ and $\lambda = 2$. The higher arrival rate of $\lambda = 2$ leads to a higher initial probability density, but also leads to quicker decay of the probability density as successes are typically achieved earlier on. It is instructive to compare the pattern of the probability densities for the exponential distribution (Figure 4.6) with the probability mass functions for the geometric distribution (Figure 4.3). Just as higher arrival rates lead to quicker decay of probabilities for the exponential distribution, higher success probabilities lead to quicker

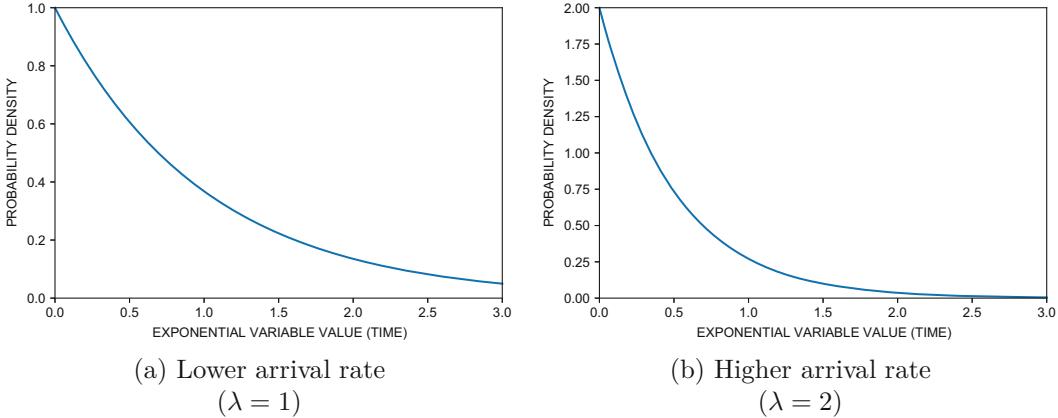


Figure 4.6: PDFs of different exponential distributions at varying arrival rates. Higher arrival rates lead to earlier success and faster decay of probabilities.

decay of probabilities for the geometric distribution. This is not particularly surprising as the geometric and exponential distributions are both modeled in terms of “time to termination” of repeated Bernoulli trials, with the difference being that the Bernoulli distribution performs trials at discrete clock ticks, whereas the exponential distribution performs trials at infinitesimally separated clock ticks.

The expected value of the exponential random variable may be computed as follows:

$$\begin{aligned} E[T] &= \int_{t=0}^{\infty} t f_T(t) dt = \lambda \int_{t=0}^{\infty} t \cdot \exp(-\lambda t) dt \\ &= [-t \cdot \exp(-\lambda t)]_0^{\infty} + \int_0^{\infty} \exp(-\lambda t) dt \quad [\text{Integrating by parts}] \\ &= [-t \cdot \exp(-\lambda t) - \exp(-\lambda t)/\lambda]_0^{\infty} = 1/\lambda \end{aligned}$$

In other words, *the expected arrival time of the exponential process is the inverse of the arrival rate*. The variance of the exponential random variable may be derived by first computing its second moment as follows:

$$\begin{aligned} E[T^2] &= \int_{t=0}^{\infty} t^2 f_T(t) dt = \lambda \int_{t=0}^{\infty} t^2 \cdot \exp(-\lambda t) dt \\ &= [-t^2 \cdot \exp(-\lambda t) - 2t \cdot \exp(-\lambda t)/\lambda - 2\exp(-\lambda t)/\lambda^2]_0^{\infty} = 2/\lambda^2 \end{aligned}$$

The above integral is also performed by integrating by parts, although some of the details have been omitted. One can then compute the variance of the random variable T as follows:

$$\sigma_T^2 = E[T^2] - (E[T])^2 = 2/\lambda^2 - 1/\lambda^2 = 1/\lambda^2$$

By taking the square-root of the variance, we obtain the standard deviation $\sigma_T = 1/\lambda$.

The exponential distribution is useful for modeling various types of processes in the real world (such as radioactive decay or the kinetics of first-order chemical reactions). Since the trials over successive infinitesimal intervals are independent of one another, the exponential variable follows the *memoryless property*, which is similar to the case of the geometric distribution. The memoryless property refers to the fact that the conditional distribution of the remaining time to arrival at any given moment of time t_0 (given that an arrival has

not yet occurred) is the same as the unconditional distribution of the time to arrival at the beginning of the process.

Example 4.15 (Median and Half Life) Use the cumulative distribution function to derive an expression for the median of the exponential distribution in terms of the arrival rate λ . Interpret the median time in terms of the expected number of decays occurring in a box containing N radioactive atoms. Radioactive decay times are exponentially distributed.

Solution: Setting $F_T(t) = 0.5$, one obtains $t = \ln(2)/\lambda$. This value represents the half-life of a radioactive atom because the probability of a decay occurring in this time is 0.5. Therefore, it is expected that $N/2$ atoms in the box will have decayed in this period. ■

Example 4.16 Show that the random variable Z representing the minimum of two independent exponential random variables X and Y with respective parameters λ_1 and λ_2 is an exponential random variable with parameter $(\lambda_1 + \lambda_2)$.

Solution: Since Z is the minimum of independent variables X and Y the following holds:

$$\begin{aligned} P(Z \geq z) &= P((X \geq z) \cap (Y \geq z)) = P(X \geq z)P(Y \geq z) \\ &= \exp(-\lambda_1 z)\exp(-\lambda_2 z) = \exp(-(\lambda_1 + \lambda_2)z) \end{aligned}$$

We have just shown that the cumulative distribution $P(Z \leq z)$ is $1 - \exp(-(\lambda_1 + \lambda_2)z)$, which is precisely the exponential distribution with parameter $(\lambda_1 + \lambda_2)$. ■

Example 4.17 A casino slot machine resets its payoff distribution each time it is played. For each play, it first generates the parameter λ from the exponential distribution $f_\Theta(\lambda) = 2\exp(-2\lambda)$. Then, it uses the realized value of $\Theta = \lambda$ to generate the payoff (reward) R in dollars from another exponential distribution conditional density function $f_{R|\Theta=\lambda}(r) = \lambda\exp(-\lambda r)$. What is the unconditional density function $f_R(r)$ of the payoff? If you are informed that on a particular play, an observed payoff of less than \$2 was observed, what is the posterior density function $f_{\Theta|R<2}(\lambda)$ of Θ for that particular play? Can you use this density function to make the best guess for Θ in that play?

Solution: One can use the total probability rule in continuous hypothesis spaces to generate the unconditional density function:

$$f_R(r) = \int_{\lambda=0}^{\infty} f_\Theta(\lambda) f_{R|\Theta=\lambda}(r) d\lambda = \int_{\lambda=0}^{\infty} 2\lambda\exp(-\lambda[r+2]) d\lambda = \frac{2}{[r+2]^2}$$

The above integration is done by parts while treating r as a constant. The probability $P(R < 2)$ is obtained by integrating the density function w.r.t. r from 0 to 2:

$$P(R < 2) = \int_{r=0}^2 [2/(r+2)^2] dr = [-2/(r+2)]_0^2 = 0.5$$

In order to find the posterior density function given $R < 2$, can use the Bayes rule in continuous hypothesis spaces:

$$f_{\Theta|R<2}(\lambda) = \frac{f_{\Theta}(\lambda)P(R < 2|\Theta = \lambda)}{P(R < 2)} = \frac{2\exp(-2\lambda)F_{R|\Theta=\lambda}(2)}{P(R < 2)} = 4\exp(-2\lambda)(1-\exp(-2\lambda))$$

The best guess for λ can be found by finding the value of λ at which the above density is maximized in $\lambda = (0, \infty)$ (i.e., mode of posterior distribution). On differentiating and setting to 0, we obtain:

$$-8\exp(-2\lambda) + 16\exp(-4\lambda) = 0$$

On solving the above equation, one obtains $\lambda = \ln(2)/2$. It can also be shown that the second derivative is negative for this value of λ , which implies a maximum. ■

Problem 4.12 (Memoryless Property of Exponential Distribution) *A particular outcome (e.g., radioactive decay) of the exponential random variable T is larger than the constant $t_0 > 0$ (e.g., the decay has not occurred till t_0). Show the following for any $t > 0$:*

$$P(T > t + t_0 | T > t_0) = P(T > t)$$

Show that the conditional density function of the remaining time to arrival after a fruitless wait of t_0 is the same as the unconditional density $f_T(t)$ at the very beginning:

$$f_{T|T>t_0}(t + t_0) = f_T(t)$$

A hint for solving the above problem is that $P((T > t + t_0) \cap (T > t_0)) = P(T > t + t_0)$. Once you have solved the first part, express it in terms of cumulative distribution functions and differentiate both sides with respect to t .

Problem 4.13 *A radioactive atom of element A decays into element B requiring a time (from starting to observe the single atom) following the exponential distribution with parameter λ_1 . Furthermore, element B decays to element C requiring a time following the exponential distribution with parameter λ_2 . You start observing a specific atom of element A and record the time it takes to decay all the way to element C. What is the expected value and variance of your observation time?*

Problem 4.14 (The Laplace Distribution) *A useful distribution for probabilistic modeling in machine learning is the Laplace distribution, which corresponds to two exponential distributions glued back to back to a specific point defined by the location parameter μ . In addition, the distribution also has a scale parameter s , which can be viewed in a manner similar to the inverse of the arrival rate of the exponential distribution. Specifically, a random variable Z is said to belong to the Laplace distribution, if its probability density function is defined as follows:*

$$f_Z(z) = \frac{1}{2s} \exp(-\|z - \mu\|_1/s) = \begin{cases} \frac{\exp((z-\mu)/s)}{2s} & \text{if } z < \mu \\ \frac{\exp(-(z-\mu)/s)}{2s} & \text{if } z \geq \mu \end{cases}$$

Show that the mean of the Laplace distribution is μ , variance is $2s^2$, and that the mean absolute deviation is s . You can do this either by using integral calculus or by using its relationship with the exponential distribution.

4.9 The Poisson Distribution

The Poisson distribution defines a discrete random variable that is closely related to the exponential distribution. Recall that an exponential random variable reports the time that it takes for an arrival to occur (given an arrival rate λ), after which the generating process is terminated. The Poisson distribution uses the same generating process, except that it restarts the generating process after each arrival, so that we now have a *sequence* of events occurring continuously in time. Therefore, we now refer to λ as the *inter-arrival rate*. Note that the time between any pair of successive arrivals follows the exponential distribution. Instead of reporting the time between successive arrivals, the Poisson random variable *counts the number of events in a fixed time interval τ* . As a practical example, consider the case where you are a bus-stop waiting for a bus and the arrival time between successive buses is drawn from an exponential distribution. Then, the number of buses arriving in τ time units is drawn from the Poisson distribution.

As in the case of the exponential distribution, one can assume that the Poisson process uses independent Bernoulli trials over infinitesimal time intervals. However, the difference from the exponential distribution is that the number of successes are counted over the fixed time interval τ rather than reporting the time to first success. Let the length of each infinitesimal time interval be δ_t . Now that we have a *fixed* time length τ , the number of Bernoulli trials is a fixed value of $m = \tau/\delta_t$ and therefore one can use the results from the binomial distribution (instead of the geometric distribution) to compute the probability of a specific number of successes in time τ . The probability of success of each Bernoulli trial is $\delta_p = \lambda\delta_t$. Since $\delta_t = \tau/m$, one can also express the success probability as $\delta_p = \lambda\tau/m$. Therefore, we have m independent trials of a Bernoulli process with infinitesimally small success probability $\lambda\tau/m$, where the number of trials $m \rightarrow \infty$ (since the Bernoulli trials are performed over infinitesimally small intervals). The interplay between the small success probability and large number of trials can be shown to lead to a finite probability for the number of successes x using mathematical limits. The probability of x successes is obtained using the probability mass function from the binomial distribution as follows:

$$\begin{aligned}
 p_X(x) &= \lim_{m \rightarrow \infty} \binom{m}{x} \left(\frac{\lambda\tau}{m}\right)^x \left(1 - \frac{\lambda\tau}{m}\right)^{m-x} \\
 &= \left[\lim_{m \rightarrow \infty} \binom{m}{x} \left(\frac{\lambda\tau}{m}\right)^x \right] \underbrace{\left[\lim_{m \rightarrow \infty} \left(1 - \frac{\lambda\tau}{m}\right)^m \right]}_{\exp(-\lambda\tau)} \underbrace{\left[\lim_{m \rightarrow \infty} \left(1 - \frac{\lambda\tau}{m}\right)^{-x} \right]}_1 \\
 &= \left[\lim_{m \rightarrow \infty} \frac{m!}{(m-x)!m^x} \right] \left[\frac{(\lambda\tau)^x}{x!} \right] \exp(-\lambda\tau) \\
 &= \underbrace{\left[\lim_{m \rightarrow \infty} \prod_{j=0}^{x-1} (1 - j/m) \right]}_1 \left[\frac{(\lambda\tau)^x}{x!} \right] \exp(-\lambda\tau)
 \end{aligned}$$

Note that the above derivation uses the simplification that the expression $(1 - a/m)^m$ converges to $\exp(-a)$ as m goes to ∞ . Furthermore, two subexpressions that are claimed to converge to 1 (in the “under-brace” notation above) do so because x is finite and therefore $x \ll m$. One can now create the expression for the PMF of the Poisson random variable as

follows:

$$p_X(x) = \left[\frac{(\lambda\tau)^x}{x!} \right] \exp(-\lambda\tau)$$

An important observation at this point is that we do not really need the parameter τ , if the inter-arrival rate λ is expressed in terms of “arrivals per τ units of time.” In other words, all parameters are re-scaled to reset 1 unit of time to τ units in the original time scale. In such a case, the value of λ is scaled up by τ and the value of τ is set to 1 (so that the all-important product $\lambda\tau$ remains unchanged). The resulting PMF with the scaled inter-arrival rate is expressed without the parameter τ as follows:

$$p_X(x) = \left[\frac{\lambda^x}{x!} \right] \exp(-\lambda)$$

Note that the value of x can be any integer from 0 to ∞ . One can easily confirm that the sum of $p_X(x)$ over all valid values of x is 1 by using the Maclaurin expansion of the exponential function:

$$\sum_{x=0}^{\infty} p_X(x) = \sum_{x=0}^{\infty} \underbrace{\left[\frac{\lambda^x}{x!} \right]}_{\exp(\lambda)} \exp(-\lambda) = 1$$

A random variable X that is drawn from the Poisson distribution is denoted as follows:

$$X \sim \text{Poisson}(\lambda)$$

Examples of the Poisson random variable for different values of λ are illustrated in Figure 4.7. Only the first 16 probability values of the PMF are shown in each case, since the remaining values are close to 0. It is noteworthy that the mode of the Poisson distribution is always an integer that is close to the inter-arrival rate λ . In fact, the mode of the Poisson random variable can always be shown to be $\lfloor \lambda \rfloor$ if λ is not an integer. If λ is an integer, both $\lambda - 1$ and λ are the modes. This result can be shown by considering the ratio $p_X(x+1)/p_X(x)$ at various values of x and checking where this ratio is greater than 1. It can be shown that the largest value of $p_X(x)$ is always achieved at $x = \lfloor \lambda \rfloor$ for non-integers and $x \in \{\lambda - 1, \lambda\}$ for integers. Therefore, all the examples in Figure 4.7 have two modes. Like the binomial distribution for a small number of trials, the Poisson distribution can be skewed at small values of λ — furthermore, the Poisson distribution becomes increasingly symmetrical about the mean at large values of λ as does the binomial distribution for a large number of trials.

Next, we discuss the computation of the expected value and variance of the Poisson random variable. The expression for the expected value of the Poisson random variable is derived as follows:

$$\begin{aligned} \mu_X &= E[X] = \sum_{x=0}^{\infty} x p_X(x) = \sum_{x=0}^{\infty} x \left[\frac{\lambda^x}{x!} \right] \exp(-\lambda) \\ &= \sum_{x=1}^{\infty} \left[\frac{\lambda^x}{(x-1)!} \right] \exp(-\lambda) = \lambda \underbrace{\sum_{x=1}^{\infty} \left[\frac{\lambda^{x-1}}{(x-1)!} \right]}_{\exp(\lambda)} \exp(-\lambda) = \lambda \end{aligned}$$

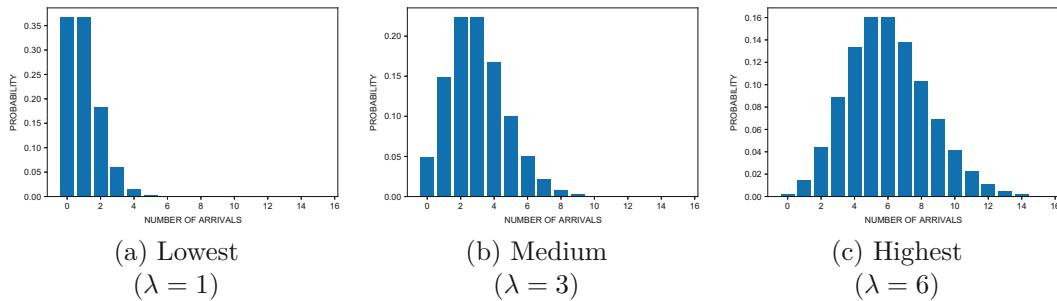


Figure 4.7: PMFs of different Poisson distributions with varying inter-arrival rate λ . The probabilities of only the first 16 outcomes of the Poisson distribution are shown in each case.

The above derivation uses the Maclaurin expansion of the exponential function $\exp(\lambda)$. In order to compute the variance, we first compute the second moment of the random variable:

$$E[X^2] = \sum_{x=0}^{\infty} x^2 p_X(x) = \underbrace{\sum_{x=1}^{\infty} x(x-1)p_X(x)}_{\lambda^2} + \underbrace{\sum_{x=0}^{\infty} x p_X(x)}_{\lambda} = \lambda^2 + \lambda$$

The above derivation also uses the Maclaurin expansion of $\exp(\lambda)$. The variance of the Poisson variable may then be computed as follows:

$$\sigma_X^2 = E[X^2] - (E[X])^2 = \lambda^2 + \lambda - \lambda^2 = \lambda$$

Example 4.18 Consider a Poisson random variable with arrival rate 2. Calculate the median of the random variable.

Solution: The approach will be to successively compute $P(X = 0)$, $P(X = 1)$, $P(X = 2)$, and so on, until a cumulative probability of at least 0.5 is achieved for the outcomes $\{1, \dots, r\}$. The first integer value r at which the cumulative probability of 0.5 is exceeded is the median:

$$\begin{aligned} P(X = 0) &= \frac{2^0}{0!} \exp(-2) = 1/e^2 \quad F_X(0) = 1/e^2 < 0.5 \\ P(X = 1) &= \frac{2^1}{1!} \exp(-2) = 2/e^2 \quad F_X(1) = (1+2)/e^2 < 0.5 \\ P(X = 2) &= \frac{2^2}{2!} \exp(-2) = 2/e^2 \quad F_X(2) = (1+2+2)/e^2 > 0.5 \end{aligned}$$

Since the cumulative probability first exceeds 0.5 at $X = 2$, the median is 2. ■

Example 4.19 (Sum of Poisson Random Variables is Poisson) Let X and Y be independent Poisson variables with inter-arrival rates λ_1 and λ_2 , respectively. Show that $(X + Y)$ is a Poisson random variable with inter-arrival rate $(\lambda_1 + \lambda_2)$.

Solution: We use an intuitive argument based on generating processes, although the result has been formally shown in Example 3.30 using the convolution operator. A Poisson random process with inter-arrival rate λ is defined by independent arrivals occurring with probability $\lambda\delta_t$ in a time period δ_t when $\delta_t \rightarrow 0^+$. When we have two processes with arrival rates λ_1 and λ_2 occurring, the probability of both types of arrivals occurring in an infinitesimal period δ_t is proportional to δt^2 , which is negligible compared to the probability of one arrival occurring. The probability of one arrival in infinitesimal time period δ_t is $(\lambda_1\delta_t(1-\lambda_2\delta_t) + \lambda_2\delta_t(1-\lambda_1\delta_t))$. Ignoring higher-order terms, this probability is $(\lambda_1 + \lambda_2)\delta_t$. This is exactly the generating process of a Poisson random variable with inter-arrival rate $(\lambda_1 + \lambda_2)$. ■

Example 4.20 Mad-Hatter runs a 24-hour bus company in which the number of arrivals X in each set of 24 hours (midnight-to-midnight) follows a Poisson distribution with arrival rate λ . Just before each midnight, he decides the value of λ by sampling from the exponential distribution $f_\Theta(\lambda) = \exp(-\lambda)$. What is the probability of no arrivals in a 24-hour period (i.e., $X = 0$)? In a particular set of 24 hours, no arrivals were observed. Find the posterior distribution $f_{\Theta|X=0}(\lambda)$.

Solution: Using the total probability rule in continuous hypothesis spaces, one can evaluate $P(X = 0)$:

$$p_X(0) = \int_{\lambda=0}^{\infty} f_\Theta(\lambda)p_{X|\Theta=\lambda}(0)d\lambda = \int_{\lambda=0}^{\infty} \exp(-\lambda)\exp(-\lambda)d\lambda = \frac{1}{2}$$

Using the Bayes rule in continuous hypothesis spaces, one can find the posterior distribution:

$$f_{\Theta|X=0}(\lambda) = \frac{f_\Theta(\lambda)p_{X|\Theta=\lambda}(0)}{p_X(0)} = 2\exp(-2\lambda)$$

Note that the posterior distribution is also exponential, but its expectation is half of that of the original distribution $f_\Theta(\lambda)$. The reason is that no arrivals were observed, which biases the posterior expectation to lower values. ■

Problem 4.15 Let X and Y be independent Poisson random variables with inter-arrival rates 3 and 4, respectively. Find the mean and variance of $X + Y$.

Problem 4.16 What is the mean and the mode of a Poisson distribution with $\lambda = 0.1$? Is this Poisson distribution left skewed or right skewed? Do you expect the mean or the median of this Poisson distribution to be greater?

4.10 The Normal Distribution

The normal distribution is also referred to as the *Gaussian distribution*, and it is the single most widely used distribution in statistics. The reason for its popularity is that its generating process can be viewed as the summing (or averaging) of many independent random variables, and this phenomenon is ubiquitous in real-world settings. In order to understand the effect of summing random variables, we will start with an exponential random variable with parameter $\lambda = 1$ and visualize the sum of multiple independent instantiations of

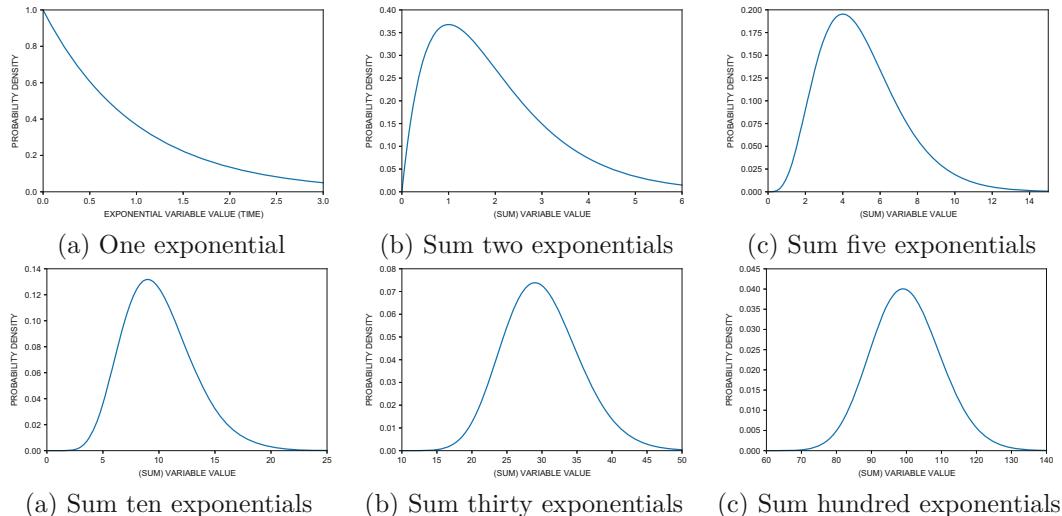


Figure 4.8: PDFs obtained by adding multiple exponential variables. As more exponentials are added, the bell-shaped normal distribution appears.

this variable. The exponential distribution is selected as an example, because its one-tailed probability distribution looks extremely different from the symmetric normal distribution.

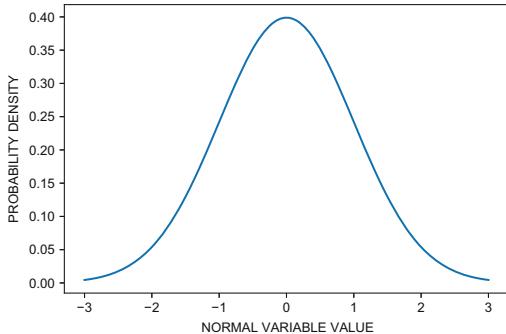
Figure 4.8(a) shows the probability distribution of a single exponential random variable \bar{X}_1 , which is clearly a skewed and highly asymmetric probability density function. Figures 4.8(b), (c), (d), (e), and (f) show the probability distributions of the random variable $Y_n = \sum_{i=1}^n X_i$ for varying values of n . It is evident that as n increases, the sum becomes increasingly symmetric and starts resembling a bell-shaped curve. This bell-shaped curve can be shown to converge to the normal distribution with increasing n . This fundamental result is referred to as the *central limit theorem* (cf. section 5.2 of Chapter 5). The aforementioned behavior can be shown to hold *irrespective of which base distribution is chosen for aggregation*. Even when different distributions are aggregated, a normal distribution is often obtained under some modest assumptions.

The central limit theorem is significant because the observed data in real-world settings are often generated by the (implicit) averaging of many “hidden” factors (i.e., factors not directly visible to us). For example, the Intelligence Quotient (IQ) of an individual is the result of the averaging of many randomly distributed factors associated with genes, nutrition, social interaction, and the environment. As a result, the IQs of the individuals in various homogeneous sub-populations are often normally distributed. Other normal distributions include height and birth-weight, which are also defined by the averaging effects of multiple hidden factors. This averaging effect is the reason for the ubiquity of the normal distribution in the real world. The density function of the normal distribution is expressed in terms of its mean μ and variance σ^2 :

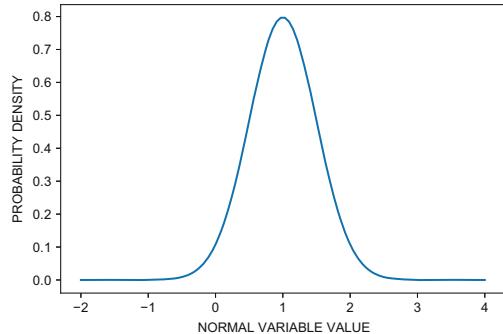
$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (4.1)$$

When a random variable X is sampled from the normal distribution with parameters μ and σ^2 , it is denoted by the following:

$$X \sim \mathcal{N}(\mu, \sigma^2)$$



(a) Mean of 0 and variance of 1



(b) Mean of 1 and variance of 0.25

Figure 4.9: Revisiting Figure 1.4: The PDFs of two normal distributions

Because of the convenient parametrization of the normal distribution, no additional computations are necessary to evaluate its mean and variance:

$$\mu_X = \mu, \quad \sigma_X^2 = \sigma^2$$

Example 4.21 Show that the mean absolute deviation of the normal random variable $X \sim \mathcal{N}(\mu, \sigma^2)$ is $\sqrt{2/\pi}\sigma$.

Solution: The mean absolute deviation MAD_X is defined as follows:

$$\begin{aligned} MAD_X &= \frac{1}{\sqrt{2\pi}\sigma} \int_{x=-\infty}^{\infty} |x - \mu| \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) dx \\ &= \frac{2}{\sqrt{2\pi}\sigma} \int_{x=\mu}^{\infty} (x - \mu) \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) dx \end{aligned}$$

The expression on the right is obtained because of symmetry about μ . On setting $w = (x - \mu)^2/(2\sigma^2)$, one can substitute $(x - \mu)dx = \sigma^2 dw$ to obtain the following:

$$MAD_X = \frac{2\sigma}{\sqrt{2\pi}} \underbrace{\int_{w=0}^{\infty} \exp(-w) dw}_1 = \sqrt{\frac{2}{\pi}}\sigma$$

Note that the mean absolute deviation is less than the standard deviation. ■

Two examples from the family of normal distributions are illustrated in Figure 4.9. The basic shape of the distribution is similar in both cases with the main difference being the placement of the central point and the level of dispersion. The distribution in Figure 4.9(a) with mean $\mu = 0$ and standard deviation $\sigma = 1$ is an important special case of the normal distribution family, which is referred to as the *standard normal distribution*. One can convert an arbitrary normal distribution to a standard normal distribution by using a process called *standardization*. The random variable $X \sim \mathcal{N}(\mu, \sigma^2)$ can be standardized to $Z \sim \mathcal{N}(0, 1)$ by using the following variable transformation:

$$Z = \frac{X - \mu}{\sigma}$$

It is common to standardize attributes in machine learning applications by using their sample means and standard deviations (cf. section 2.4.1 of Chapter 2). This type of preprocessing ensures the relative comparability (in magnitude) of the values of different attributes that are drawn on vastly different scales (e.g., birth-weight and height). The standardized height of the 95th percentile of the population will be the same as the standardized weight of the 95th percentile for normally distributed data — this unit-less value turns out to be about 1.65. Furthermore, various computations such as multidimensional distance computations in Euclidean space become probabilistically interpretable with easily defined distributions.

Because of the ability to standardize normally distributed points and map them to percentiles, the cumulative distribution of the standard normal variable is very useful. However, this cumulative distribution is not available as a closed-form expression but as a table (since the normal density function does not have a closed-form integral). The cumulative normal distribution is shown in Figure 4.10. Some percentile values of the standard normal distribution are so common (and important) that it is helpful to simply commit them to memory. These standardized values correspond to the 95th, 97.5th, and 99.5th percentiles together with their symmetric counterparts corresponding to the 5th, 2.5th and 0.5th percentiles. These important values are illustrated in Table 4.1. It is evident from the table that most of the data is concentrated in the center of the distribution within three standard deviations of the mean. Furthermore, only 2.5% of the standard normal distribution has a value less than -1.96 and another 2.5% of the data has a value greater than 1.96 . In other words, 95% of the standard normal distribution corresponds to values that are less than 1.96 in absolute magnitude:

$$P(|Z| \leq 1.96) = 0.95$$

One can also write this condition in terms of the general normal variable $X \sim \mathcal{N}(\mu, \sigma^2)$:

$$\begin{aligned} P(|X - \mu|/\sigma \leq 1.96) &= 0.95 \\ P(\mu - 1.96\sigma \leq X \leq \mu + 1.96\sigma) &= 0.95 \end{aligned}$$

One can make similar assertions about other important percentile values and two-tailed cut-offs by using the values of Table 4.1:

$$\begin{aligned} P(\mu - 1.65\sigma \leq X \leq \mu + 1.65\sigma) &= 0.9 \\ P(\mu - 2.58\sigma \leq X \leq \mu + 2.58\sigma) &= 0.99 \\ P(\mu - 3\sigma \leq X \leq \mu + 3\sigma) &= 0.997 \end{aligned}$$

In cases where the population mean is not known, these types of bounds are also very useful for setting up ranges in which the population mean is likely to lie with high probability by using means of data samples as the central points of these intervals. Such ranges are referred to as *confidence intervals*, which are discussed in detail in Chapter 5. Here, we provide a brief example of the type of application that is often enabled with this approach:

Example 4.22 A factory produces widgets that are meant to be 100 cm long. However, because of small variations in the manufacturing process, it produces widgets which are normally distributed with a mean of 100 cm and a standard deviation of 0.15 cm. In what range of lengths do the 95% of the widgets closest to the central length of 100 cm lie? The factory will have to discard widgets with lengths greater than 100.4 centimeters (although abnormally small widgets are considered acceptable). What percentage of the widgets will the factory have to discard?



**Probability Content
from $-\infty$ to Z**

Z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5873	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.4	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
2.6	0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
2.7	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
2.8	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9979	0.9980	0.9981
2.9	0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
3.0	0.9987	0.9987	0.9987	0.9988	0.9988	0.9989	0.9989	0.9989	0.9990	0.9990



**Far Right
Tail Probabilities**

Z	$P\{Z \text{ to } \infty\}$						
2.0	0.02275	3.0	0.001350	4.0	0.00003167	5.0	2.867 E-7
2.1	0.01786	3.1	0.0009676	4.1	0.00002066	5.5	1.899 E-8
2.2	0.01390	3.2	0.0006871	4.2	0.00001335	6.0	9.866 E-10
2.3	0.01072	3.3	0.0004834	4.3	0.00000854	6.5	4.016 E-11
2.4	0.00820	3.4	0.0003369	4.4	0.000005413	7.0	1.280 E-12
2.5	0.00621	3.5	0.0002326	4.5	0.000003398	7.5	3.191 E-14
2.6	0.004661	3.6	0.0001591	4.6	0.000002112	8.0	6.221 E-16
2.7	0.003467	3.7	0.0001078	4.7	0.000001300	8.5	9.480 E-18
2.8	0.002555	3.8	0.00007235	4.8	7.933 E-7	9.0	1.129 E-19
2.9	0.001866	3.9	0.00004810	4.9	4.792 E-7	9.5	1.049 E-21

Figure 4.10: A public-domain normal distribution table. The table was produced by William Knight using APL programs. The notation “ ∞ ” in the table should be read as ∞ . The entries in this table for any Z -value z can also be generated using the Excel function `normdist(z,μ,σ,1)` where μ and σ are set to 0 and 1, respectively.

Table 4.1: The values of the standard normal variable for various percentile points

Standard normal variable value	Percentile
-3	0.13
-2.58	0.5
-1.96	2.5
-1.65	5
-1	15.87
0	50
1	84.13
1.65	95
1.96	97.5
2.58	99.5
3	99.87

Solution: Since 95% of the widgets lie within ± 1.96 standard deviations of the mean, it follows that 95% of the manufactured widgets lie within $1.96 * 0.15$ centimeters of the mean of 100 centimeters, which provides a range of [99.706, 100.294]. In order to compute the percentage of widgets that will have to be discarded, it makes sense to transform a widget with length 100.4 to the standardized scale, where it is easier to judge its relative position on the bell curve of the standard normal distribution:

$$z = \frac{100.4 - \mu}{\sigma} = \frac{100.4 - 100}{0.15} = \frac{8}{3}$$

One can use the cumulative distribution tables of the normal distribution to verify that $P(Z > 2.667) = 0.38\%$. In other words, about 0.38% of the widgets will have to be discarded. ■

When a data point modeled by the normal distribution is transformed to the standardized scale, it is referred to as the *Z-value* of the point.

Definition 4.1 (Z-value) *The Z-value is the number of population standard deviations σ by which an observation is distant from the population mean μ :*

$$z = \frac{x - \mu}{\sigma}$$

The Z-value is often denoted using the variable z , which is consistent with the name of the term. Normal distribution tables (or appropriate statistical software) are often used to map a Z-value z to a percentile value. Therefore Z-values are very useful from a practical perspective. We provide an example of such an application:

Example 4.23 *The scores in a standardized test are (roughly) normally distributed with a mean of 530 and a standard deviation of 90. What is the Z-value of a student scoring 710 on the examination? What percentage of the students appearing on the examination scored higher than this student? What is the cut-off for the 95th percentile?*

Solution: The Z-value of the student can be computed as follows:

$$z = \frac{710 - \mu}{\sigma} = \frac{710 - 530}{90} = 2$$

One can use the normal distribution table to show that $P(Z \leq 2) = 0.97725$. The percentage of students scoring *higher* than this student is therefore $100(1 - 0.97725) = 2.275$.

In order to convert a percentile score to a raw score, the reverse mapping from percentile value to a Z-score is used. First note that the percentile value of 95% maps to a Z-value of 1.65. Therefore, the cut-off score is 1.65 standard deviations greater than the mean, which is $530 + 1.65 * 90 = 678.5$. ■

The aforementioned examples assume that the *population* mean μ and standard deviation σ are known. This is often not possible in the real world, where one is forced to use the *sample* mean and standard deviation to compute Z-scores (which, in this case, might correspond to the sample mean and standard deviation of the particular students taking the examination). In other words, instead of translating and scaling a normal random variable with population parameters, we are translating and scaling it with two functions (i.e., mean and standard deviation) of multiple independent and identically distributed (i.i.d.) random variables, which may vary from sample to sample. If the sample size is large enough (say, 30 or more), the resulting Z-scores will still follow the standard normal distribution closely enough to use this approach meaningfully. For example, the preprocessing (whitening) techniques discussed in section 2.4.1 convert attributes to Z-values using sample means and standard deviations.

Problem 4.17 *Fishermen in Boston Harbor catch a particular type of rare fish with mean length of 8 inches and standard deviation of 1.5 inches. Mr. Unhygienix catches a fish that is 11 inches long and boasts that it was the largest fish of that type ever caught in Boston Harbor. Assuming that harvested fish lengths are i.i.d. normally distributed values, and also assuming that about ten-thousand fish of that rare type were caught in Boston Harbor before Unhygienix's catch, do you think that the fisherman's claim is likely to be correct?*

A hint for solving this problem is to compute the probability that the maximum of the lengths of 10,000 other fish that were caught earlier is greater than 11 inches.

Problem 4.18 *Champion athletes in a national camp training for the 100 meter dash historically have a mean clock-time of 9.7 seconds and standard deviation of 0.04 seconds. Assume that their running times are normally distributed. Compute the Z-value of an athlete, Mr. Lightening Bolt, who runs the 100 meter dash in 9.78 seconds. What is the probability that a randomly chosen athlete from this camp runs faster than Mr. Bolt? Mr. Bolt estimates that he needs to be among the top-0.5% of athletes in the camp in order to qualify for the Olympics. What is the running time that he should target?*

Problem 4.19 *The lengths of widgets produced by a factory are drawn from a normal distribution with mean 100 centimeters and standard deviation of 1 centimeter. Widget lengths on the production line are independent of one another and identically distributed. Widgets are divided into batches of 100 widgets, and each batch is given a percentile rank based on its average length. What is the percentile rank of a batch for which the average length is 100.1 centimeters? [Be careful — we are not talking about an individual widget here. You need to consider the variance of average batch length.]*

4.10.0.1 Closure Properties of the Normal Distribution Family

A nice property of the normal distribution is that it is closed under all sorts of linear transformations. In other words, performing linear operations on one or more normal random

variables will also result in a normal random variable. In particular, the following can be shown:

1. If $X \sim \mathcal{N}(\mu, \sigma^2)$, then $Y = aX + b$ can be shown to be drawn the normal distribution $\mathcal{N}(a\mu + b, a^2\sigma^2)$. Here, a and b are constant coefficients.
2. Let $X_1 \dots X_k$ be (possibly dependent) normal random variables, so that $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ and σ_{ij} be the corresponding covariances. Let Y be an affine function of these random variables using coefficients $[a_1, a_2, \dots, a_k]$ as follows:

$$Y = b + \sum_{i=1}^k a_i X_i$$

Then, the random variable Y is drawn from the following normal distribution:

$$Y \sim \mathcal{N}\left(b + \sum_{i=1}^k a_i \mu_i, \sum_{i=1}^k a_i^2 \sigma_i^2 + 2 \sum_{i=1}^k \sum_{j=i+1}^k a_i a_j \sigma_{ij}\right)$$

A formal proof of these results is omitted, although a special case is explored in Exercise 21 of Chapter 3 and the general case is explored in Exercise 35 of this chapter.

The above results show that making any type of linear (or affine) transforms of one or more normal random variables is guaranteed to yield another normal random variable! In fact, it can even be shown that adding a sufficient number of (not necessarily identical) independent random variables also results in a normal distribution (as long as their standard deviations are sufficiently similar). This result is referred to as the *generalized central limit theorem*. This is the reason that normal distributions are so ubiquitous in the real world — the underlying data generating processes often (implicitly) aggregate the contributions of several independent factors. Once a normal distribution has been reached by linear aggregation, the closure property ensures that further linear transformations will not disturb this property. In other words, the normal distribution is the final and stable stop in the pipeline of linear transformations.

Example 4.24 Let $X \sim \mathcal{N}(3, 2^2)$ and $Y \sim \mathcal{N}(5, 1^2)$ be normally distributed random variables with covariance of 2. Find the distribution of $Z = 2X + 3Y + 5$.

Solution: We start by computing the mean and variance of Z using the affine results from the previous chapter:

$$\begin{aligned}\mu_Z &= 2\mu_X + 3\mu_Y + 5 = 2 * 3 + 3 * 5 + 5 = 26 \\ \sigma_Z^2 &= 4\sigma_X^2 + 9\sigma_Y^2 + 2 * 2 * 3 * \sigma_{XY} = 16 + 9 + 24 = 49\end{aligned}$$

Therefore, the variable Z is normally distributed with a mean of 26 and variance of 49. In other words, we have $Z \sim \mathcal{N}(26, 7^2)$. ■

4.10.1 Multivariate Normal Distribution: Independent Attributes

In the case of independent attributes, it is assumed that we have a random vector $\vec{X} = [X_1, \dots, X_d]$, where the i th variable X_i is independent of other components of the vector and

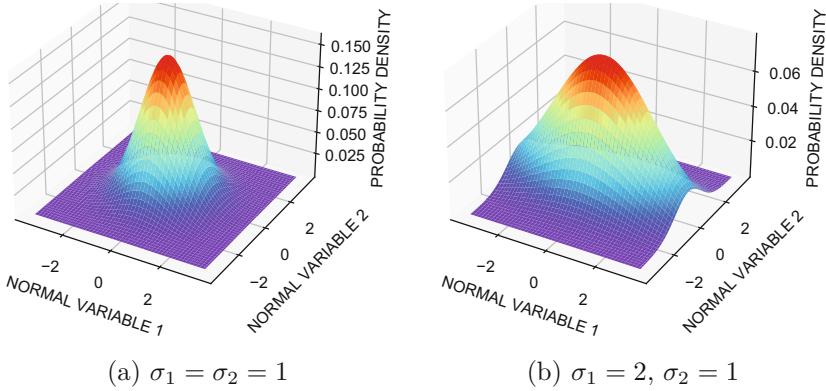


Figure 4.11: PDFs for two multivariate normal distributions with zero mean and independent attributes. The two distributions differ in variance.

is normally distributed with mean μ_i and standard deviation σ_i :

$$X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$$

Therefore, the 1-dimensional marginal density function for X_i is as follows:

$$f_{X_i}(x_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right)$$

Since the different attributes are independent of one another, the joint probability density function is obtained as a product of marginal density functions. In other words, the joint distribution of the vector \vec{X} is as follows:

$$f_{\vec{X}}(x_1, \dots, x_d) = \prod_{i=1}^d f_{X_i}(x_i) = \frac{1}{(2\pi)^{d/2} \prod_{i=1}^d \sigma_i} \exp\left(-\sum_{i=1}^d \frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right)$$

One can also write the above function in vector and matrix notation by defining Δ to be a $d \times d$ diagonal matrix containing the variances on its diagonal entries (in the same order as the variables in \vec{X}) and also defining $\vec{\mu}$ to be a d -dimensional row vector containing the means of different attributes. Let $\vec{x} = [x_1 \dots x_d]$ be a row vector containing an instantiation of the normal variable. Let $|\Delta|$ denote the determinant of Δ , which also happens to be the product of the diagonal entries (variances) in this special case. Then, the probability density function may be written as follows:

$$f_{\vec{X}}(\vec{x}) = \frac{1}{(2\pi)^{d/2} |\Delta|^{1/2}} \exp\left(-\frac{[\vec{x} - \vec{\mu}] \Delta^{-1} [\vec{x} - \vec{\mu}]^T}{2}\right) \quad (4.2)$$

This form is helpful because it is closely related to the general form of how the Gaussian distribution is written. This form will be explored in the next section.

The multivariate Gaussian distribution shows up on scatter plots in the form of centrally concentrated clusters. When the standard deviations along different dimensions are the same, the clusters are spherical. On the other hand, when the standard deviations along different dimensions are different, the clusters are elongated to an elliptical shape. When the attributes are independent, it is assumed that the axes of these ellipses are aligned with the axis system. In order to understand this point, we show examples of two joint density

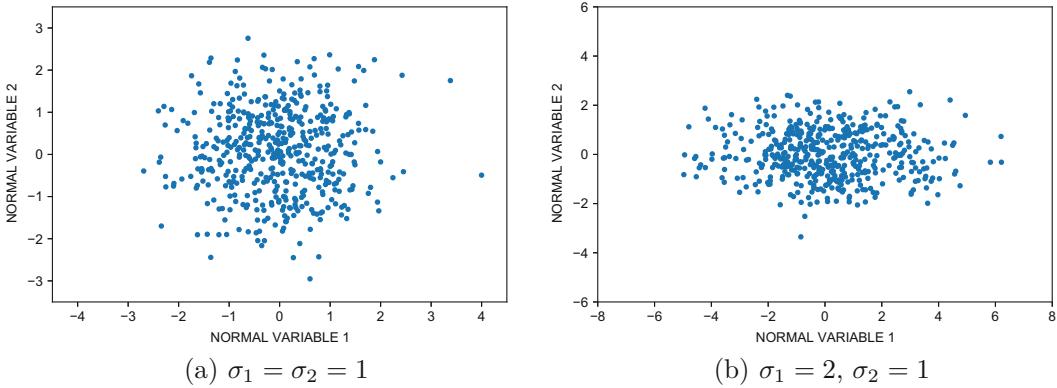


Figure 4.12: The scatter plots for the multivariate normal distributions of Figure 4.11.

functions in Figure 4.11. The density function in Figure 4.11(a) is that of a 2-dimensional normal distribution with a mean of 0 and a standard deviation of 1 along both directions. On the other hand, the joint density function of a 2-dimensional normal distribution with standard deviations of 1 and 2 along the two directions is shown in Figure 4.11(b). It is evident that the joint distribution of Figure 4.11(b) looks elongated along one of the variables as compared to the joint distribution of Figure 4.11(a). Even more insight may be obtained by examining the scatter plots of data sampled from these distributions. The scatter plots are shown in Figures 4.12(a) and (b) in the two respective cases. It is evident that the cluster in Figure 4.12(a) is spherical, whereas that in Figure 4.12(b) is elongated.

4.10.2 Multivariate Normal Distribution: Dependent Attributes

Section 2.4.2 discusses how certain preprocessing methods decorrelate data attributes using *whitening methods* that work with the eigenvectors of the *sample covariance matrix* in order to transform the data along a new orthogonal axis system. The discussion in this section provides a probabilistic view of this process at the population level. The assumption in the previous section that the different attributes of the multivariate Gaussian distribution are independent is unnecessarily restrictive. The most general case of the multivariate Gaussian sheds this assumption and allows dependence among attributes with additional parameters defined by a $d \times d$ *population covariance matrix* for a d -dimensional data distribution. The (i, j) th entry of the symmetric covariance matrix contains the population covariance between dimensions i and j . The diagonal entries of the covariance matrix contain the population variances. Therefore, the number of parameters required for this case is significantly greater than in the case of independent attributes where only the means and variances need to be specified. For simplicity, we first assume that the Gaussian distribution is centered at the origin, and we will later discuss the minor change needed for it to have an arbitrary mean.

Even though the general form of the multivariate Gaussian distribution allows dependence among attributes, *there are always d orthogonal directions along which the attributes are independent of one another*. These independent directions are defined by the eigenvectors of the covariance matrix. The positive semi-definite matrix C can be decomposed into eigenvector and eigenvalue matrices as follows:

$$C = P\Delta P^T$$

Here, P is a $d \times d$ matrix whose columns contain the orthonormal eigenvectors and Δ is a $d \times d$ diagonal matrix containing the eigenvalues (in the same order as the eigenvector columns). The main assumption of the multivariate normal distribution is that the different data coordinates are not generated independently of one another along the original axis directions. Rather, the *data coordinates are generated independently along an orthonormal basis system corresponding to the eigenvector directions of the covariance matrix and the variances of these coordinates are defined by the corresponding eigenvalues*. Using such an approach ensures the existence of pairwise covariances among attributes based on values in C . Let the d -dimensional row-vector $\vec{y} = [y_1, \dots, y_d]$ denote the coordinates of a Gaussian sample along the basis system of eigenvectors (columns of P). This representation can be transformed back to coordinates in the original axis system as the row vector $\vec{x} = [x_1, \dots, x_d]$ using the following transformation:

$$\vec{x}^T = P\vec{y}^T$$

One can also equivalently write $\vec{y} = \vec{x}P$ by using orthogonality of P . Since the marginal probability density functions along the d eigenvector directions are independent (and the data is mean centered), one can directly use Equation 4.2 to represent the joint probability density function by substituting \vec{y} instead of \vec{x} and setting means to $\vec{0}$:

$$f_{\vec{X}}(\vec{x}) = \frac{1}{(2\pi)^{d/2}|\Delta|^{1/2}} \exp\left(-\frac{[\vec{y} - \vec{0}] \Delta^{-1} [\vec{y} - \vec{0}]^T}{2}\right) \quad \text{where } \vec{y} = \vec{x}P$$

On substituting $\vec{y} = \vec{x}P$ in the above equation and recognizing the determinant property¹ that $|C| = |P\Delta P^T| = |\Delta|$, one obtains the following:

$$\begin{aligned} f_{\vec{X}}(\vec{x}) &= \frac{1}{(2\pi)^{d/2}|C|^{1/2}} \exp\left(-\frac{[\vec{x}P] \Delta^{-1} [P^T \vec{x}^T]}{2}\right) \\ &= \frac{1}{(2\pi)^{d/2}|C|^{1/2}} \exp\left(-\frac{[\vec{x}] [P \Delta^{-1} P^T] [\vec{x}^T]}{2}\right) \\ &= \frac{1}{(2\pi)^{d/2}|C|^{1/2}} \exp\left(-\frac{[\vec{x}] C^{-1} [\vec{x}^T]}{2}\right) \quad [\text{Substituting } C^{-1} = P \Delta^{-1} P^T] \end{aligned}$$

This density function assumes that the Gaussian distribution has zero mean. In the case that the probability distribution has an arbitrary mean vector $\vec{\mu}$, the same mean-centered density function as above can be used after subtracting the mean vector from \vec{x} as follows:

$$f_{\vec{X}}(\vec{x}) = \frac{1}{(2\pi)^{d/2}|C|^{1/2}} \exp\left(-\frac{[\vec{x} - \vec{\mu}] C^{-1} [\vec{x} - \vec{\mu}]^T}{2}\right) \quad (4.3)$$

This is the general form of the multivariate Gaussian probability density function, which can be written fully in terms of the means and the covariance matrix. Note that this density function is a natural generalization of Equation 4.2, which has a diagonal covariance matrix Δ because the different variables are independent. Therefore, setting $C = \Delta$ in the above density function yields Equation 4.2. An illustration of the general form of the Gaussian in two dimensions is shown in Figure 4.13. Gaussian data are generally distributed in an ellipsoidal cloud, although the density of the data reduces as it moves further out from the center of the cloud. The axes of the ellipsoid are oriented along the orthogonal eigenvector directions of C .

¹The determinant of matrix products is the product of determinants. Orthogonal matrices have a determinant of 1.

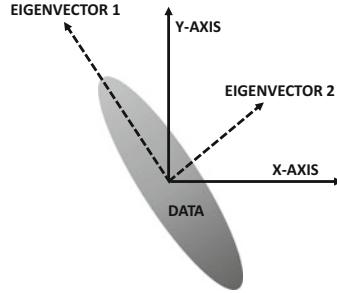


Figure 4.13: An illustration of a 2-dimensional general Gaussian distribution

In many cases, the exponent of the normal distribution may not be specified using a covariance matrix but as the quadratic polynomial $g(\vec{x})$. For example, a 3-dimensional quadratic polynomial of $\vec{x} = [x_1, x_2, x_3]$ could be of the form $g(\vec{x}) = 7x_1^2 + 3x_1x_2 + 5x_2^2 + x_3^2$ and the density function $f_{\vec{X}}(\vec{x})$ is as follows:

$$f_{\vec{X}}(\vec{x}) \propto \exp(-g(\vec{x}))$$

The raw quadratic polynomial seems to provide little insight about the directions of correlation unless it is expressed in vector-matrix form using the inverse covariance matrix. It is the covariance matrix that provides the eigenvector directions. How does one derive the covariance matrix from the polynomial? It turns out that the inverse covariance matrix $C^{-1} = [a_{ij}]$ is the *Hessian* of the polynomial:

$$a_{ij} = \frac{\partial^2 g(\vec{x})}{\partial x_i \partial x_j} \quad \forall i, j \in \{1, \dots, d\}$$

The function $g(\vec{x})$ must satisfy the property that it is positive for all \vec{x} , which occurs only when the eigenvalues of the C^{-1} (or C) are positive. In other words, an arbitrarily exponentiated quadratic function $g(\vec{x})$ cannot be used to create a Gaussian distribution.

One can denote the generation of a vector sample from the multivariate normal distribution with mean $\vec{\mu}$ and covariance matrix C using the following notation:

$$\vec{X} \sim \mathcal{N}(\vec{\mu}, C)$$

The main disadvantage of the general form of the multivariate distribution is the large number parameters in the covariance matrix, which can lead to problems when estimating them from a limited amount of data. Examples of the density function and scatter plot of the multivariate Gaussian distribution with eigenvector directions $[-1, 1]$ and $[1, 1]$ (and corresponding variances/eigenvalues equal to 4 and 1) are illustrated in Figure 4.14. On careful examination, one can also see that the data of Figure 4.14 is very similar to the data of Figures 4.11(b) and Figure 4.12(b), except that the elongated cluster is rotated by 45° in the clockwise direction. This rotation is the result of using rotated axis directions (eigenvector directions) for independently generating coordinate values along those directions.

The discussion in this section provides a probabilistic view of the whitening approach discussed in section 2.4.2. The implicit assumption in whitening is that the data are generated originally from the general multivariate Gaussian. By using whitening, the data is rotated and re-normalized to a spherical Gaussian with no correlations among attributes. It is also instructive to examine how one can use decorrelated directions of a Gaussian

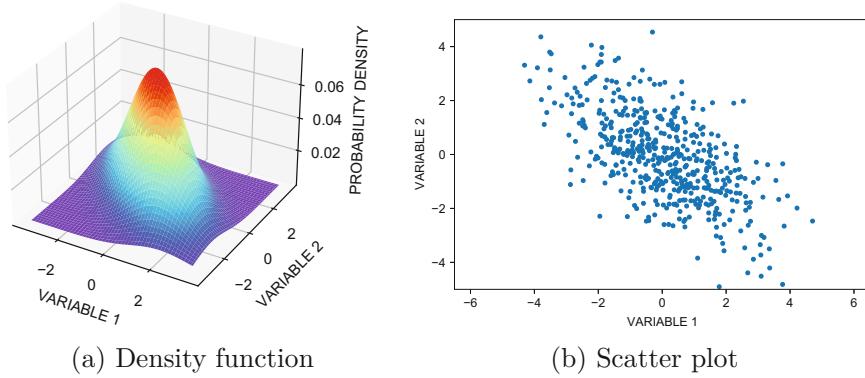


Figure 4.14: A multivariate normal distribution with negatively correlated attributes

(with their variances) to derive the density functions and vice versa. We will use example of Figure 4.14 to show this computation:

Example 4.25 Based on the eigenvector directions $[-1, 1]$ and $[1, 1]$ (and corresponding variances/eigenvalues equal to 4 and 1) for the Gaussian scatterplot of Figure 4.14(b), derive the bivariate probability density function of this Gaussian distribution. The distribution is centered at the origin.

Solution: The eigenvector matrix P and variance matrix Δ is as follows:

$$P = \begin{bmatrix} -1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}, \quad \Delta = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$$

The covariance matrix and its inverse can be calculated using the eigenvector directions as follows:

$$\begin{aligned} C &= P\Delta P^T = \begin{bmatrix} -1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}^T \\ &= \frac{1}{2} \begin{bmatrix} 5 & -3 \\ -3 & 5 \end{bmatrix} \quad C^{-1} = \frac{1}{8} \begin{bmatrix} 5 & 3 \\ 3 & 5 \end{bmatrix} \end{aligned}$$

The negative covariances among the attributes are noteworthy, since the off-diagonal entries of C are negative. Correspondingly, the probability density function in Figure 4.14(a) shows a contour in which the higher density values are aligned along the vector $[-1, 1]$. This is the eigenvector with the larger eigenvalue (variance) and it also provides the direction of elongation.

The determinant of the covariance matrix is 4. On substituting the inverse covariance matrix into the equation of the Gaussian density function with $\vec{\mu} = \vec{0}$, the following is obtained:

$$f_{\vec{X}}(x_1, x_2) = \frac{1}{4\pi} \exp \left(-\frac{5x_1^2 + 6x_1x_2 + 5x_2^2}{16} \right) \quad (4.4)$$

Note the interaction term x_1x_2 in the exponent, which causes the correlation among the attributes. ■

Example 4.26 Suppose you are only provided the closed form of the 2-dimensional density function according to Equation 4.4. You are told that the density function is consistent with a Gaussian distribution. Derive the mean and covariance parameters of the Gaussian distribution. Derive the independent directions and their variances.

Solution: The Gaussian density function is maximized at the mean. By setting the partial derivatives of the density function with respect to x_1 and x_2 to 0, one obtains the constraints $5x_1 - 3x_2 = 0$ and $3x_1 + 5x_2 = 0$. Solving these equations, it follows that the mean vector of this Gaussian is $[0, 0]$.

For the second-order polynomial $g(\vec{x}) = (5x_1^2 + 6x_1x_2 + 5x_2^2)/16$ in the exponent of the Gaussian, we compute the (i, j) th second derivative, which yields the entries of the inverse covariance matrix $C^{-1} = [a_{ij}]$:

$$a_{ij} = \frac{\partial^2 g(\vec{x})}{\partial x_i \partial x_j} \quad i, j \in \{1, 2\}$$

On performing this derivative computation and subsequent matrix inversion, the inverse covariance matrix and covariance matrix can be shown to be the same as that in Example 4.25:

$$C^{-1} = \frac{1}{8} \begin{bmatrix} 5 & 3 \\ 3 & 5 \end{bmatrix} \quad C = \frac{1}{2} \begin{bmatrix} 5 & -3 \\ -3 & 5 \end{bmatrix}$$

The directions of zero correlation are the eigenvectors of this covariance matrix, which are $[1, -1]$ and $[1, 1]$. The corresponding variances (eigenvalues) are 4 and 1, respectively. ■

Problem 4.20 Repeat the problem of Example 4.25 with the modification that the distribution is centered at $(1, 2)$ rather than at the origin. Derive the density function of this modified Gaussian distribution.

Problem 4.21 You are given a piece of code that can generate independent samples from the 1-dimensional standard normal distribution. You are also given a linear algebra package that can perform eigendecomposition of matrices. Show how you can use these pieces of code to efficiently generate independent samples from the d -dimensional Gaussian distribution with mean vector $\vec{\mu}$ and covariance matrix C .

4.11 The Student's t -Distribution

The Student's t -distribution is defined as a function of the normal distribution, and it is helpful for analyzing observations of normally distributed data in which the population variance is not known. This type of situation arises often in *hypothesis testing*, where only observed data are available but not the population variance. A population mean value is suspected (or *hypothesized*). Therefore, one can assume a population mean but not a population variance. In other words, the variance needs to be estimated from the sample. The resulting standardization, therefore, has additional variability caused by the estimation of the variance from the specific sample at hand. Therefore, the resulting standardized

variable is no longer drawn from a standard normal distribution but a *t*-distribution (which has a very similar shape to a standard normal distribution but with greater variability). In the following, we will concretely show how to derive the *t*-distribution from m independent samples from the normal distribution.

Let $X_1 \dots X_m$ be independent random variables drawn from a normal distribution with mean μ and variance σ^2 :

$$X_i \sim \mathcal{N}(\mu, \sigma^2)$$

Let the random variable corresponding to the mean of these values be \bar{X} :

$$\bar{X} = \frac{\sum_{i=1}^m X_i}{m}$$

The random variable \bar{X} has an expected value of μ and a variance of σ^2/m , since the different X_i are i.i.d. random variables. It is easy to standardize \bar{X} to $Z \sim \mathcal{N}(0, 1)$:

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{m}} \tag{4.5}$$

This type of standardization is useful in a variety of extreme-value analysis applications. However, in cases where the *population-level* standard-deviation σ is not known, one is compelled to use the *sample-level* standard deviation $\hat{\sigma}$ for standardization. On making this replacement, the newly “standardized” random variable T is as follows:

$$T = \frac{\bar{X} - \mu}{\sqrt{\sum_{i=1}^m (X_i - \bar{X})^2 / [m(m-1)]}} \tag{4.6}$$

The random variable T is distinct from the standard normal variable Z . This is because the denominator of Equation 4.6 is a random variable rather than the constant σ/\sqrt{m} (as in Equation 4.5). The random variable T follows a *t*-distribution with $\nu = (m-1)$ degrees of freedom. The probability density function of the random variable T with ν -degrees of freedom is as follows:

$$f_T(t) = \frac{\Gamma((\nu-1)/2)}{\Gamma(\nu/2)\sqrt{\nu\pi}} \left(1 + \frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

The *gamma function* $\Gamma(p)$ is defined as $\Gamma(p) = \int_0^\infty x^{p-1} \exp(-x) dx$. For positive integer values of p , the Gamma function is equal to $(p-1)!$. Although the generating process of the *t*-distribution is described above for positive integer values of ν , it is possible to instantiate the PDF for positive real values of ν as well.

When a random variable belongs to the *t*-distribution with $\nu > 0$ degrees of freedom, it is denoted as follows:

$$T \sim t(\nu)$$

The *t*-distribution has a bell-shaped curve like the standard normal distribution but it has appreciably heavier tails for small values of ν . The root of the greater dispersion of the *t*-distribution is the use of the sample-level (rather than population-level) standard deviation for normalization (which is notable for small sample sizes). Examples of $t(1)$, $t(5)$, and the standard normal distribution are illustrated in Figure 4.15. It is evident that both $t(1)$ and $t(5)$ have similar shape as the standard normal distribution but with heavier tails. Furthermore, $t(5)$ is much closer to the standard normal distribution as compared to $t(1)$. With increasing value of the sample size m (or degrees of freedom), the denominator of Equation 4.6 converges

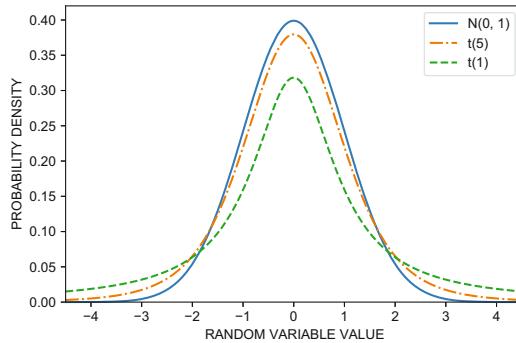


Figure 4.15: A standard normal distribution and two t -distributions with 1 and 5 degrees of freedom

to that of Equation 4.5, causing the t -distribution to also converge to the standard normal distribution. When observed data is used to construct a sample from the t -distribution by using Equation 4.6, it is referred to as a *t -value*.

The t -distribution is always centered at 0 like the standard normal distribution but its variance reduces with increasing number of degrees of freedom ν (or sample size):

$$\mu_T = 0, \quad \sigma_T^2 = \begin{cases} \infty & \text{if } \nu \leq 2 \\ \nu/(\nu - 2) & \text{if } \nu > 2 \end{cases}$$

Like the normal distribution, the t -distribution has tremendous utility in applications involving extreme-value analysis. The t -values derived from Equation 4.6 are compared to cut-off t -values corresponding to specific tail probabilities. An example of the t -distribution table is shown in Figure 4.16. The table shows *critical* values of t at which the upper-tail probability is equal to particular standardized values that are used frequently. A few trends are apparent on careful examination of the table. First, note that the Z -value in the final row is not very different from the t -value at 100 degrees of freedom. Even the use of 30 degrees of freedom provides an excellent approximation. This means that one can often use the normal distribution (instead of the more complicated t -distribution) for sample sizes greater than 30.

Example 4.27 The t -distribution is “standardized” like a standard normal distribution in the sense that it has mean 0 and variance that converges to 1 with increasing degrees of freedom. Propose a scaled and translated generalization of the t -distribution, which has three parameters corresponding to the location μ , convergent variance σ^2 , and degrees of freedom ν . This distribution should converge to the normal distribution $\mathcal{N}(\mu, \sigma^2)$ with increasing degrees of freedom, and it should be identical to the t -distribution for $\mu = 0$, $\sigma^2 = 1$. What is the variance of this random variable at ν degrees of freedom?

Solution: Let T be the t -distribution with μ degrees of freedom. The random variable $X = \mu T + \sigma$ satisfies all the conditions of the above problem. Example 3.28 discusses how one can generate the distribution corresponding to X because X is an affine function of T . In particular the variable t in the PDF of the t -distribution is replaced

Degrees of Freedom (rows)	Upper-Tail Probability Values (columns)								
	0.25	0.1	0.05	0.025	0.01	0.005	0.001	0.0005	
1	1.0000	3.0777	6.3138	12.7062	31.8205	63.6567	318.3088	636.6192	
2	0.8165	1.8856	2.9200	4.3027	6.9646	9.9248	22.3271	31.5991	
3	0.7649	1.6377	2.3534	3.1824	4.5407	5.8409	10.2145	12.9240	
4	0.7407	1.5332	2.1318	2.7764	3.7469	4.6041	7.1732	8.6103	
5	0.7267	1.4759	2.0150	2.5706	3.3649	4.0321	5.8934	6.8688	
6	0.7176	1.4398	1.9432	2.4469	3.1427	3.7074	5.2076	5.9588	
7	0.7111	1.4149	1.8946	2.3646	2.9980	3.4995	4.7853	5.4079	
8	0.7064	1.3968	1.8595	2.3060	2.8965	3.3554	4.5008	5.0413	
9	0.7027	1.3830	1.8331	2.2622	2.8214	3.2498	4.2968	4.7809	
10	0.6998	1.3722	1.8125	2.2281	2.7638	3.1693	4.1437	4.5869	
11	0.6974	1.3634	1.7959	2.2010	2.7181	3.1058	4.0247	4.4370	
12	0.6955	1.3562	1.7823	2.1788	2.6810	3.0545	3.9296	4.3178	
13	0.6938	1.3502	1.7709	2.1604	2.6503	3.0123	3.8520	4.2208	
14	0.6924	1.3450	1.7613	2.1448	2.6245	2.9768	3.7874	4.1405	
15	0.6912	1.3406	1.7531	2.1314	2.6025	2.9467	3.7328	4.0728	
16	0.6901	1.3368	1.7459	2.1199	2.5835	2.9208	3.6862	4.0150	
17	0.6892	1.3334	1.7396	2.1098	2.5669	2.8982	3.6458	3.9651	
18	0.6884	1.3304	1.7341	2.1009	2.5524	2.8784	3.6105	3.9216	
19	0.6876	1.3277	1.7291	2.0930	2.5395	2.8609	3.5794	3.8834	
20	0.6870	1.3253	1.7247	2.0860	2.5280	2.8453	3.5518	3.8495	
21	0.6864	1.3232	1.7207	2.0796	2.5176	2.8314	3.5272	3.8193	
22	0.6858	1.3212	1.7171	2.0739	2.5083	2.8188	3.5050	3.7921	
23	0.6853	1.3195	1.7139	2.0687	2.4999	2.8073	3.4850	3.7676	
24	0.6848	1.3178	1.7109	2.0639	2.4922	2.7969	3.4668	3.7454	
25	0.6844	1.3163	1.7081	2.0595	2.4851	2.7874	3.4502	3.7251	
26	0.6840	1.3150	1.7056	2.0555	2.4786	2.7787	3.4350	3.7066	
27	0.6837	1.3137	1.7033	2.0518	2.4727	2.7707	3.4210	3.6896	
28	0.6834	1.3125	1.7011	2.0484	2.4671	2.7633	3.4082	3.6739	
29	0.6830	1.3114	1.6991	2.0452	2.4620	2.7564	3.3962	3.6594	
30	0.6828	1.3104	1.6973	2.0423	2.4573	2.7500	3.3852	3.6460	
35	0.6816	1.3062	1.6896	2.0301	2.4377	2.7238	3.3400	3.5911	
40	0.6807	1.3031	1.6839	2.0211	2.4233	2.7045	3.3069	3.5510	
45	0.6800	1.3006	1.6794	2.0141	2.4121	2.6896	3.2815	3.5203	
50	0.6794	1.2987	1.6759	2.0086	2.4033	2.6778	3.2614	3.4960	
60	0.6786	1.2958	1.6706	2.0003	2.3901	2.6603	3.2317	3.4602	
70	0.6780	1.2938	1.6669	1.9944	2.3808	2.6479	3.2108	3.4350	
80	0.6776	1.2922	1.6641	1.9901	2.3739	2.6387	3.1953	3.4163	
90	0.6772	1.2910	1.6620	1.9867	2.3685	2.6316	3.1833	3.4019	
100	0.6770	1.2901	1.6602	1.9840	2.3642	2.6259	3.1737	3.3905	
∞ (Z-value)	0.6745	1.2816	1.6449	1.9600	2.3263	2.5758	3.0902	3.2905	

Figure 4.16: The critical t -values at various degrees of freedom (rows) and important upper-tail probabilities (columns). Each entry in the table shows the critical (one-tailed) t -value at a specific degree of freedom and upper-tail probability. For example, the entry 2.9200 in the second row (two degrees of freedom) of the third column (upper-tail probability of 0.05) means that $P(T > 2.9200) \approx 0.05$ when $T \sim t(2)$. Each entry for tail probability p and ν degrees of freedom was generated using the Excel function `t.inv(1 - p, ν)`.

with $(x - \mu)/\sigma$. Therefore, the PDF of the generalized t -distribution is as follows:

$$f_X(x) = \frac{\Gamma((\nu - 1)/2)}{\Gamma(\nu/2)\sqrt{\nu\pi}} \left(1 + \frac{(x - \mu)^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}}$$

The variance σ_X^2 is $\sigma^2\sigma_T^2$ using the results on variance of affine functions of random variables (cf. Lemma 3.11). Plugging in the variance of the t -distribution at ν degrees of freedom, one obtains $\sigma_X^2 = \sigma^2\nu/(\nu - 2)$ for $\nu > 2$ and ∞ , otherwise. ■

Example 4.28 The lengths of widgets produced by a factory are drawn from a normal distribution with mean 100 centimeters. Widgets are divided into batches of 9 widgets, and each batch is given a percentile rank based on its average length. What is the percentile rank of a batch for which the average length is 101 centimeters when (a) the standard deviation of widget lengths of the population produced by the factory is known to be 3 centimeters, and (b) the population standard deviation of widget lengths is not known, but the sample standard deviation of this specific batch of 9 widgets at hand is 3 centimeters?

Solution: This example is useful because it clearly illustrates the difference between the cases where one must use the Z-value and those where one must use the t -value. In the case where the population standard deviation is known, one can standardize the sample of 101 centimeters to the standard normal distribution to obtain the Z-value. The standard-deviation of the individual widgets is $\sigma = 3$, and the standard-deviation of the average of a batch of 9 widgets is $\sigma/\sqrt{9} = 1$. Therefore, one obtains the following Z-value:

$$z = \frac{(101 - 100)}{\sigma/\sqrt{9}} = \frac{1}{(3/3)} = 1$$

By using the normal distribution tables, one obtains the percentile rank as $100*P(Z \leq 1) = 84.13$. In the case where the population standard deviation is not known, one must substitute the *sample standard deviation* to obtain a t -value with $(9 - 1) = 8$ degrees of freedom:

$$t = \frac{(101 - 100)}{\hat{\sigma}/\sqrt{9}} = \frac{1}{(3/3)} = 1$$

Note that the computations are identical in the two cases, except that the sample standard deviation was used in the second case to obtain a t -value. This t -value was mapped to a tail probability and the corresponding percentile rank turns out to be 82.67. Although Figure 4.16 is an inverse mapping from probabilities to critical values, one can compute the forward mapping from t -value to probability using any statistical software. An example is the use of the Excel function `t.dist(t, k, 1)` for $t = 1$ and $k = 8$. This percentile value is lower than in the normally distributed case because of the heavier tails of the t -distribution. In other words, the additional “uncertainty” caused by using the sample standard deviation for normalization of the length difference from mean (instead of using the population standard deviation to create a standard normal random variable) lowers the percentile rank of the batch. ■

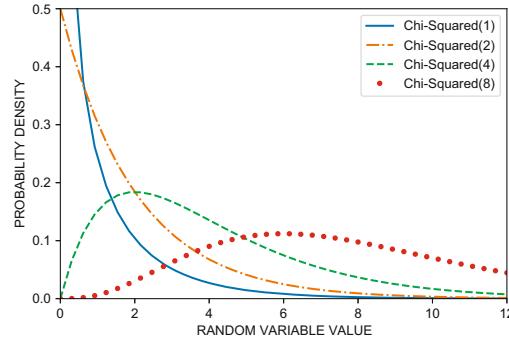


Figure 4.17: The χ^2 -distribution at various degrees of freedom

Problem 4.22 *The number of minor blemishes in the cars acquired by a used-car dealer is normally distributed with a mean of 17.2 blemishes. The dealer examines a batch of four cars and finds it to have an average of 18.5 blemishes per car. Should the dealer consider this batch to be among the top-2.5 percent of four-car batches with most blemishes, if (a) the population variance of the number of blemishes in individual cars is known to be 1, and (b) if the population variance is not known but the sample variance of the number of blemishes in the specific batch of four cars is 1?*

4.12 The χ^2 -Distribution

The χ^2 -distribution (also referred to as the chi-squared distribution) is most simply defined as the sum of squares of independent and identically distributed (i.i.d.) standard normal random variables. Let Z_1, Z_2, \dots, Z_k be k independent standard normal random variables:

$$Z_i \sim \mathcal{N}(0, 1)$$

Then, consider the following random variable derived from k independent standard normal variables:

$$V = \sum_{i=1}^k Z_i^2$$

The random variable V belongs to a χ^2 -distribution with k degrees of freedom, for which the probability density function is defined as follows:

$$f_V(v) = \frac{1}{2^{k/2}\Gamma(k/2)} v^{k/2-1} \exp(-v/2)$$

The above PDF takes on nonzero values only for $v \geq 0$, since it is defined using a sum of squares of normal random variables. The notation $\Gamma(\cdot)$ refers to the same gamma function that was defined in section 4.11 on the Student's t -distribution. When V belongs to the χ^2 -distribution with k degrees of freedom, it is denoted as follows:

$$V \sim \chi^2(k)$$

The only parameter for the χ^2 -distribution is the number of degrees of freedom k . The χ^2 -distribution for various degrees of freedom is shown in Figure 4.17. At one degree of

freedom, the χ^2 -distribution is the square of a standard normal distribution, which looks awfully like an exponential distribution. This is not particularly surprising because the normal distribution also has an exponential form of the probability density function (albeit with a different exponent), and squaring the normal random variable maps it to the domain of nonnegative values (like an exponentially distributed variable). As the number of degrees of freedom increases, the distribution becomes increasingly symmetric and starts resembling the familiar bell-shaped curve of the normal distribution. Note that since the χ^2 -distribution is defined by summing k independent random variables, it converges to the normal distribution as $k \rightarrow \infty$ (because of the central limit theorem).

The mean and variance of a variable of a random variable $V = \sum_i Z_i^2$ drawn from the χ^2 -distribution are as follows (see Example 4.35):

$$\mu_V = k, \quad \sigma_V^2 = 2k$$

For a large number of degrees of freedom k , the χ^2 -distribution converges to the following normal distribution:

$$\chi^2(k) \rightarrow \mathcal{N}(k, 2k)$$

Various hypothesis tests that use the χ^2 -distribution will be discussed in detail in Chapter 5. The inverse cumulative distribution tables for the upper-tail and lower-tail probabilities are provided in Figures 4.18 and 4.19, respectively. Both tables are needed because this distribution is not symmetric about the mean. The χ^2 -distribution satisfies closure under addition:

Lemma 4.1 *Let X_1 and X_2 be two independent random variables drawn from χ^2 -distributions with degrees of freedom k_1 and k_2 . Then $(X_1 + X_2)$ has a χ^2 -distribution with $(k_1 + k_2)$ degrees of freedom.*

Another remarkable property of the χ^2 -distribution is Cochran's Theorem, which turns out to be useful in hypothesis testing:

Theorem 4.1 (Cochran's Theorem) *Let X_1, X_2, \dots, X_n be i.i.d. normal random variables with mean \bar{X} and population standard deviation σ_0 . Let U be the random variable defined by the following:*

$$U = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{\sigma_0^2}$$

Then, the random variable U is drawn from a χ^2 -distribution with $(n-1)$ -degrees of freedom. In other words, U satisfies $U \sim \chi^2(n-1)$.

A formal proof of this result is omitted.

Example 4.29 Mr. Archer's shots at a bullseye have an aim that is expected to be centered at the target but with some random variability. The X - and Y -coordinates of his shots are statistically independent — each coordinate is normally distributed with a mean at the corresponding bullseye coordinate but with a standard deviation of 2 centimeters. What is the maximum Euclidean distance from the bullseye of the best 10% of his shots? What is the minimum distance from the bullseye of the worst 10% of his shots?

Solution: The squared Euclidean distance of each of his shots from the bullseye, when scaled by 2^2 , follows a χ^2 -distribution with two degrees of freedom. The lower-tail and upper-tail critical values of the $\chi^2(2)$ distribution at the 10% threshold are 0.2107

Degrees of Freedom (rows)	Upper-Tail Probability Values (columns)							
	0.25	0.1	0.05	0.025	0.01	0.005	0.001	0.0005
1	1.3233	2.7055	3.8415	5.0239	6.6349	7.8794	10.8276	12.1157
2	2.7726	4.6052	5.9915	7.3778	9.2103	10.5966	13.8155	15.2018
3	4.1083	6.2514	7.8147	9.3484	11.3449	12.8382	16.2662	17.7300
4	5.3853	7.7794	9.4877	11.1433	13.2767	14.8603	18.4668	19.9974
5	6.6257	9.2364	11.0705	12.8325	15.0863	16.7496	20.5150	22.1053
6	7.8408	10.6446	12.5916	14.4494	16.8119	18.5476	22.4577	24.1028
7	9.0371	12.0170	14.0671	16.0128	18.4753	20.2777	24.3219	26.0178
8	10.2189	13.3616	15.5073	17.5345	20.0902	21.9550	26.1245	27.8680
9	11.3888	14.6837	16.9190	19.0228	21.6660	23.5894	27.8772	29.6658
10	12.5489	15.9872	18.3070	20.4832	23.2093	25.1882	29.5883	31.4198
11	13.7007	17.2750	19.6751	21.9200	24.7250	26.7568	31.2641	33.1366
12	14.8454	18.5493	21.0261	23.3367	26.2170	28.2995	32.9095	34.8213
13	15.9839	19.8119	22.3620	24.7356	27.6882	29.8195	34.5282	36.4778
14	17.1169	21.0641	23.6848	26.1189	29.1412	31.3193	36.1233	38.1094
15	18.2451	22.3071	24.9958	27.4884	30.5779	32.8013	37.6973	39.7188
16	19.3689	23.5418	26.2962	28.8454	31.9999	34.2672	39.2524	41.3081
17	20.4887	24.7690	27.5871	30.1910	33.4087	35.7185	40.7902	42.8792
18	21.6049	25.9894	28.8693	31.5264	34.8053	37.1565	42.3124	44.4338
19	22.7178	27.2036	30.1435	32.8523	36.1909	38.5823	43.8202	45.9731
20	23.8277	28.4120	31.4104	34.1696	37.5662	39.9968	45.3147	47.4985
21	24.9348	29.6151	32.6706	35.4789	38.9322	41.4011	46.7970	49.0108
22	26.0393	30.8133	33.9244	36.7807	40.2894	42.7957	48.2679	50.5111
23	27.1413	32.0069	35.1725	38.0756	41.6384	44.1813	49.7282	52.0002
24	28.2412	33.1962	36.4150	39.3641	42.9798	45.5585	51.1786	53.4788
25	29.3389	34.3816	37.6525	40.6465	44.3141	46.9279	52.6197	54.9475
26	30.4346	35.5632	38.8851	41.9232	45.6417	48.2899	54.0520	56.4069
27	31.5284	36.7412	40.1133	43.1945	46.9629	49.6449	55.4760	57.8576
28	32.6205	37.9159	41.3371	44.4608	48.2782	50.9934	56.8923	59.3000
29	33.7109	39.0875	42.5570	45.7223	49.5879	52.3356	58.3012	60.7346
30	34.7997	40.2560	43.7730	46.9792	50.8922	53.6720	59.7031	62.1619
35	40.2228	46.0588	49.8018	53.2033	57.3421	60.2748	66.6188	69.1986
40	45.6160	51.8051	55.7585	59.3417	63.6907	66.7660	73.4020	76.0946
45	50.9849	57.5053	61.6562	65.4102	69.9568	73.1661	80.0767	82.8757
50	56.3336	63.1671	67.5048	71.4202	76.1539	79.4900	86.6608	89.5605
60	66.9815	74.3970	79.0819	83.2977	88.3794	91.9517	99.6072	102.6948
70	77.5767	85.5270	90.5312	95.0232	100.4252	104.2149	112.3169	115.5776
80	88.1303	96.5782	101.8795	106.6286	112.3288	116.3211	124.8392	128.2613
90	98.6499	107.5650	113.1453	118.1359	124.1163	128.2989	137.2084	140.7823
100	109.1412	118.4980	124.3421	129.5612	135.8067	140.1695	149.4493	153.1670

Figure 4.18: The critical values at various degrees of freedom (rows) for various upper-tail probabilities (columns) of the χ^2 -distribution. Each entry in the table shows the critical value of a random variable drawn from the χ^2 -distribution for which the upper-tail probability takes on a specific value. For example, the entry 5.9915 in the second row (two degrees of freedom) of the third column (upper-tail probability of 0.05) means that $P(V > 5.9915) \approx 0.05$ when $V \sim \chi^2(2)$. Each entry for tail probability p and k degrees of freedom was generated using the Excel function `chisq.inv(1 - p, k)`.

Degrees of Freedom (rows)	Lower-Tail Probability Values (columns)							
	0.25	0.1	0.05	0.025	0.01	0.005	0.001	0.0005
1	0.1015	0.0158	0.0039	0.0010	0.0002	0.0000	0.0000	0.0000
2	0.5754	0.2107	0.1026	0.0506	0.0201	0.0100	0.0020	0.0010
3	1.2125	0.5844	0.3518	0.2158	0.1148	0.0717	0.0243	0.0153
4	1.9226	1.0636	0.7107	0.4844	0.2971	0.2070	0.0908	0.0639
5	2.6746	1.6103	1.1455	0.8312	0.5543	0.4117	0.2102	0.1581
6	3.4546	2.2041	1.6354	1.2373	0.8721	0.6757	0.3811	0.2994
7	4.2549	2.8331	2.1673	1.6899	1.2390	0.9893	0.5985	0.4849
8	5.0706	3.4895	2.7326	2.1797	1.6465	1.3444	0.8571	0.7104
9	5.8988	4.1682	3.3251	2.7004	2.0879	1.7349	1.1519	0.9717
10	6.7372	4.8652	3.9403	3.2470	2.5582	2.1559	1.4787	1.2650
11	7.5841	5.5778	4.5748	3.8157	3.0535	2.6032	1.8339	1.5868
12	8.4384	6.3038	5.2260	4.4038	3.5706	3.0738	2.2142	1.9344
13	9.2991	7.0415	5.8919	5.0088	4.1069	3.5650	2.6172	2.3051
14	10.1653	7.7895	6.5706	5.6287	4.6604	4.0747	3.0407	2.6967
15	11.0365	8.5468	7.2609	6.2621	5.2293	4.6009	3.4827	3.1075
16	11.9122	9.3122	7.9616	6.9077	5.8122	5.1422	3.9416	3.5358
17	12.7919	10.0852	8.6718	7.5642	6.4078	5.6972	4.4161	3.9802
18	13.6753	10.8649	9.3905	8.2307	7.0149	6.2648	4.9048	4.4394
19	14.5620	11.6509	10.1170	8.9065	7.6327	6.8440	5.4068	4.9123
20	15.4518	12.4426	10.8508	9.5908	8.2604	7.4338	5.9210	5.3981
21	16.3444	13.2396	11.5913	10.2829	8.8972	8.0337	6.4467	5.8957
22	17.2396	14.0415	12.3380	10.9823	9.5425	8.6427	6.9830	6.4045
23	18.1373	14.8480	13.0905	11.6886	10.1957	9.2604	7.5292	6.9237
24	19.0373	15.6587	13.8484	12.4012	10.8564	9.8862	8.0849	7.4527
25	19.9393	16.4734	14.6114	13.1197	11.5240	10.5197	8.6493	7.9910
26	20.8434	17.2919	15.3792	13.8439	12.1981	11.1602	9.2221	8.5379
27	21.7494	18.1139	16.1514	14.5734	12.8785	11.8076	9.8028	9.0932
28	22.6572	18.9392	16.9279	15.3079	13.5647	12.4613	10.3909	9.6563
29	23.5666	19.7677	17.7084	16.0471	14.2565	13.1211	10.9861	10.2268
30	24.4776	20.5992	18.4927	16.7908	14.9535	13.7867	11.5880	10.8044
35	29.0540	24.7967	22.4650	20.5694	18.5089	17.1918	14.6878	13.7875
40	33.6603	29.0505	26.5093	24.4330	22.1643	20.7065	17.9164	16.9062
45	38.2910	33.3504	30.6123	28.3662	25.9013	24.3110	21.2507	20.1366
50	42.9421	37.6886	34.7643	32.3574	29.7067	27.9907	24.6739	23.4610
60	52.2938	46.4589	43.1880	40.4817	37.4849	35.5345	31.7383	30.3405
70	61.6983	55.3289	51.7393	48.7576	45.4417	43.2752	39.0364	37.4674
80	71.1445	64.2778	60.3915	57.1532	53.5401	51.1719	46.5199	44.7910
90	80.6247	73.2911	69.1260	65.6466	61.7541	59.1963	54.1552	52.2758
100	90.1332	82.3581	77.9295	74.2219	70.0649	67.3276	61.9179	59.8957

Figure 4.19: The critical values at various degrees of freedom (rows) for various lower-tail probabilities (columns) of the χ^2 -distribution. Each entry in the table shows the critical value of a random variable drawn from the χ^2 -distribution for which the lower-tail probability takes on a specific value. For example, the entry 0.1026 in the second row (two degrees of freedom) of the third column (lower-tail probability of 0.05) means that $P(V < 0.1026) \approx 0.05$ when $V \sim \chi^2(2)$. Each entry for tail probability p and k degrees of freedom was generated using the Excel function `chisq.inv(p, k)`.

and 4.6052, respectively. Therefore, the corresponding squared Euclidean distances are $2^2 \times 0.2107$ and $2^2 \times 4.6052 \text{ cm}^2$, respectively. In other words, the maximum Euclidean distance for the best 10% of his shots is $2\sqrt{0.2107} \approx 0.918$ centimeters, and the minimum distance for the worst 10% of his shots is $2\sqrt{4.6052} \approx 4.292$ centimeters. ■

Example 4.30 The velocity of a gas particle along each of the positive X-, Y-, and Z- directions is an independent normal random variable with mean 0 and variance $8314T/M \text{ m}^2/\text{s}^2$, where T is the temperature in Kelvin and M is the particle mass in atomic mass units (amu). A negative value of the random variable indicates a negative direction of movement. You have a large tank at 300 Kelvin containing two isotopes of helium with masses 3 and 4 amu. The respective abundances are 1% and 99%. A particle trap can uniformly sample atoms in the tank and isolate those with (3-dimensional) speed greater than 3000 m/s. What percentage of the trapped atoms will be the lighter isotope of helium? Assume that the percentage of trapped atoms is so small that the remaining atoms in the tank continue to have practically the same isotope distribution as that in the initial mixture.

Solution: The squared speed of a helium atom is the sum of squares of three identical normal distributions with zero mean and variance $8314T/M$ (which is $831400 \text{ m}^2/\text{s}^2$ for $M = 3$ and $623550 \text{ m}^2/\text{s}^2$ for $M = 4$). Dividing the squared speed by the isotope-specific variance yields a $\chi^2(3)$ random variable. A minimum speed threshold of 3000 m/s maps to a $\chi^2(3)$ -threshold of $3000^2/831400 = 10.825$ for the lighter isotope and $3000^2/623550 = 14.433$ for the heavier isotope. Although Figure 4.18 is not sufficiently detailed to provide specific upper-tail probabilities, one can use data analysis software functions, such as the `chisq.dist(v, 3, 1)` function in Excel, to compute this probability:

$$\begin{aligned} P(\text{Trap}|\text{Light}) &= 1 - \text{chisq.dist}(10.825, 3, 1) = 0.01271 \\ P(\text{Trap}|\text{Heavy}) &= 1 - \text{chisq.dist}(14.433, 3, 1) = 0.002371 \end{aligned}$$

One can then compute $P(\text{Light}|\text{Trap})$ using the Bayes rule:

$$P(\text{Light}|\text{Trap}) = \frac{0.01 * 0.01271}{0.01 * 0.01271 + 0.99 * 0.002371} = 0.0514$$

In other words, even though the tank contains only 1% of the lighter isotope of helium, the trapped particles were enriched to 5.14% in light isotope atoms. ■

4.12.1 Application: Mahalanobis Method for Outlier Detection

Aside from hypothesis testing, a useful application of the χ^2 -distribution occurs in outlier detection with the use of the Mahalanobis distance (cf. section 2.4.2 of Chapter 2 and section 9.4.1 of Chapter 9). As discussed in section 9.4.1, the squared Mahalanobis distance is used as an outlier score, and it is derived from the exponent of the following d -dimensional multivariate Gaussian distribution:

$$f_{\vec{X}}(\vec{x}) = \frac{1}{(2\pi)^{d/2}|C|^{1/2}} \exp\left(-\frac{[\vec{x} - \vec{\mu}]C^{-1}[\vec{x} - \vec{\mu}]^T}{2}\right) \quad (4.7)$$

Here, $\vec{\mu}$ is the d -dimensional vector containing the distribution mean and C is the distribution covariance (matrix). The squared Mahalanobis distance of the d -dimensional data point \vec{x} from the center of the distribution (i.e., outlier score of \vec{x}) is as follows:

$$\text{Score}(\vec{x}) = [\vec{x} - \vec{\mu}]C^{-1}[\vec{x} - \vec{\mu}]^T$$

The Gaussian density is low when the score is large, which justifies the use of the scores for quantification of outliers. However, the density is not an interpretable measure of outliers; a better approach is to quantify the probability that a randomly chosen point from the data will have a higher outlier score. In order to do so, one first needs to know the probability distribution of outlier scores. As discussed in section 4.10.2, the squared Mahalanobis distance in the exponent of the Gaussian distribution is the sum of the squares of d standard normal variables. In other words, the outlier score is a χ^2 -distribution with d degrees of freedom. Therefore, one can use the tail probabilities of Figure 4.18 to quantify the degree to which a point may be considered an outlier after computing its squared Mahalanobis distance.

Example 4.31 Consider a whitened data set that is modeled as a d -dimensional Gaussian distribution with independent attributes, each of which has mean 0 and variance 1. Let v be a sample of the squared Euclidean distance between any pair of points in the data set. Find the PDF $f_V(v)$ of this squared Euclidean distance.

Solution: Let $[X_1, X_2, \dots, X_d]$ and $[Y_1, Y_2, \dots, Y_n]$ be two random vectors corresponding to samples from the data set. Since each X_i and Y_i is an independent normal random variables with mean 0 and variance 1, $(X_i - Y_i)$ is also a normal random variables with mean 0 and variance 2 (because of the closure property of normal random variables). Consider the random variable W corresponding to half the squared Euclidean distance:

$$W = \sum_{i=1}^d \frac{(X_i - Y_i)^2}{2}$$

It is evident that W is a χ^2 -distribution with d -degrees of freedom with the following distribution:

$$f_W(w) = \frac{1}{2^{d/2}\Gamma(d/2)} w^{d/2-1} \exp(-w/2)$$

The random variable V corresponding to the squared Euclidean distance is simply equal to $2W$. Using the results on functions of random variables (cf. Example 3.28), the distribution of V is as follows:

$$f_V(v) = \frac{1}{2^{1+(d/2)}\Gamma(d/2)} (v/2)^{d/2-1} \exp(-v/4)$$

4.13 Mixture Distributions: The Realistic View

The probability distributions discussed thus far are somewhat homogeneous in the sense that a single closed-form expression defines the probability density in all regions of the

data. However, the observed data in the real world are rarely so homogenous, and multiple density functions may be needed to recreate different parts of the data. In a sense, the probability distributions discussed thus far are the building blocks of modeling — mixture models are the glue that binds these building blocks together to create a more complex and realistic distribution. The observed data encountered in machine learning applications often require the additional complexity of mixture modeling.

4.13.1 Why Mixtures are Ubiquitous: A Motivating Example

In order to motivate mixture models, we will use a real-world example. Consider the case where we are trying to model the heights of gorillas. While a simplistic approach would be to model all heights with a single normal distribution, males and females are different enough that the true distribution is bimodal (with different peaks for males and females). While humans also show differences in distribution between the heights of males and females, the differences are particularly pronounced for gorillas. This behavior is referred to as *sexual dimorphism*, and it is indicative of fundamental differences in “generative mechanisms” of the heights of female and male gorillas in the real world. As a result, it is no longer reasonable to model the entire population of gorilla heights with a *single* normal distribution (even as an approximation) but with two different normal distributions; one distribution is for males and the other is for females. Male gorillas are six feet tall on the average, whereas female gorillas are four feet tall on the average. The heights of both the male group and the female group are (roughly) normally distributed, with males exhibiting significantly greater variability in height as compared to females. We denote the (normally distributed) *conditional* density function of males by $f_{X|\text{male}}(x)$ and the density function of females by $f_{X|\text{female}}(x)$. Since both males and females each have 0.5 probability to be represented in the population, one can determine the unconditional density function $f_X(x)$ of the height of all gorillas by using the total probability rule discussed in Chapter 3:

$$\begin{aligned} f_X(x) &= P(\text{female})f_{X|\text{female}}(x) + P(\text{male})f_{X|\text{male}}(x) \\ &= [f_{X|\text{female}}(x) + f_{X|\text{male}}(x)]/2 \end{aligned}$$

The probability density functions of the heights of female gorillas, male gorillas, and the heights of all gorillas are shown in Figure 4.20(a). It is evident that the heights of all gorillas are *not* normally distributed, although the heights can be represented as a mixture of two normal distributions. The individual probability distributions for male and female gorillas are referred to as *mixture components*. In general, a mixture component is the probability distribution of one of the generating mechanisms in the data (e.g., probability distribution of female gorilla heights).

We have repeated the same exercise with male and female heights from the US (human) population and reported the results in Figure 4.20(b). In this case, a unimodal distribution is obtained because there are large overlapping regions between the two distributions, and one can use a single normal distribution to model the entire data as a rough approximation.

4.13.2 The Basic Generative Process of a Mixture Model

In this section, we will discuss a basic generative process by which one instance of the observed data is produced by a mixture model. The probability distribution of the full population is denoted by $f_{\vec{X}}(\vec{x})$, although the mixture components have their own conditional probability distributions. We assume that the different mixture components are denoted by

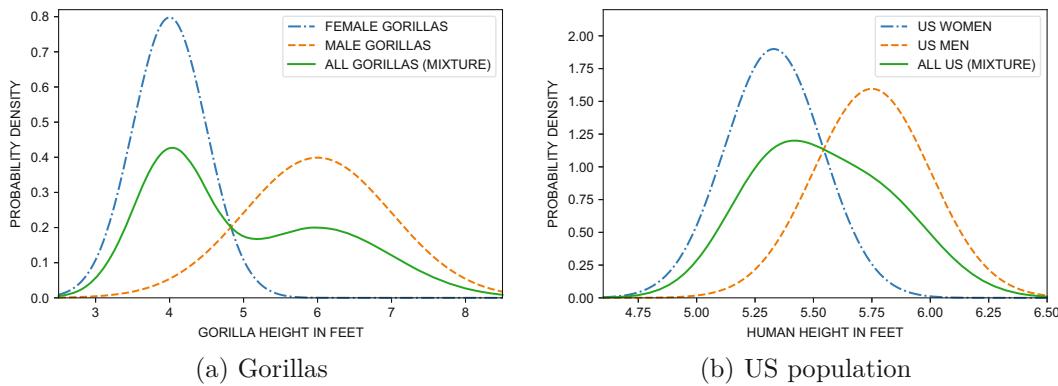


Figure 4.20: The mixture distributions for gorilla and US population height

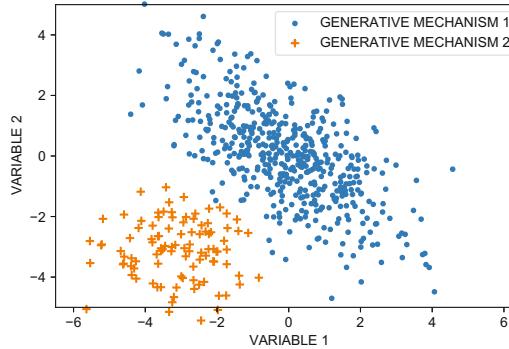


Figure 4.21: A mixture of two multivariate Gaussian distributions

$\mathcal{G}_1, \dots, \mathcal{G}_k$, with corresponding conditional probability distributions denoted by $f_{\vec{X}|\mathcal{G}_1}(\vec{x})$, $f_{\vec{X}|\mathcal{G}_2}(\vec{x}), \dots, f_{\vec{X}|\mathcal{G}_k}(\vec{x})$. Therefore, there are a total of k mixture components. In the case when the data are discrete, one can use probability mass functions $p_{\vec{X}|\mathcal{G}_1}(\vec{x})$, $p_{\vec{X}|\mathcal{G}_2}(\vec{x}), \dots, p_{\vec{X}|\mathcal{G}_k}(\vec{x})$ instead of probability density functions. In the earlier example of gorilla heights, here are $k = 2$ distinct generative processes for male and female heights; the corresponding distributions $f_{X|\mathcal{G}_1}(x)$ and $f_{X|\mathcal{G}_2}(x)$ are univariate normal distributions with their own sets of parameters. Another example of $k = 2$ generating distributions in two dimensions is given in Fig. 4.21. The probability that the i th generative process is selected in the generation of an instance is given by $P(\mathcal{G}_i)$. The value of $P(\mathcal{G}_i)$ is also a parameter of the mixture distribution. Since the probabilities of different mixture components sum to 1, we have the following:

$$\sum_{i=1}^k P(\mathcal{G}_i) = 1$$

The probability $P(\mathcal{G}_i)$ is the *prior* probability of an observed data point belonging to mixture component \mathcal{G}_i , because it refers to the probability that we would predict an observation to belong to mixture component \mathcal{G}_i without knowing anything about the data point. This prior probability shows up in the observed data in terms of the relative frequencies of the points belonging to different mixture components. In the example containing male and female

gorilla heights, the prior probabilities reflect the relative frequencies of male and female gorillas. The basic generative process for a single observation in the data is as follows:

1. Roll a biased die whose k sides have probabilities $P(\mathcal{G}_1) \dots P(\mathcal{G}_k)$. Let the outcome of the die roll be the side i , which provides the identity of the mixture component from which the observation is generated.
2. Sample the data point \vec{x} from the probability distribution $f_{\vec{X}|\mathcal{G}_i}(\vec{x})$. The point \vec{x} is the output of one iteration of the generative model.

The aforementioned process can be repeated again and again in order to generate a data set of arbitrary size. In machine learning settings, *it is assumed that the observed data is the outcome of precisely such a generative process*. An example could be the observed data of gorilla heights. The analyst chooses the number of family of distributions in the mixture — the parameters of the model (such as the prior probabilities and the parameters of the distribution components) are estimated using the observed data. These methods are described in Chapter 6.

Problem 4.23 Write a program to generate 1-dimensional data that is a mixture of three normal distributions, centered at -2 , 0 , and 1 . The three normal distributions have standard deviations of 0.4 , 0.5 and 1 . You may use a programming language of your choice.

4.13.3 Some Useful Results for Prediction

A number of useful results relate the conditional mixture probabilities to the unconditional values. Given the prior probabilities and the conditional density functions, one can also derive the *unconditional* density function for the entire data (containing all mixture components) with the use of the total probability rule from Chapter 3:

$$f_{\vec{X}}(\vec{x}) = \sum_{i=1}^k P(\mathcal{G}_i) f_{\vec{X}|\mathcal{G}_i}(\vec{x})$$

Aside from the unconditional probability distribution, it is often useful to estimate the *posterior probability* $P(\mathcal{G}_i|\vec{x})$, which not only accounts for the relative frequencies of the mixture components but also for the fact that knowing the value of \vec{x} changes the probability of the observation belonging to a particular component. For example, even though the probability of an observation being a male gorilla depends only on the relative frequencies of males and females, the information that a particular gorilla's height is 7 feet drastically increases the probability that it is male. This type of approach is used commonly for classification of data with the use of a *Bayes classifier*. The posterior probability can be estimated using the Bayes rule:

$$P(\mathcal{G}_i|\vec{X} = \vec{x}) = \frac{P(\mathcal{G}_i) f_{\vec{X}|\mathcal{G}_i}(\vec{x})}{\sum_{r=1}^k P(\mathcal{G}_r) f_{\vec{X}|\mathcal{G}_r}(\vec{x})} \quad (4.8)$$

The relationships in this section are used by many machine learning algorithms.

Example 4.32 The normal distribution of the US population generated in Figure 4.20(b) is a mixture distribution containing two normal components. Women have a mean height of 5.33 feet and a standard deviation of 0.21 feet, whereas men have a mean height of 5.75 feet and a standard deviation of 0.25 feet. What is the probability density function of the entire mixture distribution? Assume that men and

women have equal frequency of presence in the population. What is the probability that an individual with a height of 5.75 feet is male?

Solution: The unconditional probability density function of the entire distribution is generated using the total probability rule. The conditional density functions of the two distributions are as follows:

$$f_{X|\text{female}}(x) = \frac{\exp(-(x - 5.33)^2 / (2 * 0.21 * 0.21))}{\sqrt{2\pi} * 0.21}$$

$$f_{X|\text{male}}(x) = \frac{\exp(-(x - 5.75)^2 / (2 * 0.25 * 0.25))}{\sqrt{2\pi} * 0.25}$$

Then, the unconditional density function of the entire mixture model can be generated using the total probability rule is as follows:

$$f_X(x) = P(\text{female})f_{X|\text{female}}(x) + P(\text{male})f_{X|\text{male}}(x)$$

$$= \frac{1}{2} \frac{\exp(-(x - 5.33)^2 / (2 * 0.21 * 0.21))}{\sqrt{2\pi} * 0.21} + \frac{1}{2} \frac{\exp(-(x - 5.75)^2 / (2 * 0.25 * 0.25))}{\sqrt{2\pi} * 0.25}$$

Next, we compute the posterior probability of an individual with height 5.75 feet being male. Using the Bayes rule and ignoring constant terms like $1/\sqrt{2\pi}$ in the numerator and denominator, one obtains the following:

$$P(\text{male}|5.75) = \frac{\exp(0)/0.25}{\exp(0)/0.25 + \exp(-0.42^2/(2 * 0.21^2))/0.21}$$

$$= \frac{4}{4 + \exp(-2)/0.21} = \frac{4}{4 + 0.6445} \approx 0.86$$

Therefore, observing that a particular height is 5.75 feet (which is on the taller side) increases the probability of the sample belonging to a man to a 0.86 posterior value from the prior value of 0.5. ■

The aforementioned approach is often used for Bayes classification in machine learning.

Problem 4.24 Using the data in Example 4.32, find the posterior probability that a person with height 5.2 feet is a man.

4.13.4 The Conditional Independence Assumption

While the data attributes of most real-world data are correlated, a common assumption that is used in many of these settings is that of *conditional independence*. What conditional independence means is that the data are *locally uncorrelated* but *globally correlated*.

Let $\vec{x} = [x_1, \dots, x_d]$ be the d -dimensional observed data vector, which is an instance of the random variable vector \vec{X} . Then, the notion of conditional independence is described

in terms of the underlying probability density/mass function as follows:

$$p_{\vec{X}|\mathcal{G}_r}(\vec{x}) = \prod_{i=1}^d p_{X_i|\mathcal{G}_r}(x_i) \quad [\text{Discrete Random Variables}]$$

$$f_{\vec{X}|\mathcal{G}_r}(\vec{x}) = \prod_{i=1}^d f_{X_i|\mathcal{G}_r}(x_i) \quad [\text{Continuous Random Variables}]$$

This type of assumption is used to simplify multivariate distributions so that they can be expressed in terms of a fewer number of parameters. Fewer parameters enable accurate estimation of their values from less data (cf. Chapter 6).

4.14 Moments of Random Variables (*)

The moments of a random variable are the expected values of powers of the variable. Formally, the k th moment of a random variable is defined as follows:

Definition 4.2 (kth Moment) *The k th moment of a random variable X is the expected value of X^k . In other words the k th moment of X is $E[X^k]$.*

The value k is a nonnegative integer, and is referred to as the *order* of the moment.

We are already familiar with the first moment of a random variable, which is the mean $\mu_X = E[X]$. Furthermore, the variance of a random variable can be expressed in terms of the first moment and the second moment (cf. Lemma 3.10):

$$\sigma_X^2 = E[X^2] - E[X]^2$$

One can already see that two of the most useful properties of a distribution (mean and variance) can be expressed in terms of the moments of a distribution. In fact, the entire distribution can be reconstructed from all its moments (of orders from 1 to ∞).

Property 4.1 *The moments of a random variable provide an alternative and equivalent description of the random variable as the distribution function.*

In the following, several useful variations of the definition of moments are discussed. These variations do not always provide an alternative description of the random variable (unless paired with additional information).

4.14.1 Central and Standardized Moments

Aside from the aforementioned definition of a moment, many variants of the definition exist in terms of central or standardized moments. The k -th central moment basically computes the moment of the random variable X after mean centering it to $(X - \mu_X)$:

Definition 4.3 (kth Central Moment) *Let X be a random variable with expected value of $\mu_X = E[X]$. The k th central moment of X is the expected value of $(X - \mu_X)^k$. In other words, the k th central moment of X is $E[(X - \mu_X)^k]$.*

It is noteworthy that the central moment of the first order is always 0, because $E[X - \mu_X] = E[X] - \mu_X = 0$. The central moment of the second order is the variance because $E[(X - \mu_X)^2]$ is the variance.

The fact that the first-order moment is 0 can be generalized to all odd orders for distributions that are symmetric about the mean:

Lemma 4.2 *All central moments of odd orders are 0 for symmetric distributions.*

Proof Sketch: This result can be shown easily by breaking up the definite integral $\int_{-\infty}^{\infty} (x - \mu_X)^k f_X(x) dx$ into two definite integrals — one integral has bounds extending from $-\infty$ to μ_X and the other has bounds extending from μ_X to ∞ . Then, symmetry can be used along with the fact that k is odd to show that this pair of integrals cancel each other out. ■

As a specific example, the normal distribution will always have odd central moments of 0 because it is symmetric about the mean. The sign of central moments of the third order tells us whether a distribution is left skewed or right skewed. A negative central moments of the third order is indicative of left skewness. On the other hand, a corresponding positive value is indicative of right skewness. For example, a random variable drawn from an exponential distribution will always have a third central moment that is positive. It is noteworthy that the notion of skewness should be considered heuristic (rather than one that defines right/left skewness), as asymmetric distributions do exist for which the third central moment is 0. Nevertheless, the sign of the third central moment provides enough insight that a standardized version is used to define a quantity called *skewness*. Therefore, we need to first define the concept of *standardized* moments:

Definition 4.4 (kth Standardized Moment) *Let X be a random variable with expected value of $\mu_X = E[X]$ and variance σ_X^2 . The k th standardized moment of X is the expected value of $([X - \mu_X]/\sigma_X)^k$. In other words, the k th standardized moment of X is $E([(X - \mu_X)/\sigma_X]^k)$.*

The standardized moment is different from the central moment only by a factor σ_X^k . Therefore, if a central moment is 0, the standardized moment will be 0 as well. The sign of the central moment is also inherited by the standardized moment as an indicator of skewness. However, since the random variable is standardized, the magnitudes of the moments provide further insights. In the case of $k = 2$, it can be shown that the second standardized moment is always 1. The third standardized moment tells us how skewed (i.e., asymmetric) the distribution is, whereas the fourth standardized moment tells us about the aggregate “fatness” of the tails of the distribution (on both sides). The skewness is defined as follows:

Definition 4.5 (Skewness) *The skewness of a random variable X is its third standardized moment.*

Left-skewed distributions usually have negative skewness values and right-skewed distributions usually have positive skewness values. It is noteworthy that the notion of left skewness and right skewness is an intuitive one, and no single measure exists to define it in a concrete way.

While skewness measures the asymmetric fatness of one tail with respect to the other, it is sometimes helpful to have a measure of the aggregate fatness of both tails. This is (heuristically) measured by the fourth standardized moment, which is referred to as the *kurtosis*.

Definition 4.6 (Kurtosis) *The kurtosis of a random variable X is its fourth standardized moment.*

Normal distributions can be shown to have a kurtosis of 3 (see Exercise 24), and therefore a value greater than 3 provides evidence that the tail is thicker than that of the normal distribution. For example, a t -distribution will always have kurtosis greater than 3. The kurtosis is not a pure indicator of tail fatness, because it also depends on the number and shape of peaks in a distribution. Measures such as skewness and kurtosis should be considered evidentiary rather than measures with provable qualities.

4.14.2 Moment Generating Functions

As we have already discussed, moments provide an alternative way to characterize the distribution of a random variable. However, computing multiple moments can be expensive as it requires the use of integral calculus. How can one capture an infinite number of moments in a concise function? It turns out that all functions can be represented as polynomials of potentially infinite degree with the use of the Taylor expansion [6]. Moment generating functions (MGFs) try to capture distributions as concise functions whose Taylor expansions contain the moments as coefficients. In order to understand this point, consider the random variable X with the following *s-transform* $f_X^T(s)$, which is an *algebraic function* of s :

$$f_X^T(s) = E[\exp(sX)]$$

The notation T in the superscript indicates that it is a transform rather than a probability density function. A useful property of the *s*-transform is that *its kth derivative at s = 0 provides the expected value of the kth moment* of the random variable, which is why it is referred to as the MGF:

$$E[X^k] = \left[\frac{d^k f_X^T(s)}{ds^k} \right]_{s=0}$$

In order to understand why the moment generating functions have the property that they do, consider the Maclaurin expansion of $\exp(sX)$:

$$\begin{aligned} f_X^T(s) &= E[\exp(sX)] = E\left[1 + Xs + \frac{X^2s^2}{2!} + \frac{X^3s^3}{3!} + \frac{X^4s^4}{4!} + \dots +\right] \\ &= 1 + E[X]s + \frac{E[X^2]s^2}{2!} + \frac{E[X^3]s^3}{3!} + \frac{E[X^4]s^4}{4!} + \dots + \end{aligned}$$

On differentiating the RHS of the above expression k times and setting $s = 0$, it is easy to show that the resulting expression is $E[X^k]$. For example, if we have Bernoulli random variable, the expression $\exp(sX)$ is $\exp(s)$ with probability p and 1 with probability $(1-p)$. Therefore, the moment generating function is as follows:

$$f_X^T(s) = p \exp(s) + (1 - p)$$

It is easy to show that the k th derivative of the above function at $s = 0$ is always p for $k > 0$. Note that the k th moment of the Bernoulli random variable is indeed p for $k > 0$. A useful result is that the moment generating function of the sum of independent random variables is a product of their moment generating functions.

Theorem 4.2 *Let X_1 and X_2 be two independent random variables. Then, the moment generating function of $(X_1 + X_2)$ is defined as follows:*

$$f_{X_1+X_2}^T(s) = f_{X_1}^T(s)f_{X_2}^T(s)$$

Note that a binomial random variable Y is the sum of n i.i.d. Bernoulli random variables X_1, \dots, X_n . Therefore, the moment generating function of the binomial distribution turns out to be the n th power of the moment generating function of the Bernoulli distribution:

$$f_Y^T(s) = \prod_{i=1}^n f_{X_i}^T(s) = [p \exp(s) + (1 - p)]^n$$

Another useful result is that the moment generating function of the affine function of a random variable can be obtained from the MGF of the original random variable:

Lemma 4.3 (MGF of Affine Function) Let X be a random variable with the moment generating function $f_X^T(s)$ and $Y = b + aX$. The moment generating function of Y is given by $f_Y^T(s) = \exp(bs)f_X^T(as)$.

The affine transform of the random variable can be used for computing central and standardized moments. This is helpful for efficiently computing quantities like skewness and kurtosis.

One can derive a wide variety of moment generating function of known distributions using some fairly simple tricks. It is noteworthy that higher order moments do not exist for some probability distributions. In such cases, the moment generating functions do not exist either. For example, higher order moments do not exist for the Student's t -distribution. As a result, the moment generating function does not exist either. In other cases, such as the χ^2 -distribution, the moment generating function may not exist for all s since it is possible for the exponential expectation $E[\exp(sX)]$ to be unbounded for some ranges of s . However, as long as it exists at $s = 0$ and its locality, its derivative at $s = 0$ can be used to compute the moments of the distribution. A list of moment generating functions for various standard distributions is provided in Table 4.2. We encourage the reader to derive each of these expressions on their own (by computing $E[\exp(sX)]$ in each case) in order to obtain a feel of the process of creating moment generating functions.

Since moment generating functions are alternative representations of probability distributions, a natural question arises as to whether a procedure exists for converting the MGF to the density function (just as a procedure exists for converting the density function to the MGF). This is achieved by observing that the *Fourier transform* of the density function is obtained by replacing s with $s\sqrt{-1} = is$ in the MGF. Therefore, the inverse Fourier transform may be used to recover the density function:

Theorem 4.3 Given the MGF $f_X^T(s)$ of a random variable X , its density function² may be recovered by using the following inverse transform, as long as $\int_{-\infty}^{\infty} |f_X^T(is)|ds$ exists:

$$f_X(x) = \frac{1}{2\pi} \int_{s=-\infty}^{\infty} \exp(-isx) f_X^T(is) ds$$

Table 4.2: Moment generating functions of some standard distributions

Distribution	Moment Generating Function
Uniform in $[a, b]$	$\frac{\exp(sb) - \exp(sa)}{s(b-a)}$ for $s \neq 0$ $f_X^T(s) = 0$ for $s = 0$
Bernoulli with probability p	$p\exp(s) + (1-p)$
Binomial with n trials of prob. p	$[p\exp(s) + (1-p)]^n$
Geometric with probability p	$\frac{p\exp(s)}{(1-(1-p)\exp(s))}$ for $s < -\ln(1-p)$
Exponential with arrival rate λ	$\frac{\lambda}{\lambda-s}$ for $s < \lambda$
Poisson with arrival rate λ	$\exp(\lambda[\exp(s) - 1])$
Normal: $\mathcal{N}(\mu, \sigma^2)$	$\exp(\mu s + \sigma^2 s^2 / 2)$
Student's t distribution	Does not exist
χ^2 with k degrees of freedom	$\frac{1}{(1-2s)^{k/2}}$ for $s < 1/2$

²A different inversion formula exists for PMFs, which is beyond the scope of this book.

It can be shown that the density function $f_X(x)$ must be continuous and bounded over $[-\infty, \infty]$ in order for $\int_{-\infty}^{\infty} |f_X^T(is)|ds$ to exist. For example, the above theorem cannot be used to derive the PDF of the exponential distribution from its MGF — the density function of the exponential distribution is discontinuous at $x = 0$. In many cases, *MGFs are hard to systematically invert to the PDF*. However, many ad hoc tricks exist for inversion, the most common of which is simple inspection. Although the inspection approach might seem rudimentary at first glance, it is surprisingly powerful — computational search methods can iterate over large databases of MGFs to find an appropriate match for a given MGF. In many cases, it is easier to derive the MGF of a function of random variables than the PDF. If this MGF evaluates to a known form, it can be used to derive the PDF as well (see Example 4.36).

Example 4.33 Show that the moment generating function of the geometric distribution with parameter p is $p \exp(s)/(1 - (1 - p)\exp(s))$.

Solution: The moment generating function $f_X^T(s)$ of the geometric random variable is given by the following:

$$f_X^T(s) = E[\exp(sX)] = \sum_{k=1}^{\infty} (1-p)^{k-1} p \exp(ks)$$

The infinite summation on the right-hand side is that of a geometric series in which the first term is $p \exp(s)$ and the geometric factor is $(1-p) \exp(s)$. Using the formula for the summation of an infinite geometric series, one obtains the following for the MGF:

$$f_X^T(s) = \frac{p \exp(s)}{(1 - (1-p)\exp(s))}$$

■

Example 4.34 Show that the MGF of the standard normal distribution is $\exp(s^2/2)$. Use this result to derive the MGF of the general normal distribution $\mathcal{N}(\mu, \sigma^2)$.

Solution: The s -transform of the standard normal distribution is as follows:

$$\begin{aligned} f_X^T(s) &= E[\exp(sX)] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp(sx) \exp(-x^2/2) dx \\ &= \exp(s^2/2) \underbrace{\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp(-[x-s]^2/2) dx}_1 \end{aligned}$$

The integral above is 1, because it integrates a normal distribution density function with mean s and standard deviation 1. Irrespective of the value of s , this integral will evaluate to 1.

The standard normal distribution is defined by the random variable $Y = \mu + \sigma X$, where $X \sim \mathcal{N}(0, 1)$. Since we already know that $f_X^T(s) = \exp(s^2/2)$ one can derive $f_Y^T(s)$ using Lemma 4.3 on affine transformations:

$$f_Y^T(s) = \exp(\mu s) f_X^T(\sigma s) = \exp(\mu s + \sigma^2 s^2/2)$$

■

Example 4.35 Use the MGF of the $\chi^2(k)$ distribution in Table 4.2 to evaluate its mean and variance.

Solution: Differentiating the MGF once as well as twice, one obtains the following:

$$\begin{aligned} E[X] &= \left[\frac{df_X^T(s)}{ds} \right]_{s=0} = \left[\frac{k}{(1-2s)^{(k+2)/2}} \right]_{s=0} = k \quad [\text{Mean}] \\ E[X^2] &= \left[\frac{d^2 f_X^T(s)}{ds^2} \right]_{s=0} = \left[\frac{k(k+2)}{(1-2s)^{(k+2)/2}} \right]_{s=0} = k(k+2) \end{aligned}$$

The variance is, therefore, given by the following:

$$\sigma_X^2 = E[X^2] - E[X]^2 = k(k+2) - k^2 = 2k$$

■

Example 4.36 Use the known MGF of the χ^2 -distribution to show that the square of a standard normal random variable belongs to the $\chi^2(1)$ distribution.

Solution: Let $Y = X^2$ be the square of the standard normal random variable X . Then, we have:

$$\begin{aligned} f_Y^T(s) &= E[\exp(sY)] = E[\exp(sX^2)] = \int_{-\infty}^{\infty} \exp(sx^2) f_X(x) dx \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp(sx^2) \exp(-x^2/2) dx = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp(-(1-2s)x^2/2) dx \end{aligned}$$

To evaluate the integral, we substitute $u = x\sqrt{1-2s}$. On making the substitution, one obtains the following:

$$f_Y^T(s) = \frac{1}{\sqrt{1-2s}} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp(-u^2) du = \frac{1}{\sqrt{1-2s}} \underbrace{\int_{-\infty}^{\infty} f_X(u) du}_{1} = \frac{1}{\sqrt{1-2s}}$$

Therefore, the MGF of the square of the normal random variable evaluates to the MGF of the $\chi^2(1)$ -distribution. This example shows the power of inspection when a function of a random variable has an MGF of a known form. Using computational methods that search over known MGFs can greatly increase the power and practicality of such an approach. ■

Problem 4.25 Show that the moment generating function of the exponential random variable with parameter λ is $\lambda/(\lambda-s)$. Use this result to show that the mean and variance of the exponential distribution are $1/\lambda$ and $1/\lambda^2$, respectively.

Problem 4.26 Show that the moment generating function of the Poisson distribution with parameter λ is $\exp(\lambda[\exp(s)-1])$. Use this result to show that the sum of two independent Poisson random variables with parameters λ_1 and λ_2 is also a Poisson variable with parameter $(\lambda_1 + \lambda_2)$.

4.15 Summary

This chapter introduces the common probability distributions that are used in machine learning. The notion of mixture-based distributions was introduced, which allows realistic modeling of observed data. The next chapter will discuss how some of these distributions can be used for hypothesis testing. Furthermore, Chapter 6 will discuss how one can work *backwards* from observed data to reconstruct probability distributions. This reconstruction approach provides the foundation on top of which much of probabilistic machine learning is built. Probability distributions can be alternately characterized with their moment generating functions, which can be used to derive various properties of the random variable.

4.16 Further Reading

An excellent discussion of various types of probability distributions is provided in [19]. The use of these probability distributions in engineering is discussed in [42]. Excellent discussions of moment generating function are provided in [13, 19].

4.17 Exercises

1. Find the mean and variance of a uniformly distributed random variable in [3, 7].
2. Suppose you have two independent uniformly distributed random variables in [0, 1]. What is the cumulative distribution function of each? What is the cumulative distribution function of the random variable defined by the minimum of the two variables? What is the probability density function of the minimum of the two variables?
3. Suppose that you interview job applicants until a candidate is successful. The interview of each candidate is independent and has probability 0.25 of success. How many candidates do you expect to interview in order to hire someone for the job?
4. If a fair die is thrown 50 times, what is the probability of obtaining four or more sixes?
5. If you toss a fair coin 16 times, what is the standard deviation of the number of heads? How many standard deviations from the expected number of heads is the outcome of getting two heads? What is the probability of getting two or fewer heads?
6. Find the median of the number of sixes on throwing a die 10 times. You may use Table 3.2 to help you with the calculations.
7. Suppose you roll a fair die six times, what is the probability that you obtain a 1 or 2 three times, and obtain a 4 three times.
8. The probability of a face in a biased die is proportional to its value. You pick between the biased dice and a fair die with equal probability and roll it six times. What is the probability that you obtain a 1 or 2 three times, and obtain a 4 three times?
9. How would your answer to Exercise 8 change if you perform the die selection before each of the six rolls rather than only once?
10. The time between successive bus arrivals at a stop follows an exponential distribution with arrival rate $\lambda = 4$ per hour. You arrived at the bus-stop 5 minutes ago and no bus has arrived yet. What is your expected waiting time (including the elapsed time)?

- 11.** Let T be an exponential random variable with arrival rate λ . Use integral calculus to show that $E[T^3] = 3!/\lambda^3$ and $E[T^4] = 4!/\lambda^4$.
- 12.** Let T be an exponential random variable with arrival rate λ and let $V = T^2$ be a derivative random variable. Derive an expression for the variance of V .
- 13.** Compute the mean, variance, and second-moment of a Poisson random variable with inter-arrival rate 5. What is the mode of this Poisson variable?
- 14.** A call center receives calls continuously, so that the time between successive calls follows an exponential distribution with inter-arrival rate 5 per hour. What is the mean and variance of the number of calls the center receives in half an hour? What is the most likely number of calls the center receives in half an hour? What is the mean and variance of the length of time the center will take to receive 10 calls?
- 15.** Suppose that the birth-weight of babies is normally distributed with a mean of 7.5 pounds and standard deviation of 0.7 pounds. What fraction of babies are 9 pounds or more? What is the cut-off weight for the top-0.5% of babies in terms of birth weight?
- 16.** The Intelligence Quotient (IQ) of the human population is assumed to be normally distributed with a mean of 100 and a standard deviation of 15. What percentage of the human population has an IQ greater than 160? What is the IQ cut-off for the top-5% of the population?
- 17.** The height of females in a particular population is normally distributed with a mean of 63 inches. A group of four randomly selected females has an average height of 59 inches. Should this group of four females be considered among the 2.5 percent of shortest four-female groups, if (i) the population standard deviation of individual female heights is known to be 4 inches, and (ii) if the population standard deviation is not known but the sample standard deviation in this specific group is 4 inches?
- 18.** You have two coins, denoted by A and B. Coin A is fair, whereas coin B shows heads with probability 0.75. You select coin A with probability 0.8 (and select coin B, otherwise). You toss the selected coin 10 times and record the number of heads. Evaluate an expression for the probability mass function of the number of heads. If you see 7 heads out of the 10 tosses, what is the probability that coin A was selected?
- 19.** Buses to Yorktown arrive at stop A with an exponential inter-arrival time of rate 10 per hour. On the other hand, buses to Yorktown arrive at stop B with an exponential inter-arrival time of rate 6 per hour. Matthew waits at stop A 70% of the time and at stop B 30% of the time. Matthew calls you to complain that he had to wait at one of the stops for exactly an hour for his destination to Ossining but counted 9 buses to Yorktown in this duration. What is the probability that he was waiting at stop A?
- 20.** You start waiting for a bus when your stopwatch is set to 0 and the arrival time follows an exponential distribution with $\lambda = 2$ per hour. Find an expression for the probability that the arrival occurs in the one-second interval $(t, t + \delta)$, where the probability density is assumed to be constant over $(t, t + \delta)$.

Suppose that one hour has already passed ($t > 1$) and no bus has arrived. Derive an expression for the probability (conditional on $t > 1$) that the arrival occurs in $(t, t + \delta)$. Express this probability in terms of t' , where $t' = (t - 1)$ is measured from now (when one hour has passed and the stop-watch is showing 1 hour). Compare this expression

to that obtained in the first part of this problem where the wait had just started. Comment on this result in light of Example 4.12.

21. Use the MGF of the Bernoulli distribution to derive the MGF of the binomial distribution. Use this MGF to derive the mean and variance of the binomial distribution.
22. Use the MGF of the standard normal distribution to show that its n th moment is $n!/[(n/2)!2^{n/2}]$ for even n and 0, otherwise. [Hint: You will find it easier to work with the Maclaurin expansion of the MGF.]
23. Use Exercise 22 to derive the variance of the χ^2 -distribution with k degrees of freedom.
24. Use Exercise 22 to show that the kurtosis of any normal distribution is 3.
25. Use the MGF of the exponential distribution with parameter λ to calculate its third moment. Use this third moment along with its known mean and variance to show that the skewness of any exponential distribution is 2.
26. Use the PDF of the t -distribution to confirm the critical t -value in Figure 4.16 for upper-tail probability 0.05 and one degree of freedom.
27. Use the PDF of the χ^2 -distribution to confirm the upper-tail critical value in Figure 4.18 for probability 0.05 and two degrees of freedom.
28. Argue using the PDF of the t -distribution as to why it converges to the standard normal distribution when $\nu \rightarrow \infty$. Your job is to show that the t -distribution PDF converges to a function proportional to $\exp(-t^2/2)$.
29. The χ^2 -distribution is obtained by summing the squares of k base *standard* normal random variables. What is the PDF of a scaled χ^2 -distribution in which the base normal distributions all have mean 0 and variance σ^2 ? [A fun fact is that the square-root of this random variable at 3-degrees of freedom is the Maxwell-Boltzmann distribution, which is used to model the speeds of gas molecules in thermodynamics.]
30. Derive the median and the mode of the χ^2 -distribution at 4 degrees of freedom. While finding the median, you might encounter an equation that cannot be solved in closed form. Feel free to use any graphing calculator like Desmos to solve the equation.
31. Derive an expression for the mean absolute deviation (MAD) of the t -distribution in terms of the degrees of freedom ν . Show that it converges to the MAD value of $\sqrt{2/\pi}$ for the standard normal distribution (cf. Example 4.21) as $\nu \rightarrow \infty$.
32. Show that the sum of m independent random variables, each drawn from an exponential distribution with parameter $(1/2)$, follows a χ^2 distribution with $2m$ degrees of freedom. [Hint: Use MGF.]
33. A normal distribution $\mathcal{N}(\mu_0, \sigma_0^2)$ has known mean μ_0 and variance σ_0^2 . A value v is generated using m samples x_1, x_2, \dots, x_m from the normal distribution as follows:

$$v = \frac{\sum_{i=1}^m (x_i - \mu_0)^2}{m}$$

The sample v can be viewed as an instantiation of the random variable V . Express the PDF $f_V(v)$ in terms of μ_0 , σ_0 , and m .

- 34.** A normal distribution has an unknown mean and known variance σ_0^2 . A value v is generated from it using m samples x_1, x_2, \dots, x_m and their sample mean \bar{x} :

$$v = \frac{\sum_{i=1}^m (x_i - \bar{x})^2}{m - 1}$$

Derive an expression for the PDF $f_V(v)$ in terms of σ_0 and m .

- 35.** Use MGFs to show the closure property of the normal distribution under affine aggregation. In other words, if $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ for $i \in \{1, \dots, d\}$, show that $b + \sum_{i=1}^d a_i X_i$ is normally distributed. Use this property to show that the 1-dimensional projection of a multivariate Gaussian distribution along an *arbitrary* direction (and not just a principal component direction) is always normally distributed.
- 36.** Consider an origin-centered 2-dimensional Gaussian distribution. The variances in the X - and Y -directions are 1 and 2, respectively. The variance of the distribution along the projection [0.6, 0.8] is also 2. Compute the PDF of the distribution. [**Hint:** The marginal distributions along the X - and Y -directions are not independent.]

- 37. [Stirling's Approximation for the Binomial Distribution]:** The (refined) Stirling's approximation states that $n! \approx \sqrt{2\pi n}(n/e)^n$ and is passable (within 8%) even for $n = 1$. Let X be drawn from the binomial distribution with parameters n and p . Then, show the following for any fraction f so that $k = nf$ is an integer in $[1, n - 1]$:

$$P(X = k) \approx \frac{1}{\sqrt{2\pi nf(1-f)}} \left(\frac{p}{f}\right)^k \left(\frac{1-p}{1-f}\right)^{(n-k)}$$

What does the expression simplify to (in terms of n and p) when $p = f$?

- 38. [Stirling's Approximation for the Multinomial Distribution]:** Let $\vec{X} = [k_1, \dots, k_d]$ be drawn from the d -dimensional multinomial distribution with parameters n and p_1, p_2, \dots, p_d . Assume that each k_r is an integer in $[1, n - 1]$ and that $f_r = k_r/n$. Then, use the Stirling's approximation of Exercise 37 to show the following:

$$p_{\vec{X}}(k_1, k_2, \dots, k_d) \approx \frac{1}{\sqrt{(2\pi n)^{(d-1)} \prod_{r=1}^d f_r}} \prod_{r=1}^d \left(\frac{p_r}{f_r}\right)^{k_r}$$

What does the expression simplify to (in terms of n and each p_r) when each $p_r = f_r$?

- 39. [Stirling's Approximation for the Poisson Mode]:** Consider a random variable X drawn from a Poisson distribution with *positive integer* arrival rate λ . It is known that $X = \lambda$ is a mode of this distribution. Use the Stirling's approximation of Exercise 37 to show that the probability of this mode is given by the following:

$$P(X = \lambda) \approx \frac{1}{\sqrt{2\pi\lambda}}$$

- 40.** A loaded die is designed in such a way that faces 5 and 6 have probabilities of $1/12$ and $1/4$, respectively, whereas other faces each have probability $1/6$. A fair die is also available. One of the two dice is selected by flipping a fair coin and then thrown 20 times. The frequency vector of the six faces is $[3, 4, 2, 5, 2, 4]$. What is the probability that the fair die was selected?

- 41.** Consider a d -dimensional normally distributed data set in which the variances along the different independent directions (eigenvectors of the covariance matrix) are $\sigma_1^2, \dots, \sigma_d^2$. Two points are randomly sampled from the data set. Find the MGF of the distribution of the squared Euclidean distance between these points.
- 42.** Consider a 1-dimensional random variable X generated from a normal distribution with mean μ and standard deviation σ . All points outside $[a, b]$ are discarded and therefore a point is generated only if its lies inside $[a, b]$. Show that the conditional mean of the generated points is as follows:

$$E[X|X \in [a, b]] = \mu + \frac{\sigma}{\sqrt{2\pi}(F_X(b) - F_X(a))} \left(\exp\left(-\frac{(a-\mu)^2}{2\sigma^2}\right) - \exp\left(-\frac{(b-\mu)^2}{2\sigma^2}\right) \right)$$

Here, $F_X(x)$ is the cumulative distribution of the normal random variable X .

- 43.** The Erlang distribution with parameters k and λ is the sum of k i.i.d. exponential distributions with parameter λ . Use MGFs to show that the r th moment of the Erlang random variable is $(\prod_{p=0}^{r-1}(k+p))/\lambda^r$. The gamma distribution has the same PDF as the Erlang distribution except that the parameter k may be fractional? What statement can you make about the k th moment of the gamma distribution?
- 44.** Show that the MGF of a mixture of distributions defined by density functions $f_{X_1}(x) \dots f_{X_k}(x)$ and prior probabilities $\alpha_1 \dots \alpha_k$ is given by $\sum_{i=1}^k \alpha_i f_{X_i}^T(s)$.
- 45.** The Laplace random variable Z with parameters μ and β has the following PDF:

$$f_Z(z) = \frac{1}{2\beta} \exp(-\|z - \mu\|_1/\beta)$$

At $\mu = 0$, the Laplace distribution is essentially a double exponential distribution (i.e., a scaled exponential with parameter $1/\beta$ and its mirror image in the Y -axis). The two back-to-back exponential density functions are scaled down by a factor of 2 (in comparison with the single exponential) so that the PDF integrates to 1. Let X be an exponential random variable with parameter $1/\beta$. Argue without using integral calculus why the following relationship must hold between the MGFs of Z and X :

$$f_Z^T(s) = \frac{\exp(\mu s)}{2} (f_X^T(s) + f_X^T(-s))$$

Use this fact to show that $f_Z^T(s) = \exp(\mu s)/(1 - \beta^2 s^2)$.

- 46.** Let X be the random variable representing the number of throws of a fair dice needed to obtain 20 sixes. Derive the MGF of X by modifying one of the MGFs in Table 4.2.
- 47.** The time between successive emails follows an i.i.d. exponential distribution with parameter λ . The probability of each email containing a virus is p and is independent of that of other emails containing a virus. Show that the time between successive virus-infested emails follows an exponential distribution with parameter λp .
- 48.** Show the following mathematical identity for $n = q + r + s$ by using the fact that the sum of the probabilities of disjoint events is the probability of their union:

$$\frac{n!}{q!r!s!} = \frac{(n-1)!}{(q-1)!r!s!} + \frac{(n-1)!}{q!(r-1)!s!} + \frac{(n-1)!}{q!r!(s-1)!}$$



Chapter 5

Hypothesis Testing and Confidence Intervals

“Experience tells you what to do; confidence allows you to do it.” — Stan Smith

5.1 Introduction

The previous chapter introduced several probability distributions, including the normal distribution, the t -distribution, and the χ^2 -distribution. These distributions are very important in statistics because they enable the use of a very important concept in experimental science, referred to as *hypothesis testing*. This method is a formal technique for evaluating the reliability of a conclusion about the population from (limited) experimental data. The foundations of machine learning rely on making predictions from limited data. This chapter will introduce the notion of hypothesis testing, and it will also provide an understanding of how hypothesis testing can be very useful in machine learning applications.

Hypothesis testing arise in all types of scientific applications in which data from a sample is used to draw conclusions about the properties of the population. Since the sample may be of limited size, its properties may not always represent the population. Hypothesis testing provides an idea of the level of confidence one can have about a claim. A typical example of such a hypothesis might be a claim about a population parameter such as the mean, median, or standard deviation. Some examples of hypotheses about the population are as follows:

- “The mean salary of working US women is \$54,000.”
- “The median salary of working US women is \$52,000.”
- “The mean salary of working US men is the same as that of working US women.”

It is impossible to affirmatively prove any of the hypotheses discussed above. However, it is often possible to affirmatively disprove (i.e., *nullify*) these hypotheses with a particular

probability. This is the reason that the aforementioned hypothesis is sometimes referred to as a *null* hypothesis. For example, at the end of the hypothesis testing process, it is possible to arrive at one of the following two conclusions about the claim comparing the salary of US men and US women:

1. The mean salary of US working men is different from that of US working women with probability 99% (i.e., original hypothesis is nullified).
2. There is not enough evidence to say whether or not the mean salary of US working men is different from that of US working women with probability 99% (i.e., original hypothesis is not nullified).

It is noteworthy that the second statement above is not an affirmative claim, and therefore results in an undetermined conclusion in which the original hypothesis has neither been nullified or affirmed. In general hypothesis tests either make an affirmative assertion or result in an indeterminate conclusion.

Associated with hypothesis testing is the notion of *confidence intervals*. Such intervals try to quantify the range of variability caused by natural differences among different statistical samples. Confidence intervals answer questions such as the following:

- I surveyed a sample of 50 US women and found their average salary to be \$ 52,364. I know that my average will not be the same as the true population mean (because of sampling variations). What is the smallest range centered at \$ 52,364 in which I expect the true population mean to lie with probability 99%?
- My classifier gave me an accuracy of 71% on a sample of 50 test instances. I know that my accuracy is different from what I would get on the universe of all possible test instances. What is the smallest range in which the true test population accuracy lies with probability 99%?

In general, the approach is useful in quantifying the variability caused by limitations in sample sizes. Almost all experiments in different scientific domains face the challenges associated with the variability caused by limited sample size. Making affirmative conclusions with “confidence” lies at the heart of the principles of hypothesis testing. Conclusions that can be made with “confidence” are also referred to as *statistically significant*.

Hypothesis testing is useful in machine learning in cases where one performs experiments under two conditions in order to compare them. For example, consider a case where one has two classifiers (cf. section 1.6.2 of Chapter 1), and one of them has an accuracy of 60%, whereas the other has an accuracy of 80%. Can one truly say with confidence that one of the classifiers is better than the other in the *statistical* sense? After all, there might be random variations caused by the quirks of a small data set used for testing. Would the superior performance of one of the classifiers be replicated if a different test data set were used? Furthermore, confidence intervals can be used to quantify the range of accuracies of the two classifiers.

This chapter will introduce hypothesis testing methods that evaluate the properties of individual populations or compare the properties of two populations. For example, while talking about individual populations, one might test whether a population mean/standard-deviation is at a particular value or greater than a particular value. While testing the properties of two populations, one might determine whether one population mean is greater than the other (with sufficient *significance*). Hypothesis testing is also used to determine how well the data fits a particular distribution or whether two attributes are independent of one another.

5.1.1 Chapter Organization

This chapter is organized as follows. The next section will revisit the central limit theorem, which enables the use of the normal distribution for hypothesis testing. The notion of sampling distributions and standard errors is provided in section 5.3. The basics of hypothesis testing are introduced in section 5.4. Hypothesis tests for differences in means are discussed in section 5.5. Hypothesis tests that use the χ^2 distribution are introduced in section 5.6. The discussion of the analysis of variance is provided in section 5.7. Machine learning applications of hypothesis testing are discussed in section 5.8. A summary is given in section 5.9.

5.2 The Central Limit Theorem

Before discussing the central limit theorem, we will introduce a simpler result, referred to as the *law of large numbers*. The law of large numbers simply states that the mean of a large number of i.i.d. samples converges to the mean of the population:

Theorem 5.1 (Law of Large Numbers) *Let X_1, X_2, \dots, X_n be n independent and identically distributed random variables with mean μ . Let the random variable \bar{X}_n be defined as the mean of these variables:*

$$\bar{X}_n = \frac{\sum_{i=1}^n X_i}{n}$$

Then, as n goes to ∞ , \bar{X}_n converges to the population mean with probability 1.

Note that the above result does not say anything about the distribution of \bar{X}_n , which is addressed by the central limit theorem. The central limit theorem states that the mean of n independent and identically distributed variables approaches a normal distribution centered at $E[X_i]$ and variance equal to $\sigma_{X_i}^2/n$. Therefore, the central limit theorem is a generalization of the law of large numbers.

In order to understand the central limit theorem, we will revisit an example used to introduce the normal distribution in Chapter 4. Consider the case where one starts with an exponential random variable with arrival rate $\lambda = 1$. This distribution is shown in Figure 5.1(a). The shape of this distribution is quite asymmetric, and it looks very different from the familiar bell-shaped curve of a normal distribution. Figures 5.1(b), (c), (d), (e), and (f) show the probability distributions of the random variable $Y_n = \sum_{i=1}^n X_i$ for varying values of n . It is evident that as n increases, the sum of these n highly skewed random variables becomes increasingly symmetric and starts resembling a bell-shaped curve. This bell-shaped curve can be shown to converge to the normal distribution with increasing values of n according to the central limit theorem. This property holds true whether we use the *sum* or the *mean* of random variables, since the two differ by only a scaling factor.

Theorem 5.2 (Central Limit Theorem) *Let X_1, X_2, \dots, X_n be n independent and identically distributed random variables with mean μ and variance σ^2 . Let the random variable \bar{X}_n be defined as the mean of these variables:*

$$\bar{X}_n = \frac{\sum_{i=1}^n X_i}{n}$$

Then, as n goes to ∞ , \bar{X}_n converges to the normal distribution with mean μ and variance σ^2/n .

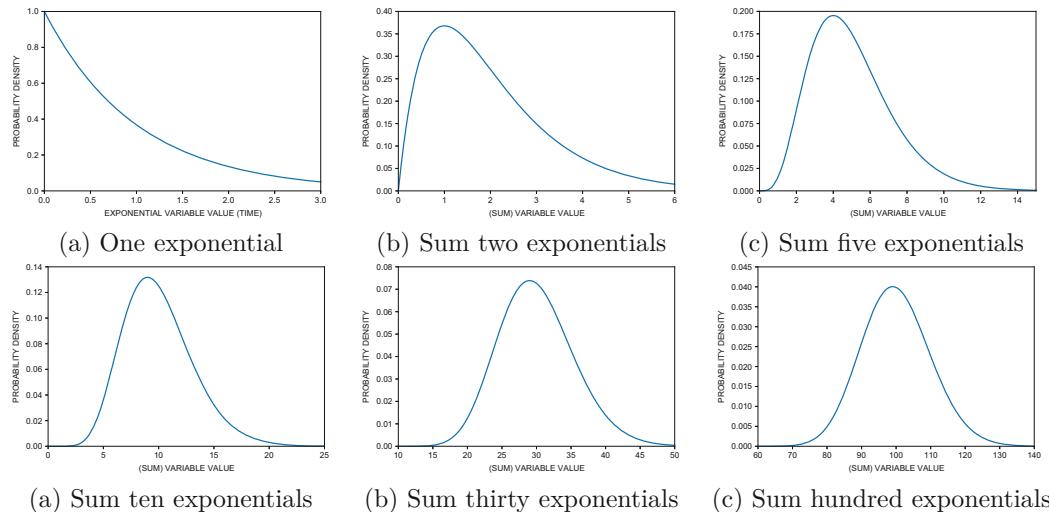


Figure 5.1: Revisiting Figure 4.8: The distributions obtained by adding multiple exponential random variables. Note that as more exponentials are added, the asymmetry of the distribution disappears and the probability density function starts resembling a bell-shaped curve approximating the normal distribution.

A proof of this result is omitted. The basic idea behind the proof is to create the logarithm of the moment generating function (cf. section 4.14.2 of Chapter 4) of \bar{X}_n , and examine its limit as $n \rightarrow \infty$.

Although the central limit theorem discusses theoretical convergence for $n \rightarrow \infty$, it is also helpful to understand its practical convergence behavior — how large does n need to be *in practical settings* in order for the distribution of \bar{X}_n to be sufficiently close to that of a normal distribution? The answer depends on the nature of the base distribution from which the i.i.d. samples are drawn. For example, the sum of just two i.i.d. normal distributions is also a normal distribution. Similarly, distributions that are highly symmetric and unimodal tend to converge quickly to a normal distribution. Asymmetric distributions like the exponential distribution are slower to converge to the normal distribution. Even in the case of the highly asymmetric exponential distribution, it is evident from Figure 5.1 that the distribution starts looking awfully like a normal distribution for sample sizes greater than 30. This is good news because it enables the use of a normal distribution assumption while testing hypotheses from a modest number of samples.

5.3 Sampling Distribution and Standard Error

The central limit theorem establishes that the mean \bar{X}_n of i.i.d. variables X_1, X_2, \dots, X_n converges to the normal distribution. The distribution of the mean over repeated samples is referred to as the *sampling distribution*. This sampling distribution converges to the normal distribution, which is expressed as follows:

$$\bar{X}_n \sim \mathcal{N} \left(\mu, \frac{\sigma^2}{n} \right)$$

Table 5.1: Examples of different sample statistics and sampling distributions

Distribution of X_i	Sample Statistic T	Sampling Distribution of T
$X_i \sim \mathcal{N}(\mu_0, \sigma_0^2)$	$T = \bar{X}_n$	$T \sim \mathcal{N}(\mu_0, \sigma_0^2/n)$
$X_i \sim \mathcal{N}(\mu_0, \sigma_0^2)$	$T = (\bar{X}_n - \mu_0)/\sigma_0$	$T \sim \mathcal{N}(0, 1)$
$X_i \sim \text{Bernoulli}(p_0)$	$T = \sum_{i=1}^n X_i$	$T \sim \text{B}(n, p_0)$
$X_i \sim \mathcal{N}(\mu_0, \sigma_0^2)$	$T = \sum_{i=1}^n (X_i - \mu_0)^2 / \sigma_0^2$	$T \sim \chi^2(n)$
$X_i \sim \mathcal{N}(\mu_0, \sigma_0^2)$	$T = \sum_{i=1}^n (X_i - \bar{X}_n)^2 / \sigma_0^2$	$T \sim \chi^2(n - 1)$

Since the variance of the sampling distribution reduces with increasing n , one can often make more definitive conclusions with increasing sample sizes.

It is possible to create more accurate sampling distributions (than the normal distribution implied by the central limit approximation) if one has some insight about the population distribution. For example, the sample mean of a set of Bernoulli samples can be shown to be a scaled version of a binomial distribution (which converges to the normal distribution for large n). Unfortunately, since the population distribution is rarely known, the normal distribution is used as an approximation for the sampling distribution.

The sample mean is an example of a *sample statistic*. A sample statistic is defined as follows:

Definition 5.1 (Sample Statistic) *A sample statistic is an estimate of some statistic of a sample of size n from a population.*

The sample statistic varies from one sample to another, which causes uncertainty about the corresponding property of the population. Hypothesis testing attempts to make claims about various properties of the population, and these properties are typically estimated in the real world using available observations (samples) of finite size. Some examples of sample statistics could be the mean, median, or the maximum value of a set of observations. The learned coefficient(s) of regression modeling from a sample can also be considered a sample statistic. Although only one estimate of these values may be available from a fixed sample of observations, one often models the theoretical distribution of the sample statistic with the notion of a sampling distribution:

Definition 5.2 (Sampling Distribution) *The sampling distribution is the probability distribution of a sample statistic from a sample of a particular size n .*

Depending on the type of sample statistic on which hypothesis tests are applied, one might obtain different types of sampling distributions. The sampling distribution depends on both the distribution of X_i and the nature of the sample statistic. Some examples of different sample statistics with their sampling distributions are illustrated in Table 5.1.

Hypothesis testing uses the probabilities of the tail areas of a distribution to decide whether to reject or not reject claims about the properties (e.g., mean) of the population. For example, if the sample mean lies in a small tail area of a distribution constructed using a hypothesized mean, it is likely that the hypothesis about the mean was incorrect in the first place. For certain types of sampling distributions like the normal distribution, its mean and standard deviation are sufficient to model these tail probabilities. The standard deviation of the sampling distribution of the sample statistic is referred to as the *standard error*:

Definition 5.3 (Standard Error) *The standard error is a measure of the dispersion of the sampling distribution of a sample statistic. It is equal to the standard deviation of the sampling distribution.*

The standard error plays a critical role in the machinery of hypothesis testing, especially when the sampling distributions are normal or drawn from a t -distribution. This is because the tail probabilities of such distributions can be quantified using the only the mean and the standard error. However, for other sampling distributions like the χ^2 -distribution, the standard error is not as helpful because it does not directly provide tail probabilities.

Hypothesis testing can be done for any statistic estimated with sampling (like the median or even a regression coefficient). The main issue is that sampling distributions are not always easy to construct for arbitrary sample statistics. This chapter will primarily focus on the sample mean, but some examples of other sample statistics (e.g., sample variance) will also be explored. The reason for the popularity of the sample mean (as the sample statistic) in hypothesis testing is the relative ease in computing its tail probabilities by using the normal distribution. Other sample statistics may not show these convenient characteristics. For example, the sampling distribution of the median converges to the normal distribution only under stricter conditions on the smoothness of the density function.

5.4 The Basics of Hypothesis Testing

In this section, the simplest case of hypothesis testing with the population mean is introduced. A standardized *test statistic* is constructed using the sample mean and the standard error. Furthermore, it will be assumed that the population standard deviation is known *a priori* in order to calculate standard errors. As we will see later, this assumption may not always hold in practical settings, and one is sometimes forced to use the sample standard deviation instead. In such a case, the sampling distribution changes to the t -distribution.

The general framework of hypothesis testing relies on a *null hypothesis* and an *alternate hypothesis*. The null hypothesis always makes a claim about the property of a population or a function of the properties of multiple populations. An example of the basic form of the null hypothesis is as follows:

The population mean μ of US working women salary is equal to μ_0 .

Here, μ_0 is a constant value. For example, if one makes the hypothesis that the average annual salary of working US women is \$50,000, then the value of μ_0 is simply 50,000 and this number is explicitly plugged into the null hypothesis instead of the symbol μ_0 . It is common to formally state the null hypothesis using the notation H_0 as follows:

$$H_0 : \mu = \mu_0$$

Note that the null hypothesis is the statement that we are trying to disprove (or *nullify*). Therefore, an alternative hypothesis must be available, which holds when the null hypothesis is untrue. This alternative hypothesis is denoted by H_1 . The natural form of the alternate hypothesis in this case is as follows:

The population mean μ of US working women salary is not equal to μ_0 .

One can also state the alternate hypothesis in formal notation as follows:

$$H_1 : \mu \neq \mu_0$$

It is noteworthy that the null hypothesis is always denoted by H_0 and the alternate hypothesis is denoted by H_1 . This particular form of the hypothesis test is referred to as a *two-tailed* or *two-sided* test, because the null hypothesis can be disproven on either side of

μ_0 . Since the alternate hypothesis does not make any specific claim on whether μ is less or greater than μ_0 , the null hypothesis can be disproven as long as the sample mean lies far away from the hypothesized population mean μ_0 . There are other forms of the alternate hypothesis such as $\mu > \mu_0$ or $\mu < \mu_0$. This type of alternate hypothesis creates a *one-tailed test*, because the null hypothesis can be disproven only when the sample mean lies in the upper tail of the distribution. Now consider the case where one has been given n samples x_1, \dots, x_n from the population (e.g., salaries of working US women), and one wants to determine whether the salary is sufficiently different from μ_0 . In such a case, the central limit theorem is used to claim that the random variable representing the sample mean \bar{X} has a sampling distribution that is normally distributed with mean μ_0 and variance σ^2/n :

$$\bar{X} \sim \mathcal{N}\left(\mu_0, \frac{\sigma^2}{n}\right)$$

Here, it is important to understand that even though the population mean might not be truly μ_0 , anything posited by the null hypothesis is assumed to be true for the purpose of testing (and possible nullification); therefore, the goal of hypothesis testing is to try to nullify this hypothesis by showing that the sample mean lies in the tail(s) consistent with the alternate hypothesis. Failing to nullify the hypothesis results in an indeterminate conclusion, wherein the differences in posited mean and sample mean *might* be explained by random variations. This is the reason that one does not “accept” the null hypothesis if one is unable to reject it.

It is common to create the test statistic as a sample from the standardized version of the sample statistic \bar{X} . The standardized form of the sample mean \bar{X} is as follows:

$$Z = \frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}}$$

Here, σ/\sqrt{n} is the standard error of the sample statistic \bar{X} . If the null hypothesis were true, the test statistic Z would have an expected value of 0 and variance of 1. Therefore, it is easy to see that the following holds true:

$$Z \sim \mathcal{N}(0, 1)$$

The aforementioned notations in upper case correspond to random variables. The realized sample x_1, x_2, \dots, x_n is denoted in lower case, with a corresponding (realized) sample mean:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

The realized test statistic z is then computed as follows:

$$z = \frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}}$$

If the null hypothesis were true, the absolute value $|z|$ will be small, and the test statistic z would lie in central regions of the standard normal distribution. Conversely, for a large test statistic that lies in the tail, it becomes more likely that the null hypothesis ought to be rejected. This probability is referred to as the *p-value*, which is defined as follows:

Definition 5.4 (p-Value) *The p-value is the probability of observing a test statistic at least as extreme as the one realized from the given sample, if the null hypothesis were true:*

$$p = P(Z > |z|) + P(Z < -|z|)$$

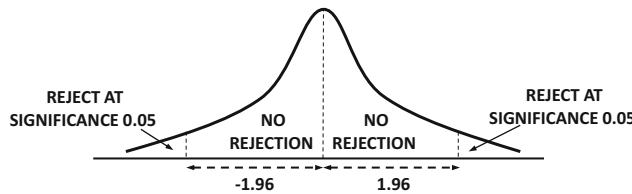


Figure 5.2: The tail rejection areas on the standard normal distribution at significance level $\alpha = 0.05$ for the two-tailed test. If the normal curve were not standardized, the tail rejection areas would be beyond 1.96 standard errors on either side of the mean.

When the null hypothesis is true, the p -value will be large (although a large p -value does not necessarily mean that the null hypothesis is true). On the other hand, when the null hypothesis is false, the p -value will be small (and the converse is true as well). A common cut-off used to reject the null hypothesis is a p -value less than 0.05. What is the (absolute) cut-off value $z_0 > 0$ of the test-statistic for this p -value of 0.05? In other words, we must have the following constraint on z_0 when Z belongs to the standard normal distribution:

$$P(Z > z_0) + P(Z < -z_0) = 0.05$$

Because of the symmetric nature of the standard normal distribution, it is known that $P(Z > z_0) = P(Z < -z_0)$. By substituting for $P(Z < -z_0)$ in the above expression, one obtains the following:

$$P(Z > z_0) = 0.025$$

In other words, 2.5% of the standard normal distribution must be included in the upper-tail greater than z_0 . By using the normal distribution tables (cf. pp 156), it can be shown that this cut-off value z_0 is 1.96. Therefore, *if the calculated test-statistic is greater than 1.96 in absolute magnitude, one can reject the null hypothesis with 95% probability*. In such a case, the hypothesis has been “nullified” with 95% confidence, which provides the semantic origins of the word “null hypothesis.” On the other hand, if the test statistic is less than 1.96, it does not mean that the null hypothesis is accepted — it simply means that the results of the hypothesis test are inconclusive, and there is not enough evidence to reject the null hypothesis with 95% probability. This is because the true mean might still be different from μ_0 but it is not reflected in the test statistic at the current sample size. One might need larger sample sizes to identify such cases.

The two terminologies used to refer to the probability value associated with hypothesis rejection are the *confidence level* and the *significance level*. The confidence level refers to the percentage of non-tail area in which the test statistic must occur (for the null hypothesis to not be rejected), whereas the significance level refers to the fraction of tail area in which the test statistic must occur in order for the null hypothesis to be rejected. Therefore, if the confidence level is 95%, the significance level is 0.05. While the confidence level is expressed as a percentage, the significance level is expressed as a fraction. The significance level is often symbolically denoted by α . A pictorial illustration of the rejection area of the null hypothesis in the standard normal distribution at $\alpha = 0.05$ is shown in Figure 5.2.

Although the p -value cut-off of 0.05 (i.e., 0.05 significance level) is the most common cut-off used in hypothesis testing, a variety of other cut-off values are used. The most common cut-off values for the p -value (significance levels) are 0.1, 0.05, and 0.01, corresponding to confidence levels of 90%, 95%, and 99%, respectively. In such cases, one is looking for the

Table 5.2: The significance level α with corresponding test-statistic cutoff for two-sided hypothesis tests. The corresponding test statistic values are referred to as *critical* values.

Significance level α	Confidence Level	Test-statistic cut-off
0.1	90%	1.65
0.05	95%	1.96
0.01	99%	2.58
0.0026	99.74%	3

(absolute) test-statistic to be greater than 1.65, 1.96, and 2.58, respectively. In addition, a test-statistic value of 3 is always assumed to be significantly different (corresponding to significance level $\alpha = 0.0026$), and one generally does not use significance levels smaller than this value. These different cut-offs are shown in Table 5.2. It is noteworthy that these cut-offs correspond to *two-tailed* (also referred to as *two-sided*) confidence tests, where rejection of the null hypothesis may happen because of the presence of the test statistic in either tail of the normal distribution. The cut-off values of the test statistic are also referred to as *critical test statistic values* for the case of the normal distribution.

Example 5.1 Consider a standardized test for which the population scores are known to have a standard deviation of 100. A sample of 100 students was found to have an average score of 531. Consider a null hypothesis that the population mean of the standardized test is 510. Would this hypothesis be rejected by a two-tailed hypothesis test at the $\alpha = 0.05$ significance level? How about the $\alpha = 0.01$ significance level?

Solution: The standard error of the sample mean is given by $\sigma/\sqrt{n} = 100/10 = 10$. Therefore, the test statistic z is as follows:

$$z = \frac{531 - 510}{10} = 2.1$$

This test statistic is greater (in absolute magnitude) than the critical value of 1.96 at $\alpha = 0.05$ but it is less than the critical value of 2.58 at $\alpha = 0.01$. Therefore, the hypothesis is rejected at the $\alpha = 0.05$ significance level but not rejected at the $\alpha = 0.01$ significance level. ■

Example 5.2 (Sample Size Effects) Babies in a particular hospital are known to have an average birth-weight of 7.1 pounds and a standard deviation of 1 pound. A drug for a particular condition in pregnant mothers is put through a clinical trial by the hospital to make sure that it does not have any side-effect on baby weight. A clinical trial with 100 mothers was found to result in an average birth-weight of 7.0 pounds. Does this clinical trial show a statistically significant effect of the drug on birth weight? A second clinical trial with 10000 mothers was found to have an average birth-weight of 7.05 pounds (which is closer to the known population mean of baby birth-weights). Repeat the statistical test at the $\alpha = 0.05$ significance level and discuss what you can learn from your two results.

Solution: The null hypothesis is that the baby-birth weight continues to stay at 7.1 pounds on taking the drug. The alternate hypothesis is that the baby birth-weight is different from 7.1 pounds on taking the drug.

The standard error of the sample mean in the first clinical trial with 100 babies is given by $\sigma/\sqrt{n} = 1/10 = 0.1$ pounds. Therefore, the test statistic z is as follows:

$$z = \frac{7.0 - 7.1}{0.1} = -1$$

This test statistic is not greater (in absolute magnitude) than the critical value of 1.96 at $\alpha = 0.05$. Therefore, the null hypothesis cannot be rejected — in other words, there is not enough evidence to know whether the effect of the drug on the birth-weight is statistically significant at the $\alpha = 0.05$ level.

The standard error of the sample mean in the second clinical trial with 10000 babies is given by $\sigma/\sqrt{n} = 1/100 = 0.01$ pounds. Therefore, the test statistic z is as follows:

$$z = \frac{7.05 - 7.1}{0.01} = -5$$

This test statistic is indeed greater (in absolute magnitude) than the critical value of 1.96 at $\alpha = 0.05$. Therefore, the null hypothesis is rejected — the effect of the drug on the birth-weight is statistically significant at the $\alpha = 0.05$ level in the second clinical trial.

An apparent contradiction is that the second clinical trial has a birth weight closer to the population mean of baby weights (7.1 pounds) — yet, the effect of the drug on baby weight is considered statistically significant only in this case. The reason is that the increased amount of data in the second trial provides the evidence needed to reject the null hypothesis. Note that we were careful not to make an affirmative conclusion (in the first clinical trial) that the baby weights are unaffected by the drug. In other words, we made a claim of lack of evidence for rejection of the null hypothesis rather than accept it. There is a difference between accepting a hypothesis and not having enough evidence to reject it.

One weakness of hypothesis tests is that they tell you nothing about how *substantial* the difference of the sample mean might be from the hypothesized population mean. With sufficient data, even very minor differences will become statistically significant and most hypotheses can be eventually rejected. ■

Problem 5.1 Consider a factory manufacturing widgets in which the standard deviation of widget length is known to be 4.3 inches. A sample of 100 widgets is found to have a mean length of 101 inches. What is the p-value of a two-tailed test for the null hypothesis that the mean length of widgets produced by the factory is equal to 100 inches. Would the null hypothesis be rejected at the (i) $\alpha = 0.05$ significance level, and the (ii) $\alpha = 0.01$ significance level?

5.4.1 Confidence Intervals

Hypothesis tests try to accept or reject a conjecture about the population mean. In practice, the population mean is often unknown, and rather than hypothesize a specific value for it, one wants to use the sample to find a range of values in which it might lie. This range of values needs to be centered at the *sample* mean and the width should be chosen in such a way that the (unknown) population mean lies in it with a high probability (i.e., high *confidence*). Therefore, confidence intervals at level 95% have the following property:

Table 5.3: Confidence intervals at different confidence levels for sample $x_1 \dots x_n$ with realized mean \bar{x} and known population standard deviation σ

Significance level α	Confidence level	Realized confidence interval
0.1	90%	$[\bar{x} - 1.65\sigma/\sqrt{n}, \bar{x} + 1.65\sigma/\sqrt{n}]$
0.05	95%	$[\bar{x} - 1.96\sigma/\sqrt{n}, \bar{x} + 1.96\sigma/\sqrt{n}]$
0.01	99%	$[\bar{x} - 2.58\sigma/\sqrt{n}, \bar{x} + 2.58\sigma/\sqrt{n}]$
0.0036	99.74%	$[\bar{x} - 3\sigma/\sqrt{n}, \bar{x} + 3\sigma/\sqrt{n}]$

If the procedure of creating a 95%-confidence interval is repeated a large number of times with independently drawn samples, 95% of these different confidence intervals will contain the true population mean. In other words, the true population mean lies in a confidence interval with probability 0.95.

These types of intervals can be constructed at any level of confidence (and not just 95% confidence). First, note that if μ_0 is the population mean, the following must hold true using the critical thresholds for the $\alpha = 0.05$ significance level:

$$P\left(-1.96 \leq \frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}} \leq 1.96\right) = 0.95$$

The above equation can be algebraically manipulated to show the following (by multiplying each term with σ/\sqrt{n}):

$$P(-1.96\sigma/\sqrt{n} \leq \bar{X} - \mu_0 \leq 1.96\sigma/\sqrt{n}) = 0.95$$

Because of the symmetry of the above pair of inequalities, one can easily show the following:

$$P(-1.96\sigma/\sqrt{n} \leq \mu_0 - \bar{X} \leq 1.96\sigma/\sqrt{n}) = 0.95$$

One can then add the sample mean \bar{X} to each term to show the following:

$$P(\bar{X} - 1.96\sigma/\sqrt{n} \leq \mu_0 \leq \bar{X} + 1.96\sigma/\sqrt{n}) = 0.95$$

In other words, the true population mean μ_0 must lie in the range $[\bar{X} - 1.96\sigma/\sqrt{n}, \bar{X} + 1.96\sigma/\sqrt{n}]$ 95% of the time. Since the standard error SE is σ/\sqrt{n} , one can also say that the confidence interval extends to $\pm 1.96 SE$ of sample mean. The above analysis assumes that \bar{X} is a random variable, and therefore confidence intervals are also presented in the form of random variables that vary from sample to sample. For the specific sample x_1, x_2, \dots, x_n with mean \bar{x} , the realized confidence interval at the 95% level is $[\bar{x} - 1.96\sigma/\sqrt{n}, \bar{x} + 1.96\sigma/\sqrt{n}]$. One can also create similar confidence intervals at the 90% level or 99% level using multiplicative factors of 1.65 and 2.58, respectively. Furthermore, a multiplicative factor of 3 corresponds to the 99.74% confidence level, and is generally considered a safe range in which the population mean ought to lie in most practical settings. These different ranges of confidence intervals are summarized in Table 5.3. A few interesting insights can be gleaned from the confidence intervals in Table 5.3. First, the higher the level of desired confidence is, the wider the sample-mean-centered interval needs to be to account for the corner cases where the sample mean is not very representative of the population mean. Second, the width of the confidence interval reduces with \sqrt{n} . Therefore, a larger sample allows us to narrow the range in which the population mean lies with a specific probability.

Confidence intervals provide an alternative way to reject (or not reject) the null hypothesis that the population mean μ is μ_0 . This alternative rule is as follows:

The null hypothesis can be rejected at a specific confidence level when the corresponding confidence interval does not contain μ_0 . The null hypothesis cannot be rejected when the interval contains μ_0 .

Therefore, confidence intervals are more informative than hypothesis tests. They are also used more frequently in machine learning in order to quantify the accuracy range of an algorithm. For example, if a classification algorithm yields an accuracy of about 65% on a particular sample of test examples, a different sample might yield a somewhat different accuracy. The use of confidence intervals allows us to specify a range of accuracy values (e.g., [63%, 67%]) in which the classification accuracy will lie 95% of the time. Furthermore, using test instance sets of larger sizes allows us to narrow the accuracy range of the algorithm. The application of confidence intervals to the testing of machine learning algorithms is discussed in section 5.8.1.

Example 5.3 (Confidence Level Effects) *Compute the confidence intervals for the problem in Example 5.1 at the 95% and 99% confidence levels. Use these confidence intervals to evaluate the null hypothesis that the population mean is 555 should be rejected at corresponding significance levels. Compare the two confidence intervals.*

Solution: The solution to Example 5.1 has already computed the standard error to be 10. Based on the mean value of 531 for the sample, the confidence interval is $[531 - 1.96 * 10, 531 + 1.96 * 10]$. This interval evaluates to $[521.4, 550.6]$. Since the value of 555 does not lie in this interval, the null hypothesis can be rejected at the $\alpha = 0.05$ significance level.

The confidence interval at the 99% confidence level is $[531 - 2.58 * 10, 531 + 2.58 * 10] = [505.2, 556.8]$. Since the value of 555 lies in this interval, the null hypothesis cannot be rejected.

The confidence interval at the 99% confidence level is wider than the interval for the 96% confidence level in order to increase the probability of the population mean lying in the sample-mean-centered interval. ■

Example 5.4 (Sample Size Effects) *Find the 95% confidence intervals on baby birth-weights for the two clinical trials of Example 5.2. Compare the two confidence intervals.*

Solution: For the smaller clinical trial with 100 participants, the mean baby weight was 7.0 pounds and the standard error was 0.1 (calculated in the solution to Example 5.2). Therefore, the 95% confidence interval is $[7.0 - 1.96 * 0.1, 7.0 + 1.96 * 0.1] = [6.804, 7.196]$.

For the larger clinical trial with 10000 participants, the mean baby weight was 7.05 pounds and the standard error was 0.01 (calculated in the solution to Example 5.2). Therefore, the 95% confidence interval is $[7.05 - 1.96 * 0.01, 7.05 + 1.96 * 0.01] = [7.0304, 7.0696]$.

The second confidence interval is much narrower because the greater amount of data allows us to estimate the range of baby weights more accurately. ■

Problem 5.2 Consider a factory manufacturing widgets in which the standard deviation of widget length is known to be 4.3 inches. A sample of 100 widgets is found to have a mean

length of 101 inches. What is the confidence interval of the widget length at (i) the 95% confidence level, and (ii) the 99% confidence level?

Suppose you make the hypothesis that the mean widget length produced by the factory is 100 inches. Would the hypothesis be rejected at (i) $\alpha = 0.05$ significance level, and (ii) $\alpha = 0.01$ significance level? Answer the last part using only the confidence interval you calculated earlier (without also calculating the p-value).

5.4.2 When Population Standard Deviations Are Not Available

The aforementioned discussion makes the assumption that population standard deviations are available. This is a strong assumption in many practical scenarios. Population standard deviations are generally not available in the absence of specific domain knowledge. A natural approach in such cases is to replace the population standard deviation with the sample standard deviation in the calculation of the test statistic. Making this change, however, also changes the distribution of the test statistic because the sample standard deviation is itself a random variable that varies from sample to sample. Consider the following null and alternate hypotheses with respect to a conjectured population mean μ_0 :

$$\begin{aligned} H_0 : \mu &= \mu_0 \\ H_1 : \mu &\neq \mu_0 \end{aligned}$$

Recall that the random variable for the test-statistic with known population standard-deviation σ and hypothesized population mean μ_0 is as follows:

$$Z = \frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}}$$

Here, \bar{X} is the sample statistic and $SE = \sigma/\sqrt{n}$ is the standard error. This test statistic has a standard normal distribution. However, when the population standard deviation σ is not available, it is replaced with the random variable $\hat{\sigma}_X$ corresponding to the sample standard deviation:

$$\begin{aligned} T &= \frac{\bar{X} - \mu_0}{\hat{\sigma}_X/\sqrt{n}} \\ &= \frac{\bar{X} - \mu_0}{\sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n(n-1)}}} \end{aligned}$$

In this case, we have used the variable T to denote the test statistic, and it is no longer drawn from a standard normal distribution. Instead, it is drawn from a t -distribution with $(n - 1)$ degrees of freedom:

$$T \sim t(n - 1)$$

The t -distribution has thicker tails compared to the normal distribution in order to account for the additional uncertainty caused by estimating the standard deviation (from the specific sample) rather than using the known population value. One can denote the *realized* values of the random variables \bar{X} and T by using the same expressions introduced above in lower case. The realized mean of the sample x_1, x_2, \dots, x_n is computed as follows:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

Table 5.4: The critical test statistic values for two-tailed hypothesis tests when the standard deviation is estimated from the sample. Figure 4.16 can be used to compute a more comprehensive set of critical test statistics.

α	Confidence	t -value ($n = \infty$)	t -value ($n = 30$)	t -value ($n = 20$)	t -value ($n = 10$)	t -value ($n = 5$)
0.1	90%	1.65	1.70	1.73	1.83	2.13
0.05	95%	1.96	2.05	2.09	2.26	2.78
0.01	99%	2.58	2.76	2.86	3.25	4.6

The realized test statistic t is then computed as follows:

$$t = \frac{\bar{x} - \mu_0}{\sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n(n-1)}}}$$

The p -value is computed using exactly the same formula as in the case when population standard deviations are available:

$$p = P(T > |t|) + P(T < -|t|)$$

The main difference is that the t -distribution with $(n - 1)$ degrees of freedom needs to be used (instead of the standard normal distribution) in order to quantify the values of $P(T > t)$ and $P(T < -t)$. Larger thresholds on the test statistic are required to reach the same p -value in the t -distribution as compared to the standard normal distribution. The critical test statistic values for varying significance levels α and degrees of freedom are shown in Table 5.4. It is interesting to note that the threshold test statistic values for $n = 30$ samples are not hugely different from those of the normal distribution ($n = \infty$). In most machine learning settings, one has at least a few hundred samples, and therefore the use of the normal distribution is adequate.

One can create confidence intervals in a similar manner to the case when population standard deviations are available. For example, when population standard deviations are available, the 95% confidence interval is given by $[\bar{x} - 1.96\sigma/\sqrt{n}, \bar{x} + 1.96\sigma/\sqrt{n}]$. However, if population standard deviations are not available, then the sample standard deviation $\hat{\sigma}_X$ needs to be used instead with an appropriately modified t -value replacing 1.96. For example, when we have $n = 30$ samples (corresponding to 29 degrees of freedom), the appropriate t -value to be used is 2.05. Therefore, the confidence interval in this case is $[\bar{x} - 2.05\hat{\sigma}_X/\sqrt{n}, \bar{x} + 2.05\hat{\sigma}_X/\sqrt{n}]$. The interval is wider in this case because of the additional variability caused by estimation of the standard deviation from the sample. In general, if the significance level is α , and $t_{\alpha/2}$ is the critical t -value at which a fraction $\alpha/2$ of the area of the t -distribution is contained in the upper-tail beyond the critical t -value, then the confidence interval at the level of significance α is $[\bar{x} - t_{\alpha/2}\hat{\sigma}_X/\sqrt{n}, \bar{x} + t_{\alpha/2}\hat{\sigma}_X/\sqrt{n}]$.

Example 5.5 Consider a set of 16 students writing a standardized test in which the sample mean is 520. Would the hypothesis that the mean is 510 be rejected at the $\alpha = 0.05$ level of significance, if (a) the population standard deviation is known and is 20, or (b) the sample standard deviation is 20 but the population standard deviation is not known.

Solution: The standard error in both cases is $20/\sqrt{16} = 5$. Therefore, the test statistic in both cases is $(520 - 510)/5 = 2$. However, in case (a), the test statistic has a normal distribution, whereas in case (b), the test statistic has a t -distribution with 15 degrees of freedom. The critical z -value is 1.96 in case (a), whereas the critical t -value is 2.13 in the second case. Since the absolute value of the test statistic is 2 in each case, the null hypothesis can be rejected in the first case, whereas the null hypothesis cannot be rejected in the second case. ■

Example 5.6 Find the 95% confidence intervals on the birth weights for the smaller clinical trial (100 participants) of Example 5.2 under the modified assumption that the standard deviation of 1 pound on the birth weights is obtained from the sample. Compare this confidence interval to the one calculated when this standard deviation is a known population standard deviation.

Solution: According to Example 5.2, the sample mean is 7.0 pounds. The standard error is $1/\sqrt{100} = 0.1$ pounds. Since the sample size is 100, we are working with a t -distribution with 99 degrees of freedom. The critical t -value at the 95% confidence level is about 1.98. On using this critical value the confidence interval obtained is $[7.0 - 1.98 * 0.1, 7.0 + 1.98 * 0.1] = [6.802, 7.198]$. Repeating the same calculation under the assumption that the standard deviation is obtained from the population, the interval is $[7.0 - 1.96 * 0.1, 7.0 + 1.96 * 0.1] = [6.804, 7.196]$. Note that the two confidence intervals are almost the same for a sample size $n = 100$, which is much smaller than the typical sample sizes used in machine learning. Therefore, unless sample sizes are extremely small, the use of a standard normal distribution instead of a t -distribution suffices. ■

Problem 5.3 Consider a factory manufacturing widgets in which the standard deviation of widget length is not known. A sample of 30 widgets is found to have a mean length of 102 inches and standard deviation of 4.3 inches. What is the p -value of the hypothesis that the mean length of the widgets produced by the factory is 100 inches? The alternate hypothesis is that mean length of the produced widgets is different from 100 inches. Would the hypothesis be rejected at (i) $\alpha = 0.05$ significance level, and (ii) $\alpha = 0.01$ significance level?

Problem 5.4 Consider a factory manufacturing widgets in which the standard deviation of widget length is not known. A sample of 30 widgets is found to have a mean length of 102 inches and standard deviation of 4.3 inches. What is the confidence interval of the widget length at (i) the 95% confidence level, and (ii) the 99% confidence level?

Suppose you make the hypothesis that the mean widget length produced by the factory is 100 inches. Would the hypothesis be rejected at (i) the $\alpha = 0.05$ significance level, and (ii) $\alpha = 0.01$ significance level? Perform the hypothesis tests using only the confidence intervals you calculated in this problem (without using the p -values calculated in the previous problem).

5.4.3 The One-Tailed Hypothesis Test

All the tests discussed thus far are two-tailed tests in which the null hypothesis can be rejected in either tail of the normal distribution. As a result, when the null hypothesis is

formulated at a confidence level of 95%, the tail area of 5% is distributed equally between the two tails. In the one-tailed test, the alternative hypothesis is different, and focuses on either the upper tail or lower tail (in response to specific application-specific considerations). Therefore, the entire probability of 5% must be concentrated in a single tail, which changes the threshold value of the test statistic to a smaller (absolute) value. For example, the one-tailed test with respect to conjectured population mean μ_0 uses the following null and alternate hypotheses:

$$\begin{aligned}H_0 &: \mu = \mu_0 \\H_1 &: \mu > \mu_0\end{aligned}$$

An immediate observation on examining this pair of hypotheses is that they are not exhaustive — the case that $\mu < \mu_0$ is not covered. However, the null hypothesis can only be rejected when the test statistic is consistent with the alternate hypothesis $\mu > \mu_0$. In other words, when the hypotheses are not exhaustive, an affirmative conclusion such as nullification cannot occur in the absence of consistency with the alternate hypothesis. Therefore, this sort of hypothesis test is useful only in cases where the uncovered part of the hypothesis space is not important from an application-centric point of view. For example, consider the case in which one wishes to test the hypothesis that the average height of US women is equal to 67 inches, with the alternative hypothesis being that the average height of US women is greater than 67 inches. In such a case, rejecting the null hypothesis at the significance level $\alpha = 0.05$ corresponds to checking whether the test-statistic lies in 5% of the upper tail area of whatever distribution is used to model the test statistic. As in the case of the two-tailed test, the test statistic might be modeled with a standard normal distribution or a t -distribution depending on whether the test statistic is computed using the population standard deviation or the sample standard deviation. In the case that the standard normal distribution is used, the threshold test statistic (i.e., critical value) will be 1.65 instead of the two-tailed critical value of 1.96. Furthermore, the test statistic has to be positive for the rejection of the null hypothesis to occur, because the alternative hypothesis requires the presence of the test-statistic in the upper tail. Recall that test statistics are compared with thresholds in terms of *absolute values* in the case of two-tailed tests. In other words, if the test statistic is drawn from the standard normal distribution, the value of the test statistic has to be positive and greater than 1.65 for the null hypothesis to be rejected. Otherwise, the null hypothesis cannot be rejected. The corresponding tail rejection area is shown in Figure 5.3(a). Note that the entire half of the standard normal distribution less than the mean of 0 corresponds to a region where the null hypothesis cannot be rejected. It is instructive to compare this tail-rejection area to that of the two-tailed test in Figure 5.2.

A similar analysis applies to the case of the lower-tail test. In the lower-tail test, the null hypothesis and the alternate hypotheses are defined as follows:

$$\begin{aligned}H_0 &: \mu = \mu_0 \\H_1 &: \mu < \mu_0\end{aligned}$$

In this case, the null hypothesis is rejected when the test statistic is negative and less than a specific (negative) threshold. For example, if the test is performed at significance level $\alpha = 0.05$ and the test statistic follows a standard normal distribution, the value of the test statistic has to be less than (i.e., more negative than) -1.65. The tail rejection area is shown in Figure 5.3(b). One-tailed tests can be useful in some types of applications. For example, if one is trying to prove that a new classification algorithm is truly better than a particular



Figure 5.3: The tail rejection areas on the standard normal distribution curve at significance level $\alpha = 0.05$ for the one-tailed test.

accuracy value μ_0 set by business needs, it makes sense to use the one-tailed test for the upper tail (based on a hypothesized population mean of μ_0).

One can also create a *one-sided confidence interval* in such cases. For example, if the alternate hypothesis is $\mu > \mu_0$ (upper-tail test), the one-sided confidence interval has a *lower* bound corresponding to how *low* the true population mean might be. The upper bound is ∞ . This is because confidence intervals correspond to the ranges of the population mean where the null hypothesis is *not* rejected. For the upper-tail test, the null hypothesis is not rejected for any arbitrarily high value of the population mean, which is reflected in the fact that the upper bound on the confidence interval is ∞ .

For the upper-tail test, consider the case when the sample mean is \bar{x} , sample size is n , and population standard deviation is σ . In this case, the one-sided confidence interval is $[\bar{x} - 1.65\sigma/\sqrt{n}, \infty]$. On the other hand, in the lower-tail test, the one-sided confidence interval would be $[-\infty, \bar{x} + 1.65\sigma/\sqrt{n}]$. Note that the test-statistic threshold used to generate the confidence interval in this case is 1.65 instead of 1.96 (which was used in the two-sided case). The single tail area in the one-sided confidence interval has twice the area of *each* of the two tail areas in the two-sided confidence interval (at the same level of confidence).

Example 5.7 This exercise repeats Example 5.5 under the one-tailed condition. Consider a set of 16 students writing a standardized test in which the sample mean is 520. Would the hypothesis that the student population mean is 510 be rejected at the $\alpha = 0.05$ level of significance (with alternate hypothesis that the population mean is greater than 510), if (a) the population standard deviation is known and is 20, or (b) the sample standard deviation is 20 but the population standard deviation is not known. What are the one-sided confidence intervals?

Repeat the above problem for exactly the same numerical values, with the only difference that the sample mean is 500.

Solution: The standard error in both cases is $20/\sqrt{16} = 5$. Therefore, the test statistic in both cases is $(520 - 510)/5 = 2$. However, in case (a), the test statistic has a normal distribution, whereas in case (b), the test statistic has a *t*-distribution with 15 degrees of freedom. The critical *z*-value is 1.65 in case (a) for the one-tailed test, whereas the critical *t*-value is 1.75 in the case (b) for the one-tailed test. Note that these critical values are lower than the 1.96 value in the two-tailed test of Example 5.5 for the same sample mean of 520. Since the absolute value of the test statistic is 2 in each case, the null hypothesis can be rejected in both cases. This is different from

the solution to Example 5.5 in which the null hypothesis could be rejected only in case (a). The one-sided confidence interval for the case of known population mean is $[520 - 1.65 * 5, \infty] = [511.75, \infty]$. The one-sided confidence interval for the case of the unknown population mean is $[520 - 1.75 * 5, \infty] = [511.25, \infty]$. In both cases, the hypothesized population mean of 510 does not lie in the confidence interval. The one-sided confidence intervals provide an alternate way of establishing that the null hypotheses can be rejected in both cases.

If the sample mean were 500 instead of 520, the hypothesis cannot be rejected for both case (a) and case (b). This is because such a sample mean of 500 is not consistent with the alternate hypothesis that the population mean is greater than 510. The corresponding confidence intervals are $[491.75, \infty]$ and $[491.25, \infty]$, respectively. Since the hypothesized population mean lies inside these confidence intervals in both cases, both hypotheses cannot be rejected. ■

Problem 5.5 Consider a factory manufacturing widgets in which the standard deviation of widget length is known to be 4.3 inches. A sample of 100 widgets is found to have a mean length of 101 inches. Consider a hypothesis that the mean length of the widget population is 100 inches. The alternative hypothesis is that the mean length of the widget population is greater than 100 inches. Would this one-tailed hypothesis be rejected at (i) $\alpha = 0.05$ significance level, and (ii) $\alpha = 0.01$ significance level?

Problem 5.6 What is the one-sided confidence interval at the 95% confidence level for the preceding problem?

5.5 Hypothesis Tests For Differences in Means

Hypothesis tests for difference in means are useful for identifying whether the means of two populations are significantly different. This type of hypothesis test is very useful in scientific settings when comparing the results of two experiments. Such tests can be used to answer questions such as the following:

- Are the results of one experiment different from another in a statistically significant sense?
- Are the results of one procedure superior to another in a statistically significant sense?

For example, in a machine learning setting, one might want to test whether the performance of two classifiers is significantly different. There are different variations of this setting, depending on whether the two populations have similar variances or different variances. Furthermore, the paired t -test assumes that the two samples have the same number of items and a one-to-one correspondence exists between the items of the two samples. In the following discussion, all these different settings will be explored.

5.5.1 Unequal Variance t -Test

The unequal variance t -test is the most general case, which can always be used in any scenario. However, it is not optimal to do so because it can lead to uncertain conclusions even in cases where the hypothesis ought to be rejected. Therefore, it should be used only

in cases in which the two populations being compared have very different variances. The null hypothesis is that the two population means μ_1 and μ_2 are the same. The alternate hypothesis is that the two population means are not the same. These hypotheses are written in terms of difference in means as follows:

$$\begin{aligned} H_0 &: \mu_1 - \mu_2 = 0 \\ H_1 &: \mu_1 - \mu_2 \neq 0 \end{aligned}$$

This version of the alternate hypothesis corresponds to the symmetric two-tailed test. Note that these formats of the hypotheses have the same form as the two-tailed hypothesis tests discussed in the previous section, except that the test needs to be applied to the difference in sample means of the two populations, which are compared to the fixed value of 0. For simplicity, we will first consider the case where the sample standard deviations of the two populations are known and are equal to σ_1 and σ_2 , respectively. This will result in a test statistic belonging to the standard normal distribution rather than the t -distribution.

It is assumed that the random variables corresponding to the sample means of the two populations are \bar{X}_1 and \bar{X}_2 , respectively. The corresponding sample sizes are n_1 and n_2 , respectively. Then, the standard error of the difference in means is given by the following:

$$SE_\delta = \sqrt{\sigma_{\bar{X}_1}^2 + \sigma_{\bar{X}_2}^2} = \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$$

The aforementioned formula for deriving the standard-deviation of the difference in sample means is obtained using the results of Lemma 3.16, which shows that the variance of the sum of independent random variables \bar{X}_1 and $-\bar{X}_2$ is the sum of their variances. Then, the test-statistic Z may be computed as follows:

$$Z = \frac{(\bar{X}_1 - \bar{X}_2) - 0}{SE_\delta} = \frac{(\bar{X}_1 - \bar{X}_2) - 0}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

In the case that the population standard deviations are known, the test-statistic Z is approximately drawn from a standard normal distribution (since it is obtained by averaging and normalizing i.i.d. random variables and the central limit theorem applies):

$$Z \sim \mathcal{N}(0, 1)$$

The computation of the test statistic can also be written with realized values for a specific sample (in lower case):

$$z = \frac{(\bar{x}_1 - \bar{x}_2) - 0}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

The p -value of the null hypothesis that the two means are the same can be computed as follows:

$$p = P(Z > |z|) + P(Z < -|z|)$$

Because of the symmetry of the standard normal distribution, it is easy to see that the p -value is equal to $2P(Z > |z|)$. The null hypothesis is rejected at the 95% confidence level when the absolute value of the above test statistic is greater than 1.96 (or the p -value is less than 0.05). In that case, one can conclude that the two population means are not the same at a significance level of $\alpha = 0.05$.

In the event that the population standard deviation is not available, the standard error is computed using the sample standard deviation rather than the population standard deviation. Therefore, the realized value t of the test statistic T can be computed as follows:

$$t = \frac{(\bar{x}_1 - \bar{x}_2) - 0}{\frac{\hat{\sigma}_{x_1}^2}{n_1} + \frac{\hat{\sigma}_{x_2}^2}{n_2}}$$

Note the circumflexes on top of $\hat{\sigma}_{X_1}$ and $\hat{\sigma}_{X_2}$ to indicate that these are estimated values for samples of X_1 and X_2 . Since the standard deviations are estimated from the sample, the resulting test statistic has a t -distribution with $\min\{n_1 - 1, n_2 - 1\}$ degrees of freedom:

$$T \sim t(\min\{n_1 - 1, n_2 - 1\})$$

The corresponding p -value is given by $2P(T \geq |t|)$ for the case of the two-tailed test.

One can also create confidence intervals for the difference in means. In the case where the two population standard deviation are known, the 95% confidence interval on the difference $\bar{x}_1 - \bar{x}_2$ in sample means is $[\bar{x}_1 - \bar{x}_2 - 1.96 SE_\delta, \bar{x}_1 - \bar{x}_2 + 1.96 SE_\delta]$. In other words, the confidence interval for difference in means lies within ± 1.96 times the standard error of the sample difference in means. On the other hand, if the standard error is computed using the sample standard deviations (rather than the population standard deviation), the t -distribution must be used. The corresponding 95% confidence interval is $[\bar{x}_1 - \bar{x}_2 - t_{0.025} SE_\delta, \bar{x}_1 - \bar{x}_2 + t_{0.025} SE_\delta]$. The value $t_{0.025}$ corresponds to the value of t at which 2.5% of the area is contained in the upper tail of the t -distribution at $\min\{n_1 - 1, n_2 - 1\}$ degrees of freedom. An important observation on the confidence intervals on difference in sample means is that it is centered at $\bar{x}_1 - \bar{x}_2$, and therefore the confidence interval for $\bar{X}_1 - \bar{X}_2$ is typically different from that of $\bar{X}_2 - \bar{X}_1$. Therefore, the convention for ordering the two groups matters.

It is also possible to predict the results of the hypothesis test at a particular significance level by examining the corresponding confidence interval. The null hypothesis may be rejected when the confidence interval does not contain 0. In such a case, the entire range of values in the confidence interval is of the same sign. The null hypothesis is rejected in this case, because the notion of confidence intervals implies that if the experiment of creating pair-wise samples were to be repeated a large number of times, the difference in population means must lie in the confidence interval more than 95% of the time (for the hypothesis not to be rejected).

Finally, one-tailed hypothesis testing can be used in settings in which the alternative hypothesis allows the mean of one of the two populations to be larger (or smaller than the other). Therefore, the alternate hypothesis H_1 is $\mu_1 - \mu_2 > 0$ for the upper-tail test, and it is $\mu_1 - \mu_2 < 0$ for the lower-tail test. The basic principle of the hypothesis testing is the same as discussed in section 5.4.3. After computing the test statistic for difference in means, the p -value is $P(T > t)$ for the upper-tail test, and the p -value is $P(T < t)$ for the lower-tail test. One can compute the critical test-statistic values at significance α by using t_α (which is always positive) for the upper-tail and $t_{1-\alpha}$ (which is always negative) for the lower tail. These values are different from the critical value of $t_{\alpha/2}$ in the two-tailed case. The test statistic needs to be more positive than the critical t -value for upper-tail tests, and it needs to be more negative than the critical t -value for lower-tail tests. One-tailed hypothesis testing can be useful in settings of machine learning where it is desired to test whether the performance of one system is better than that of the other (rather than simply being different from the other).

Example 5.8 Consider two sets of students from New York and San Francisco, respectively, of sample sizes 5 and 25 who have taken standardized tests and have respective sample means 530 and 517.5. The sample standard deviations of the two sets of scores are 10 and 5, respectively. Consider the null hypothesis that the mean scores of New York and San Francisco students are the same. Can this hypothesis be rejected at the $\alpha = 0.05$ level of significance for the two-tailed test?

Solution: The standard error of the difference in means is as follows:

$$SE_{\delta} = \sqrt{\frac{10^2}{5} + \frac{5^2}{25}}$$

The value above for the standard error evaluates to $\sqrt{21}$. Since the difference in means is 12.5, the corresponding test statistic is $12.5/\sqrt{21} = 2.73$. The resulting test statistic has $(5 - 1)$ degrees of freedom, since the smaller sample size is 5. The critical t -value at 4 degrees of freedom at $\alpha = 0.05$ is 2.78 (for the two-tailed test). Since the test statistic is less than the critical t -value, the null hypothesis cannot be rejected. ■

Example 5.9 For the setting in Example 5.8, set up a two-sided 95% confidence interval for the difference in mean scores between New York and San Francisco. Use this confidence interval to set up a two-tailed hypothesis test evaluating the null hypothesis that the mean scores in New York and San Francisco are the same.

Solution: As evaluated in the previous example, the difference in means is 12.5 and the standard error is $\sqrt{21}$. Since the two-sided critical t -value at the 95% level of confidence is 2.78, it follows that the relevant confidence interval is $[12.5 - 2.78\sqrt{21}, 12.5 + 2.78\sqrt{21}] = [-0.24, 24.76]$. This interval represents the range in which the difference of mean scores between New York and San Francisco truly lies (at the population level) with 95% probability. Since the value of 0 is included in the confidence interval, the null hypothesis cannot be rejected. Therefore, there is not enough evidence to say that the mean scores of New York and San Francisco are different in a statistically significant way. ■

Problem 5.7 Suppose that two factories produce widgets that are hypothesized to have the same mean length (i.e., null hypothesis). The alternate hypothesis is that the mean lengths are not the same. Two samples of 30 widgets are drawn with mean lengths of 101 inches and 103 inches, respectively. Furthermore, both samples are of size 30 each and have sample standard deviations of 3 and 4, respectively. Is the null hypothesis rejected at the (i) $\alpha = 0.1$ significance level, (ii) $\alpha = 0.05$ significance level, and (iii) $\alpha = 0.01$ significance level?

Problem 5.8 For the preceding problem, create the confidence intervals at each of the confidence levels of 90%, 95%, and 99%. Use these confidence intervals in order to either accept or reject the null hypothesis at each significance level listed in the previous problem.

5.5.1.1 Tightening the Degrees of Freedom

Fewer degrees of freedom makes it harder to reject the null hypothesis and (thereby) have a conclusive result. This is because fewer degrees of freedom make the tails of the t -distribution thicker, as a result of which larger absolute values of the test statistic are required to reject the null hypothesis. More degrees of freedom are preferable. It turns out that the methodology used in the previous section for setting the number of degrees of freedom to $\min\{n_1 - 1, n_2 - 1\}$ is overly conservative. For the general difference in means test, the number of degrees of freedom can be tightened (i.e., increased) to the following:

$$df = \frac{(\hat{\sigma}_1^2/n_1 + \hat{\sigma}_2^2/n_2)^2}{[(\hat{\sigma}_1^2/n_1)^2/(n_1 - 1)] + [(\hat{\sigma}_2^2/n_2)^2/(n_2 - 1)]}$$

Here, $\hat{\sigma}_1$ and $\hat{\sigma}_2$ are the estimated standard deviation from the two samples, and the value is always larger than $\min\{n_1 - 1, n_2 - 1\}$. Furthermore, when the value evaluates to a non-integer, it means that the test-statistic has a sampling distribution somewhere between the sampling distributions of the t -distributions at the two surrounding degrees of freedom with integer values. In such a case, one can interpolate the critical t -value between the two surrounding integer degrees of freedom in a linear way and compare this critical value to the test statistic. An example is given below:

Example 5.10 Repeat the problem in Example 5.8, except that the degrees of freedom are computed using the tightened estimate.

Solution: As in Example 5.8, the test statistic is still 2.73. However, the degrees of freedom for the critical t -value are as follows:

$$df = \frac{(20 + 1)^2}{[20^2/4] + [1^2/24]} \approx 4.4$$

The critical t -value for (two-sided) $\alpha = 0.05$ at 4 degrees of freedom is 2.78, and the critical t -value at 5 degrees of freedom is 2.57. Linearly interpolating the critical t -value to 4.4, we obtain the following:

$$t^* = 2.78 * (1 - 0.4) + 2.57 * 0.4 = 2.70$$

In this case, the test statistic value of 2.73 is greater than the critical t -value of 2.70 and therefore the hypothesis can be rejected. This result is different from that obtained in Example 5.8, where the null hypothesis was not rejected. The use of the tightened degrees of freedom leads to more frequent rejections of the null hypothesis (i.e., more frequent conclusive results and fewer “not enough evidence” results). This is because the use of the tightened degrees of freedom leads to more efficient use of limited data. ■

Problem 5.9 Repeat Problems 5.7 and 5.8 using the tightened estimate of the number of degrees of freedom.

5.5.2 Equal Variance t -Test

The equal-variance t -test is applicable to cases in which the two populations are similar in terms of their variance. This is often the case in real-world settings because the populations are frequently drawn from the same domain and tend to have similar variance. The variances do not need to be exactly equal, as the condition of equality is treated in a heuristic way.

As in the case of the unequal-variance test, the two hypotheses are defined in a similar manner for population means μ_1 and μ_2 :

$$\begin{aligned} H_0 &: \mu_1 - \mu_2 = 0 \\ H_1 &: \mu_1 - \mu_2 \neq 0 \end{aligned}$$

Let \bar{x}_1 and \bar{x}_2 be the two sample means, with corresponding sample variances $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$. Let n_1 and n_2 be the sample sizes. Then, the test statistic is defined as follows:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{(n_1-1)\hat{\sigma}_1^2 + (n_2-1)\hat{\sigma}_2^2}{n_1+n_2-2} \left(\frac{1}{n_1} + \frac{1}{n_2} \right)}}$$

Note that this definition of the test statistic is different from the unequal variance test for difference in means. This is because the standard error is computed differently. The standard-error (i.e., denominator of the test statistic) can also be expressed by dividing the pooled sample standard-deviation by the square-root of half the harmonic mean \bar{n} of n_1 and n_2 . In other words, the denominator of the above equation can be expressed as follows:

$$SE_\delta = \frac{\hat{\sigma}_{pool}}{\sqrt{\bar{n}/2}}$$

Here, $\hat{\sigma}_{pool}$ and \bar{n} are defined as follows:

$$\begin{aligned} \hat{\sigma}_{pool}^2 &= \frac{(n_1-1)\hat{\sigma}_1^2 + (n_2-1)\hat{\sigma}_2^2}{n_1+n_2-2} \\ \frac{2}{\bar{n}} &= \frac{1}{n_1} + \frac{1}{n_2} \end{aligned}$$

The random variable T corresponding to the test-statistic t has a t -distribution with $(n_1 + n_2 - 2)$ degrees of freedom, which is larger than in the case of the unequal variance test. The larger number of degrees of freedom allows a hypothesis test in which the rejection of the null hypothesis becomes more likely because of a smaller tail. Therefore, if the two sample means are unequal, a definitive conclusion can be reached with less data. This is a desirable property because limitations on the number of observations represent the single biggest hindrance to robust and conclusive results in statistics. One can compute the p -value for the null hypothesis as follows:

$$p = P(T > |t|) + P(T < -|t|)$$

The null hypothesis is rejected if the p -value is less than the level of significance α . Alternatively, one can reject the null hypothesis when the computed test statistic is greater than the critical t value (in absolute magnitude) at the level of significance α . The critical t -values for a particular level of significance remain the same for all two-tailed tests with the t -distribution. The confidence intervals can also be created in a similar manner as the

unequal variance test, except that the standard error and the number of degrees of freedom are computed differently in this case. For example, the two-sided confidence interval at the 95% level is $[\bar{x}_1 - \bar{x}_2 - t_{0.025} SE_\delta, \bar{x}_1 - \bar{x}_2 + t_{0.025} SE_\delta]$.

Example 5.11 Consider two sets of students from New York and San Francisco, respectively, of sample sizes 5 and 25. The respective sample means of standardized test scores are 530 and 517.5. The sample standard deviations are 10 and 5, respectively. Consider the null hypothesis that the mean population scores of New York and San Francisco students are the same. Can this hypothesis be rejected at the $\alpha = 0.05$ level of significance for the two-tailed test? You may use the similarity assumption that the standard deviations of the two populations are similar (although the sample-level standard deviations of 10 and 5 are different).

Solution: This exercise repeats Example 5.8 under the similarity assumption. Therefore, the calculations are different as well. The harmonic mean \bar{n} of 5 and 25 is as follows:

$$\bar{n} = 2 * 5 * 25 / 30 = 25/3$$

The pooled standard deviation may be computed as follows:

$$\hat{\sigma}_{pool} = \sqrt{\frac{4 * 10^2 + 24 * 5^2}{5 + 25 - 2}} = \sqrt{250/7}$$

The standard error of the difference in means is as follows:

$$SE_\delta = \hat{\sigma}_{pool} / \sqrt{\bar{n}/2} = \sqrt{60/7} \approx 2.928$$

Since the difference in means is 12.5, the corresponding test statistic is $12.5 / 2.928 \approx 4.27$. The resulting test statistic has $(25 + 5 - 2)$ degrees of freedom. The critical t -value at 28 degrees of freedom at $\alpha = 0.05$ is 2.048. Since the test statistic of 4.27 is greater than the critical t -value, the null hypothesis can be rejected. In fact, this hypothesis can also be rejected easily for $\alpha = 0.01$, which is not possible in the case of the unequal variance t -test even after tightening the degrees of freedom. The use of the similarity assumption allows an easier rejection of the null hypothesis as compared to the unequal variance test. ■

Problem 5.10 Repeat Problems 5.7 and 5.8 under the equal-variance assumption.

5.5.3 Paired t -Test

The paired t -test is useful when there are two sets of samples so that a one-to-one correspondence exists between the elements of the two sets. Therefore, the corresponding values in the two samples are related in some way (and not statistically independent). Some examples are as follows:

- The 100-meter sprinting times for the same athletes in Los Angeles (at low altitude) and Denver (at high altitude) are measured. We want to determine if athletes run faster at low altitudes as compared to high altitudes. Each athlete, therefore, has a pair of running times — one from the low-altitude group and the other from the

high-altitude group. Note that the running times are paired because they are not statistically independent. The champion runner in Denver will often be the champion runner in Colorado as well.

- The blood-sugar measurements for a set of diabetic patients could be measured before and after a particular treatment. Therefore, the blood-sugar measurements are paired, corresponding to the conditions before and after the treatment.
- The binary classification accuracy of each data point is measured with two different classifiers over the *same* sample of data points. These measurements are used to identify whether one classifier is better than the other. Note that some points may be mislabeled or be otherwise resistant to accurate classification. Such points will show similar behavior across the two classifiers.

Let μ_1 and μ_2 be the means of the two populations that are being compared. Then, the null and alternate hypotheses are as follows:

$$\begin{aligned} H_0 &: \mu_1 - \mu_2 = 0 \\ H_1 &: \mu_1 - \mu_2 \neq 0 \end{aligned}$$

One cannot simply use the unequal or equal variance tests in these case *because of the dependence among the samples drawn from the two populations*. For example, the standard error of the difference in means cannot be computed from the standard error of the individual means, when it is known that there is a pair-wise correlation among the samples. However, it turns out that one can rewrite the above hypothesis in terms of a single hypothesis on a derived random variable. Let Y_i be the difference between the i th pair of values $X_i(1)$ and $X_i(2)$ in the two populations. Therefore, we have the following:

$$Y_i = X_i(1) - X_i(2)$$

Then, the mean μ_Y of the derived random variable is equal to the difference $(\mu_1 - \mu_2)$ of the means in the two populations. Therefore, the hypothesis can now be rewritten in terms of a single random variable as follows:

$$\begin{aligned} H_0 &: \mu_Y = 0 \\ H_1 &: \mu_Y \neq 0 \end{aligned}$$

Note that this hypothesis has the same structure as a single-population hypothesis test. Therefore, one can compute the realized differences in values y_1, y_2, \dots, y_n over the n samples. These are used to compute the realized mean $\bar{y} = \sum_i y_i/n$ and sample standard deviation $\hat{\sigma}_Y$ of the differences:

$$\hat{\sigma}_Y^2 = \frac{\sum_{i=1}^n y_i^2}{n} - \bar{y}^2$$

The standard-error of the differences is, therefore, equal to $\hat{\sigma}_Y/\sqrt{n}$. The test-statistic is then computed as follows:

$$t = \frac{\bar{y} - 0}{\hat{\sigma}_Y/\sqrt{n}} \tag{5.1}$$

The corresponding random variable T (with realized test-statistic t of the specific sample) is drawn from a t -distribution with $(n - 1)$ -degrees of freedom. As in the case of a t -test with a single population, the p -value is computed as follows:

$$p = P(T > |t|) + P(T < -|t|)$$

The null hypothesis is rejected at significance level α , when the p -value is less than α . Alternatively, one can compare the realized test statistic t to (two-tailed) critical t -value at significance level α and reject the null hypothesis when the realized test statistic is greater than the critical t -value in absolute magnitude (for the two-tailed test). The corresponding confidence intervals on the difference in means at significance level α are as follows:

$$\left[\bar{y} - t_{\alpha/2} \frac{\hat{\sigma}_Y}{\sqrt{n}}, \bar{y} + t_{\alpha/2} \frac{\hat{\sigma}_Y}{\sqrt{n}} \right]$$

This confidence interval tells us the interval in which the difference in population means lies a fraction $(1 - \alpha)$ of the time, if the same experiment were to be repeated a large number of times. Note that one can reject the hypothesis at significance level α , when the confidence interval does not contain 0 and the entire interval has values of the same sign.

One can also use one-sided tests to determine if the first set of values are greater on average than the second set of values. The null hypothesis and the alternate hypothesis may now be defined as follows (in terms of the differences in paired values):

$$H_0 : \mu_Y = 0$$

$$H_1 : \mu_Y > 0$$

The main difference from the two-tailed test is in terms of the alternate hypothesis, which is one-sided. One can compute the test statistic as before and reject the null hypothesis if $P(T > t)$ is less than α . Alternatively, the null hypothesis can be rejected when the realized value of t is larger than t_α . A similar approach applies to the case when the alternate hypothesis is $\mu_Y < 0$. In that case, the null hypothesis is rejected when $P(T < t)$ is less than α . Alternatively, the null hypothesis can be rejected when the realized value of t is less than $-t_\alpha$. The derivation of expressions for confidence intervals is left as an exercise (see Problem 5.11).

Example 5.12 A data set of ten test instances is used to predict class labels with classification algorithms A, B, and C. Classification algorithm A gets the first two instances incorrect and the others correct. Classification algorithm B gets the first five instances incorrect and the others correct. Classification algorithm C gets the first five instances correct and the last five instances incorrect. Compare all pairs of algorithms and determine if one algorithm is better than the other at 90% level of confidence using a two-tailed test.

Solution: For each algorithm, we have 10 accuracy values, where a correct prediction is set to 1 and an incorrect prediction is set to 0. We test the average difference in accuracy values for each instance.

Algorithm B versus C: Classification algorithms B and C are not different from one another at any level of significance because the difference in their accuracy is 0.

Algorithm A versus B: The difference in prediction accuracies between the algorithms over the ten points are 0, 0, 1, 1, 1, 0, 0, 0, 0, 0. The mean accuracy difference and standard deviation of difference in prediction accuracy are as follows:

$$\hat{\mu}_{AB} = 0.3$$

$$\hat{\sigma}_{AB} = \sqrt{\frac{2.1}{10 - 1}}$$

The standard error over 10 points is therefore $\hat{\sigma}_{AB}/\sqrt{10} = \sqrt{2.1/90}$. The test statistic is computed as follows:

$$t = 0.3\sqrt{90/2.1} \approx 1.964$$

The critical test statistic at 9 degrees of freedom is 1.833. Therefore, the hypothesis that the algorithms perform similarly can be rejected at the 90% confidence level. Algorithm A is statistically more accurate than algorithm B at the 90% confidence level.

Algorithm A versus C: The difference in prediction accuracies between the algorithms over the ten points are -1, -1, 0, 0, 0, 1, 1, 1, 1, 1. The mean accuracy difference and standard deviation of difference in prediction accuracy are as follows:

$$\begin{aligned}\hat{\mu}_{AC} &= 0.3 \\ \hat{\sigma}_{AC} &= \sqrt{\frac{6.1}{10-1}}\end{aligned}$$

The standard error over ten points is therefore $\hat{\sigma}_{AC}/\sqrt{10} = \sqrt{6.1/90}$. The test statistic is computed as follows:

$$t = 0.3\sqrt{90/6.1} \approx 1.15$$

Since the critical test statistic is 1.8333 at the 90% confidence level, one cannot reject the hypothesis that the algorithms A and C are similar.

Algorithms B and C have the same accuracy. Yet, algorithm A is significantly better than algorithm B but algorithm A is not significantly better than algorithm C (from a hypothesis testing perspective). Why is this? The key point is that algorithms A and C tend to be correct on different test instances. The negative correlation between the predictions of the two algorithms A and C magnifies the variance of their differences. This causes their difference in accuracy to become less significant (even though their absolute difference in accuracy is the same as that between algorithms A and B). ■

Problem 5.11 Derive the expressions for the one-sided confidence intervals for the paired t -tests in cases where the alternate hypotheses are $\mu_Y > 0$ and $\mu_Y < 0$, respectively.

5.6 χ^2 -Hypothesis Tests

A number of hypothesis tests are based on the χ^2 -distribution, which is introduced in section 4.12 of Chapter 4. The χ^2 -distribution is defined by the sum of squares of standard normal variables. The χ^2 -distribution is used for various types of tests that consider sample statistics in sum-of-squares form. It turns out that many hypothesis tests can be converted to this form such as testing the standard-deviation of a normal distribution, checking the goodness of fit of observed data to a normal distribution, and checking independence of variables in contingency tables. This section will discuss these different types of hypothesis tests.

5.6.1 Standard Deviation Hypothesis Test

A χ^2 -test can be used to test if the population standard deviation is equal to a specified value σ_0 (which is the null hypothesis). The alternate hypothesis (in the case of the two-tailed test) is that the population standard deviation is not equal to the specified value σ_0 . Therefore, the null hypothesis and the alternate hypothesis are formally stated as follows:

$$\begin{aligned} H_0 &: \sigma = \sigma_0 \\ H_1 &: \sigma \neq \sigma_0 \end{aligned}$$

It is assumed in this case that the population is normally distributed. This assumption is particularly important in this case, as the central limit theorem cannot be used. Therefore, the populations should at least be roughly bell-shaped for the hypothesis test to provide meaningful results.

Let $X_1 \dots X_n$ be the random variables corresponding to the i.i.d. observations from the normal distribution with mean \bar{X} . Then, consider the following random variable representing the test statistic T :

$$T = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{\sigma_0^2}$$

Since each X_i is normally distributed, the random variable T can be shown to have a χ^2 -distribution with $(n - 1)$ -degrees of freedom. This result is based on Cochran's theorem (cf. page 171):

$$T \sim \chi^2(n - 1)$$

Given the specific samples x_1, x_2, \dots, x_n with sample mean \bar{x} , the realized test statistic t can be computed as follows:

$$t = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{\sigma_0^2}$$

One does not have to compute the sample variance, since the population variance has already been hypothesized to be σ_0 , which can be directly used in the computation of the test statistic. Then, at the significance level α , the two critical values $\chi_{\alpha/2}^2$ and $\chi_{1-\alpha/2}^2$ are identified. Here, $\chi_{\alpha/2}^2$ is the upper-tail critical value from Figure 4.18, so that values above this threshold contain a fractional area of $\alpha/2$ in the upper-tail of the χ^2 -distribution. In other words, we have $P(T > \chi_{\alpha/2}^2) = \alpha/2$. Similarly, the value $\chi_{1-\alpha/2}^2$ is the critical value¹ from Figure 4.19, and the lower-tail area of $\alpha/2$ occurs below this critical value in the χ^2 -distribution. In other words, we have $P(T < \chi_{1-\alpha/2}^2) = \alpha/2$. Then, the null hypothesis is rejected if the realized test statistic t is either greater than $\chi_{\alpha/2}^2$ or if it is less than $\chi_{1-\alpha/2}^2$.

For example, consider a setting in which the test statistic is constructed with 10 samples, and therefore we have a χ^2 -distribution with 9 degrees of freedom. When $\alpha = 0.05$, one must look for the critical values in Figures 4.18 and 4.19 that correspond to the tail probabilities of $\alpha/2 = 0.025$ at 9 degrees of freedom. The corresponding critical values from these tables are $\chi_{0.975}^2 = 2.7004$ and $\chi_{0.025}^2 = 19.0228$. Therefore, the null hypothesis is rejected when the test statistic lies outside the range [2.7004, 19.0228].

One can also generate confidence intervals for the standard deviation of the normal distribution. The confidence interval on the standard deviation at significance level α tells us that if the experiment of drawing samples and constructing the confidence interval is

¹From a notational perspective, if $F_X(x)$ is the cumulative distribution of the χ^2 -distribution with the appropriate degrees of freedom, then $F_X(\chi_\alpha^2) = 1 - \alpha$.

repeated a large number of times, the population standard deviation will lie in the correspondingly computed range a fraction $(1 - \alpha)$ of the time. First, note that since the test statistic follows the χ^2 -distribution, the following holds true:

$$P(\chi_{1-\alpha/2}^2 \leq T \leq \chi_{\alpha/2}^2) = 1 - \alpha$$

Substituting for T , one obtains the following:

$$P\left(\chi_{1-\alpha/2}^2 \leq \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{\sigma_0^2} \leq \chi_{\alpha/2}^2\right) = 1 - \alpha$$

Taking the reciprocal of each term and inverting the corresponding inequalities, one obtains the following:

$$P\left(\frac{1}{\chi_{\alpha/2}^2} \leq \frac{\sigma_0^2}{\sum_{i=1}^n (X_i - \bar{X})^2} \leq \frac{1}{\chi_{1-\alpha/2}^2}\right) = 1 - \alpha$$

Multiplying all sides with $\sum_{i=1}^n (X_i - \bar{X})^2$ and taking the square-root, one obtains the following:

$$P\left(\sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{\chi_{\alpha/2}^2}} \leq \sigma_0 \leq \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{\chi_{1-\alpha/2}^2}}\right) = 1 - \alpha$$

Therefore, for the specific sample x_1, x_2, \dots, x_n with mean \bar{x} , the confidence interval at significance level α is the following:

$$\left[\sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{\chi_{\alpha/2}^2}}, \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{\chi_{1-\alpha/2}^2}}\right]$$

Here, the χ^2 -distribution is computed for $(n - 1)$ degrees of freedom.

The aforementioned results are defined for the case of the two-tailed confidence tests. For the upper-tail confidence tests, the hypotheses are as follows:

$$H_0 : \sigma = \sigma_0$$

$$H_1 : \sigma > \sigma_0$$

The test statistic is computed in the same way as in the two-tailed case. The hypothesis is rejected when the test statistic is larger than χ_α^2 at $(n - 1)$ degrees of freedom. Note that the critical value χ_α^2 is used for the one-tailed test (instead of $\chi_{\alpha/2}^2$), since the rejection probability is concentrated only in the upper tail. The corresponding confidence interval for the standard deviation has no upper bound, and is defined as follows:

$$\left[\sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{\chi_\alpha^2}}, \infty\right]$$

The hypotheses for the lower-tail confidence tests are as follows:

$$H_0 : \sigma = \sigma_0$$

$$H_1 : \sigma < \sigma_0$$

The test statistic is computed in the same manner as the two-tailed case, and the hypothesis is rejected when the test statistic is smaller than $\chi_{1-\alpha}^2$. The corresponding confidence interval is as follows:

$$\left[0, \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{\chi_{1-\alpha}^2}}\right]$$

Note that the lower bound is 0, since a standard deviation cannot be less than 0. The critical values of the χ^2 -distribution for making these computations are obtained from the tables in Figures 4.18 and 4.19.

Example 5.13 Suppose that you draw the data points 1, 3, 4, 4, 5, 8, 10 from an unknown distribution. You want to test the null hypothesis that the standard deviation of the data set is 1.5. Can the hypothesis be rejected at the $\alpha = 0.05$ significance level? Based on the data, formulate the 95% confidence interval for the population standard deviation.

Solution: The mean of the data set is $35/7 = 5$. The test statistic t is the normalized sum-of-square differences from the mean:

$$t = \frac{(-4)^2 + (-2)^2 + (-1)^2 + (-1)^2 + 0^2 + 3^2 + 5^2}{1.5^2} = \frac{56}{2.25} \approx 24.889$$

The test statistic follows a χ^2 distribution with 6 degrees of freedom. The lower-tail critical value is 1.2373 and the upper-tail critical value is 14.4494 from Figures 4.18 and 4.19. Since the test statistic exceeds the upper-tail critical value, the hypothesis is rejected at the $\alpha = 0.05$ significance level. The true standard deviation is likely greater than 1.5.

The 95% confidence interval for the population standard-deviation is as follows:

$$\left[\sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{\chi_{\alpha/2}^2}}, \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{\chi_{1-\alpha/2}^2}} \right] = \left[\sqrt{\frac{56}{14.4494}}, \sqrt{\frac{56}{1.2373}} \right]$$

The above calculation yields the confidence interval [1.97, 6.73]. Note that the hypothesized standard deviation of 1.5 does not lie in this interval. This is yet another way of rejecting the null hypothesis.

Comment: This test relies on the normal distribution assumption. All the drawn points are integers, which is unlikely if the points had been drawn from a normal distribution. However, the normal distribution is only a heuristic assumption in such tests. ■

Problem 5.12 Consider the three points 1.32, 6.22, -3.11. Perform a hypothesis test at the $\alpha = 0.05$ significance level that the standard deviation of the data set is 2.0.

5.6.2 χ^2 -Goodness-of-Fit Test

The χ^2 -goodness-of-fit test is useful for measuring how well the sample data fits a specific distribution. This test is easiest to apply to distributions of discrete random variables, although it can be generalized to continuous random variables as well. The null hypothesis and the alternate hypothesis are defined as follows:

H_0 : The population follows the specified distribution

H_1 : The population does not follow the specified distribution

Consider the case where it is hypothesized that a set of n data values are drawn from a categorical distribution with parameters p_1, p_2, \dots, p_k for the k categorical values. Then, the

expected number of items belonging to the i th category is $E_i = n \cdot p_i$. Let O_i be the random variable corresponding to the number of observed occurrences of the i th category. Note that O_i belongs to a binomial distribution with mean np_i and variance $np_i(1 - p_i)$. When n is large, one can approximate the distribution of O_i by a normal distribution with mean $E_i = np_i$ and variance $np_i(1 - p_i)$ because of the central limit theorem. This approximation is true only when np_i is relatively large, and therefore one has to be careful of settings when the number of values in a particular category is small. In such cases, the goodness-of-fit test will not work. Now consider the following test statistic that is defined by the sum of squares of these zero-centered normally distributed variables:

$$T = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i}$$

Although there are k terms, they are not independent since the different values of O_i add to a constant. It can be shown that the test statistic follows a χ^2 -distribution with $(k - 1)$ -degrees of freedom, where k is the number of categories. A formal proof of this result is omitted. The reader is encouraged to work out Example 5.14 for intuition in the special case of $k = 2$. The null hypothesis is rejected at significance level α when the test statistic is greater than χ_α^2 . Here, χ_α^2 refers to the critical test-statistic value (cf. Figure 4.18) for which a fraction α of the area of the χ^2 -distribution lies above this value. It is noteworthy that only the upper tail is used for the goodness-of-fit tests.

A natural question arises as to how the goodness-of-fit can be applied to continuous distributions, such as the normal or exponential distribution. This is achieved with the use of binning. The data is discretized into k bins so that a roughly similar number of items are expected to fall in each bin. For most distributions, this approach will lead to varying bin widths. In the case of distributions where random variables do not have upper and lower bounds, the bins on the unbounded sides of the distribution may be unbounded as well. There is no specific guidance on how k should be selected, except that very small values of k will only be able to check whether the data matches a very rough linear approximation of the hypothesized distribution. On the other hand, very large values of k will result in too few items in each bin, which will cause the frequency of each bin to deviate from the normal approximation. At the end of the day, the test is only evidentiary in nature.

Example 5.14 Let p be the probability of success of a Bernoulli trial. A total of n trials are performed, where n is much larger than $\max\{1/p, 1/(1-p)\}$. The expected number of successes and failures are $E_1 = np$ and $E_2 = n(1 - p)$. Let the actual number of successes and failures be O_1 and O_2 , respectively. Show that the following statistic follows the χ^2 -statistic with one degree of freedom:

$$T = \sum_{i=1}^2 \frac{(O_i - E_i)^2}{E_i}$$

Solution: Although O_1 follows a binomial distribution, it can be approximated by a normal distribution for large n . The means and variances of O_1 and O_2 are $E_1 = np$ and $np(1 - p)$. Now consider the standardized version of O_1 :

$$Z = \frac{O_1 - E_1}{\sqrt{np(1 - p)}}$$

Note that Z is a standard normal distribution, and therefore Z^2 is a χ^2 -distribution with one degree of freedom. One can expand Z^2 as follows:

$$Z^2 = \frac{(O_1 - E_1)^2}{np(1-p)} = \frac{(O_1 - E_1)^2}{n} \left[\frac{1}{p} + \frac{1}{1-p} \right] = \frac{(O_1 - E_1)^2}{np} + \frac{(O_1 - E_1)^2}{n(1-p)}$$

A key observation here is that $O_1 + O_2 = E_1 + E_2 = n$. This means that $(O_1 - E_1)$ can be replaced with $(O_2 - E_2)$ in the last term of the above expression:

$$Z^2 = \frac{(O_1 - E_1)^2}{np} + \frac{(O_2 - E_2)^2}{n(1-p)}$$

However, the above expression is exactly the same as T . This proves that T is a χ^2 -distribution with one degree of freedom. ■

Example 5.15 A die is thrown 102 times and it results in face frequency vector of [15, 20, 12, 19, 17, 19] for the six faces. Evaluate the hypothesis that these frequencies represent the outcomes of a fair die at the $\alpha = 0.05$ significance level.

Solution: The value of E_i for each face is $102/6 = 17$. The corresponding test statistic for the first set of face frequencies may be computed as follows:

$$t = \frac{(-2)^2 + 3^2 + (-5)^2 + 2^2 + 0^2 + 2^2}{17} = \frac{46}{17} = 2.7059.$$

At five degrees of freedom, the upper-tail critical value is 11.07. Since the test statistic is less than this value, the hypothesis cannot be rejected. ■

Problem 5.13 Consider the normal distribution $X \sim \mathcal{N}(2, 6^2)$. Define six partitions of $[-\infty, \infty]$ so that one-sixth of the samples of X are expected to lie in these six ranges. Use these ranges to perform a goodness-of-fit test of the samples 6.260, 0.338, -5.340, 0.153, 3.618, 2.701, -8.787 with respect to $\mathcal{N}(2, 6^2)$ at the $\alpha = 0.05$ significance level.

5.6.3 Independence Tests

The χ^2 -goodness-of-fit test can also be used to test independence of attributes in contingency tables. Consider a movie theater that is trying to determine whether the type of food bought by patrons is related to the likelihood of buying drinks. The movie theater sells popcorn, hot dogs, and ice cream as the different types of foods. They track the number of sales of each type of food along with information about the sale of a drink with the food. The corresponding frequencies are shown in Table 5.5. The columns correspond to the different types of foods and the rows contain to the option of buying a drink or not buying it. It is immediately evident by glancing at the table that people buying salty or warm foods like popcorn and hot dogs are more likely to buy a drink than those buying ice cream. However, is the association significant? Are the two categorical attributes (corresponding to food and drink) independent of one another? The χ^2 -goodness-of-fit test is a way of achieving this goal.

Table 5.5: Contingency table of absolute frequencies of food/drink items sold by a movie theater.

	Popcorn	Hot dog	Ice cream	Total
Drink	3,497	2,932	800	7,229
No drink	513	657	1,601	2,771
Total	4,010	3,589	2,401	10,000

The two hypotheses in the case of the goodness-of-fit test for independence are as follows:

H_0 : The two attributes are independent of one another

H_1 : The two attributes are not independent of one another

Let n be the total number of items in the contingency table. Let r be the number of row items with fractional presence p_1, p_2, \dots, p_r . Let c be the number of column items with fractional presence q_1, q_2, \dots, q_c . Therefore, we have $\sum_{i=1}^r p_i = \sum_{j=1}^c q_j = 1$.

The χ^2 -goodness-of-fit test is similar to the goodness-of-fit test for checking the fit of data to a distribution. The first step is to compute the expected frequency of each cell E_{ij} using the total frequency n of the items over the entire table:

$$E_{ij} = n \cdot p_i \cdot q_j$$

For example, the value of E_{ij} for the popcorn-drink combination is as follows:

$$E_{11} = 10,000 * 0.7229 * 0.4010 = 2,898.8$$

The corresponding observed frequency in that cell was 3,497, which is larger than the expected value. In general, the observed frequency of the (i, j) th cell is O_{ij} and any difference from expected values is indicative of lack of independence of the two attributes. The corresponding test statistic T is as follows:

$$T = \sum_{i=1}^r \sum_{j=1}^c \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

The test-statistic T has a χ^2 -distribution with $(r - 1)(c - 1)$ degrees of freedom. The null hypothesis is rejected at significance level α when the test statistic is greater than χ_α^2 . Here, χ_α^2 refers to the critical test-statistic value (cf. Figure 4.18) for which a fraction α of the area of the χ^2 -distribution lies above this value. As in the case of the goodness-of-fit test for fitting data to distributions, only the upper tail is used in order to accept or reject the null hypothesis.

Example 5.16 Consider the observed frequency values O_{ij} shown in Table 5.5 between various types of food and the decision of choosing to have a drink. Evaluate the hypothesis that the choice of food is independent of the decision of having or not having a drink. You may use the significance level $\alpha = 0.05$.

Solution: The expected values of various types of foods with having or not having a drink are shown in Table 5.6. The corresponding realized value of the test statistic

Table 5.6: Expected values of absolute frequencies of food/drink items sold by a movie theater.

	Popcorn	Hot dog	Ice cream	Total
Drink	2,898.8	2,594.4	1,735.7	7,229
No drink	1111.2	994.5	665.3	2,771
Total	4,010	3,589	2,401	10,000

is as follows:

$$\begin{aligned}
 t &= \frac{(3497 - 2,898.8)^2}{2,898.8} + \frac{(2,932 - 2,594.4)^2}{2,594.4} + \frac{(800 - 1,735.7)^2}{1,735.7} + \\
 &\quad + \frac{(513 - 1,111.2)^2}{1,111.2} + \frac{(657 - 994.5)^2}{994.5} + \frac{(1,601 - 665.3)^2}{665.3} \\
 &= 123.45 + 43.93 + 504.43 + 322.03 + 114.54 + 1,316.00 = 2,424.8
 \end{aligned}$$

This test-statistic should be used in conjunction with the χ^2 -statistic with $(3 - 1) * (2 - 1) = 2$ degrees of freedom. The corresponding critical values are illustrated in Figure 4.18. Even at the $\alpha = 0.0005$ level of significance (99.95% confidence), the critical test statistic value is only 15.2018. Since the realized test statistic greatly exceeds this value, it is evident the attributes corresponding to food and drink are not independent of one another. Intuitively, the reason is that people rarely have a drink with ice cream, whereas they frequently have a drink with hot dogs. This dependence shows up in the increased value of the test statistic and therefore the hypothesis is violated. ■

This type of independence test is used frequently in machine learning to identify important inter-attribute associations. It is also used for feature selection applications in domains like text (cf. section 5.8.3). When particular words are related to important topics, a high χ^2 -statistic tells us that it is useful to retain those words for applications like classification that depend on the identification of those topics.

Problem 5.14 Consider a 2×2 contingency table on which the independence test is constructed and the χ^2 test statistic for independence is found to be 11.12. Would you reject the null hypothesis claiming independence at significance levels $\alpha = 0.05, 0.01, 0.001$? For the same test-statistic value of 11.12, would your answer change if your contingency table had been of size 2×3 ? How about a table of size 3×4 ?

5.7 Analysis of Variance (ANOVA)

Analysis of variance (ANOVA) was designed to identify whether there is a difference among the means of more than two groups. The basic idea was proposed by Ronald Fisher in the early twentieth century [23]. Many methods that are inspired by ANOVA have found their way into machine learning, such as the use of the *Fisher's discriminant index* and *linear discriminant analysis*. The most basic version of ANOVA is one-way ANOVA, which is the subject of this presentation. One-way ANOVA uses a single grouping to compare the means of the different groups in terms of a numerical variable, whereas multi-way ANOVA analyzes the effect of multiple groupings on the numerical variable. Although multi-way

ANOVA is popular in statistics, it is rarely used in machine learning. Therefore, the focus of this section will be on one-way ANOVA.

Consider a setting in which there are $k \geq 2$ groups $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_k$ of n numerical values denoted by x_1, x_2, \dots, x_n . The size of the j th group is denoted by $n_j = |\mathcal{G}_j|$, and therefore we have $\sum_{j=1}^k n_j = n$. It is assumed that the data in each group are normally distributed. Another important assumption in ANOVA is *variance equality*, which is that the variances of the different groups are equal. Note that variance equality is often not true in many practical settings in spite of which the approach continues to retain its usefulness in many settings on a heuristic basis. The two hypotheses may be stated as follows:

$$H_0 : \text{The groups } \mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_k \text{ have the same mean.}$$

$$H_1 : \text{The groups } \mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_k \text{ do not have the same mean.}$$

A key point is that since the groups are assumed to have the same variance, a difference in means will show up in terms of the ratio of inter-group variances with respect to intra-group variances. The first step is to compute the global sample mean $\hat{\mu}$ as well as the group-wise means $\hat{\mu}_j$ as follows:

$$\begin{aligned}\hat{\mu} &= \frac{\sum_{i=1}^n x_i}{n} \\ \hat{\mu}_j &= \frac{\sum_{i \in \mathcal{G}_j} x_i}{n_j} \quad \forall j \in \{1, \dots, k\}\end{aligned}$$

Even when the null hypothesis is true, there will be differences among the group-wise means $\hat{\mu}_j$ because of sampling variations. However, the differences ought to be modest in that case. The ANOVA test is based on this principle.

In order to compute the within-group and across-group differences, the sum of squared differences from the mean are computed both globally and for each group. Then, the total sum of squares TSS is defined as follows:

$$TSS = \sum_{i=1}^n (x_i - \hat{\mu})^2$$

Note that if σ_0^2 is the population variance, then TSS/σ_0^2 can be shown to be drawn from a χ^2 -distribution with $(n-1)$ degrees of freedom according to Cochran's theorem (cf. Theorem 4.1 of Chapter 4). Similarly, the within-group sum-of-squares WSS can be computed as follows:

$$WSS = \sum_{j=1}^k \sum_{x_i \in \mathcal{G}_j} (x_i - \hat{\mu}_j)^2$$

Note that WSS sums up the squares of the deviances within each group and it therefore quantifies the intra-group variance (without normalizing by the number of points).

One of the assumptions of ANOVA is that all groups have the same variance, which is denoted by σ_0^2 . Note that σ_0^2 is also the global variance when the null hypothesis is true and all group-wise means are the same as the global mean. Assuming that the population variance of each group is σ_0^2 , the quantity WSS/σ_0^2 can be shown to be drawn from a χ^2 -distribution with $(n-k)$ degrees of freedom. This is because the component of WSS from the i th group is a χ^2 -distribution with $(n_j - 1)$ degrees of freedom (based on Cochran's theorem).

One can also define the between-group sum-of-squares BSS by summing up the squares of the distances of the group means from the global mean *for each point*:

$$BSS = \sum_{j=1}^k n_j(\hat{\mu}_j - \hat{\mu})^2$$

Note that the j th term is weighted by n_j . Assuming that the null hypothesis is true and all groups have the same variance σ_0^2 , the quantity $n_j\hat{\mu}_j$ can be shown to be a sample from the normal distribution $\mathcal{N}(0, \sigma_0^2)$. Correspondingly, the quantity BSS/σ_0^2 can be shown to be drawn from a χ^2 -distribution with $(k - 1)$ of freedom using Cochran's theorem.

The quantities TSS , WSS , and BSS are related² as follows:

$$TSS = WSS + BSS$$

The *F-value* is defined as follows:

$$f = \frac{BSS/(k - 1)}{WSS/(n - k)} \quad (5.2)$$

Unequal group means will lead to an increased F-value. Furthermore, it turns out that the F-value has a specific distribution, referred to as the F-distribution (under the null hypothesis assumption). Therefore, the percentile values from this distribution can be used for hypothesis testing. The F-distribution has two parameters corresponding to the number of degrees of freedom in the numerator and the number of degrees of freedom in the denominator of the two χ^2 variables used to compute it. The number of degrees of freedom in the numerator is much smaller than the number of degrees of freedom in the denominator because the former depends on the number of groups, whereas the latter depends on the number of points. The expected value of the corresponding random variable F can be shown to be $(n - k)/(n - k - 2)$, which is close to 1 for $n \gg k$. The probability density function of the F-distribution is illustrated in Figure 5.4. When the random variable F is drawn from the F-distribution, the following notation is used for numerator degrees of freedom df_1 and denominator degrees of freedom df_2 :

$$F \sim Fdist(df_1, df_2)$$

It is evident from Figure 5.4 that the F-distribution is more sensitive to the number of degrees in the numerator as compared to the number of degrees of freedom in the denominator. For one or two degrees of freedom in the numerator, the mode of the F-distribution occurs at a value close to 0. The number of degrees of freedom in the denominator is much larger than the number of degrees of freedom in the numerator in practical applications. The shape of the F-distribution is more sensitive to the number of degrees of freedom in the numerator as compared to that in the denominator.

The null hypothesis is rejected at a level of significance α , if for realized F -value f , the corresponding random variable from the F -distribution (with appropriate numerator/denominator degrees of freedom) satisfies the following:

$$P(F > f) < \alpha$$

Rather than compute this probability explicitly, one can compute the critical value of F at the appropriate level of significance and reject the null hypothesis when the realized

²See Exercise 18 of Chapter 2 for hints on proving the result. The exercise is broken up into a series of parts that are easier to prove.

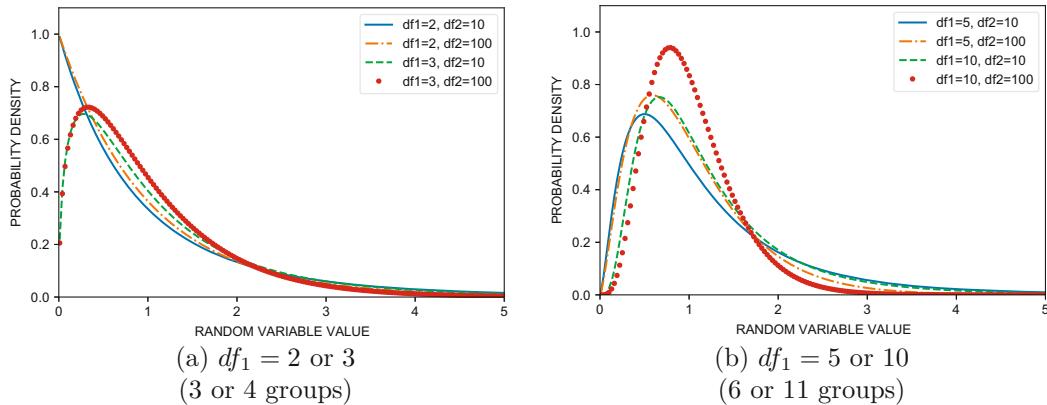


Figure 5.4: The F -distribution for varying number of groups. The value of df_2 is chosen to be 30 or 100, corresponding to using between 32 to 110 samples.

value f is greater than this critical value. The critical values of the F -distribution can be summarized in an F -table. The critical values of the F -distribution at levels of significance $\alpha = 0.05$ and $\alpha = 0.01$ are shown in Figures 5.5 and 5.6, respectively.

Example 5.17 The sample variance in heights of 75 students (25 each from Boston, Denver, and Albany) is 0.43 ft^2 . The mean heights of the samples from Boston, Denver, and Albany are 5.35 ft , 5.13 ft , and 5.21 ft , respectively. Evaluate the null hypothesis that the three groups have the same means at $\alpha = 0.05$.

Solution: Note that the values of WSS and BSS have not been directly provided, but can be computed from the provided information. First, note that TSS is $74 * 0.43 = 31.82$. The global mean of the heights and BSS is as follows:

$$\hat{\mu} = (5.35 + 5.13 + 5.21)/3 = 5.23$$

$$BSS = 25 * (5.35 - 5.23)^2 + 25 * (5.13 - 5.23)^2 + 25 * (5.21 - 5.23)^2 = 0.62$$

The value of WSS is the difference between TSS and BSS , which is $31.82 - 0.62 = 31.2$. Now, one can compute the F -value using Equation 5.2 as follows:

$$f = \frac{0.62/(3-1)}{31.2/(75-3)} = \frac{0.31}{0.433} \approx 0.715$$

The numerator degrees of freedom is 2 and the denominator degrees of freedom is 72. The critical value of the F -distribution for these degrees of freedom at the 95% level of confidence is 3.124 (see Figure 5.5). Since the F -value is less than this critical value, one cannot reject the null hypothesis that the group means are the same. ■

Problem 5.15 The sample variance in heights of 100 students (25 each from Boston, Denver, Albany, and Cincinnati) is 0.21 ft^2 . The mean heights of the samples from Boston, Denver, Albany, and Cincinnati are 5.52 ft , 5.33 ft , 5.10 ft , and 5.01 ft respectively. Evaluate the null hypothesis that the four groups have the same means at $\alpha = 0.05$.

Denominator Degrees of Freedom (rows)	Numerator Degrees of Freedom (columns)									
	1	2	3	4	5	6	7	8	9	10
1	161.4476	199.5000	215.7073	224.5832	230.1619	233.9860	236.7684	238.8827	240.5433	241.8817
2	18.5128	19.0000	19.1643	19.2468	19.2964	19.3295	19.3532	19.3710	19.3848	19.3959
3	10.1280	9.5521	9.2766	9.1172	9.0135	8.9406	8.8867	8.8452	8.8123	8.7855
4	7.7086	6.9443	6.5914	6.3882	6.2561	6.1631	6.0942	6.0410	5.9988	5.9644
5	6.6079	5.7861	5.4095	5.1922	5.0503	4.9503	4.8759	4.8183	4.7725	4.7351
6	5.9874	5.1433	4.7571	4.5337	4.3874	4.2839	4.2067	4.1468	4.0990	4.0600
7	5.5914	4.7374	4.3468	4.1203	3.9715	3.8660	3.7870	3.7257	3.6767	3.6365
8	5.3177	4.4590	4.0662	3.8379	3.6875	3.5806	3.5005	3.4381	3.3881	3.3472
9	5.1174	4.2565	3.8625	3.6331	3.4817	3.3738	3.2927	3.2296	3.1789	3.1373
10	4.9646	4.1028	3.7083	3.4780	3.3258	3.2172	3.1355	3.0717	3.0204	2.9782
11	4.8443	3.9823	3.5874	3.3567	3.2039	3.0946	3.0123	2.9480	2.8962	2.8536
12	4.7472	3.8853	3.4903	3.2592	3.1059	2.9961	2.9134	2.8486	2.7964	2.7534
13	4.6672	3.8056	3.4105	3.1791	3.0254	2.9153	2.8321	2.7669	2.7144	2.6710
14	4.6001	3.7389	3.3439	3.1122	2.9582	2.8477	2.7642	2.6987	2.6458	2.6022
15	4.5431	3.6823	3.2874	3.0556	2.9013	2.7905	2.7066	2.6408	2.5876	2.5437
16	4.4940	3.6337	3.2389	3.0069	2.8524	2.7413	2.6572	2.5911	2.5377	2.4935
17	4.4513	3.5915	3.1968	2.9647	2.8100	2.6987	2.6143	2.5480	2.4943	2.4499
18	4.4139	3.5546	3.1599	2.9277	2.7729	2.6613	2.5767	2.5102	2.4563	2.4117
19	4.3807	3.5219	3.1274	2.8951	2.7401	2.6283	2.5435	2.4768	2.4227	2.3779
20	4.3512	3.4928	3.0984	2.8661	2.7109	2.5990	2.5140	2.4471	2.3928	2.3479
21	4.3248	3.4668	3.0725	2.8401	2.6848	2.5727	2.4876	2.4205	2.3660	2.3210
22	4.3009	3.4434	3.0491	2.8167	2.6613	2.5491	2.4638	2.3965	2.3419	2.2967
23	4.2793	3.4221	3.0280	2.7955	2.6400	2.5277	2.4422	2.3748	2.3201	2.2747
24	4.2597	3.4028	3.0088	2.7763	2.6207	2.5082	2.4226	2.3551	2.3002	2.2547
25	4.2417	3.3852	2.9912	2.7587	2.6030	2.4904	2.4047	2.3371	2.2821	2.2365
26	4.2252	3.3690	2.9752	2.7426	2.5868	2.4741	2.3883	2.3205	2.2655	2.2197
27	4.2100	3.3541	2.9604	2.7278	2.5719	2.4591	2.3732	2.3053	2.2501	2.2043
28	4.1960	3.3404	2.9467	2.7141	2.5581	2.4453	2.3593	2.2913	2.2360	2.1900
29	4.1830	3.3277	2.9340	2.7014	2.5454	2.4324	2.3463	2.2783	2.2229	2.1768
30	4.1709	3.3158	2.9223	2.6896	2.5336	2.4205	2.3343	2.2662	2.2107	2.1646
35	4.1213	3.2674	2.8742	2.6415	2.4851	2.3718	2.2852	2.2167	2.1608	2.1143
40	4.0847	3.2317	2.8387	2.6060	2.4495	2.3359	2.2490	2.1802	2.1240	2.0772
45	4.0566	3.2043	2.8115	2.5787	2.4221	2.3083	2.2212	2.1521	2.0958	2.0487
50	4.0343	3.1826	2.7900	2.5572	2.4004	2.2864	2.1992	2.1299	2.0734	2.0261
60	4.0012	3.1504	2.7581	2.5252	2.3683	2.2541	2.1665	2.0970	2.0401	1.9926
70	3.9778	3.1277	2.7355	2.5027	2.3456	2.2312	2.1435	2.0737	2.0166	1.9689
80	3.9604	3.1108	2.7188	2.4859	2.3287	2.2142	2.1263	2.0564	1.9991	1.9512
90	3.9469	3.0977	2.7058	2.4729	2.3157	2.2011	2.1131	2.0430	1.9856	1.9376
100	3.9361	3.0873	2.6955	2.4626	2.3053	2.1906	2.1025	2.0323	1.9748	1.9267
Infinity	3.8415	2.9957	2.6049	2.3719	2.2141	2.0986	2.0096	1.9384	1.8799	1.8307

Figure 5.5: The critical values at various numerator/denominator degrees of freedom of the F -distribution at $\alpha = 0.05$. Each entry in the table shows the critical value of a random variable drawn from the F -distribution for which the upper-tail probability is $\alpha = 0.05$. For example, the entry 5.4095 in the table for numerator degrees of freedom set to 3 and denominator degrees of freedom set to 5 means that $P(F > 5.4095) \approx 0.05$ when $F \sim F_{dist}(3, 5)$.

Denominator Degrees of Freedom (rows)	Numerator Degrees of Freedom (columns)									
	1	2	3	4	5	6	7	8	9	10
1	4052.181	4999.500	5403.352	5624.583	5763.650	5858.986	5928.356	5981.070	6022.473	6055.847
2	98.5025	99.0000	99.1662	99.2494	99.2993	99.3326	99.3564	99.3742	99.3881	99.3992
3	34.1162	30.8165	29.4567	28.7099	28.2371	27.9107	27.6717	27.4892	27.3452	27.2287
4	21.1977	18.0000	16.6944	15.9770	15.5219	15.2069	14.9758	14.7989	14.6591	14.5459
5	16.2582	13.2739	12.0600	11.3919	10.9670	10.6723	10.4555	10.2893	10.1578	10.0510
6	13.7450	10.9248	9.7795	9.1483	8.7459	8.4661	8.2600	8.1017	7.9761	7.8741
7	12.2464	9.5466	8.4513	7.8466	7.4604	7.1914	6.9928	6.8400	6.7188	6.6201
8	11.2586	8.6491	7.5910	7.0061	6.6318	6.3707	6.1776	6.0289	5.9106	5.8143
9	10.5614	8.0215	6.9919	6.4221	6.0569	5.8018	5.6129	5.4671	5.3511	5.2565
10	10.0443	7.5594	6.5523	5.9943	5.6363	5.3858	5.2001	5.0567	4.9424	4.8491
11	9.6460	7.2057	6.2167	5.6683	5.3160	5.0692	4.8861	4.7445	4.6315	4.5393
12	9.3302	6.9266	5.9525	5.4120	5.0643	4.8206	4.6395	4.4994	4.3875	4.2961
13	9.0738	6.7010	5.7394	5.2053	4.8616	4.6204	4.4410	4.3021	4.1911	4.1003
14	8.8616	6.5149	5.5639	5.0354	4.6950	4.4558	4.2779	4.1399	4.0297	3.9394
15	8.6831	6.3589	5.4170	4.8932	4.5556	4.3183	4.1415	4.0045	3.8948	3.8049
16	8.5310	6.2262	5.2922	4.7726	4.4374	4.2016	4.0259	3.8896	3.7804	3.6909
17	8.3997	6.1121	5.1850	4.6690	4.3359	4.1015	3.9267	3.7910	3.6822	3.5931
18	8.2854	6.0129	5.0919	4.5790	4.2479	4.0146	3.8406	3.7054	3.5971	3.5082
19	8.1849	5.9259	5.0103	4.5003	4.1708	3.9386	3.7653	3.6305	3.5225	3.4338
20	8.0960	5.8489	4.9382	4.4307	4.1027	3.8714	3.6987	3.5644	3.4567	3.3682
21	8.0166	5.7804	4.8740	4.3688	4.0421	3.8117	3.6396	3.5056	3.3981	3.3098
22	7.9454	5.7190	4.8166	4.3134	3.9880	3.7583	3.5867	3.4530	3.3458	3.2576
23	7.8811	5.6637	4.7649	4.2636	3.9392	3.7102	3.5390	3.4057	3.2986	3.2106
24	7.8229	5.6136	4.7181	4.2184	3.8951	3.6667	3.4959	3.3629	3.2560	3.1681
25	7.7698	5.5680	4.6755	4.1774	3.8550	3.6272	3.4568	3.3239	3.2172	3.1294
26	7.7213	5.5263	4.6366	4.1400	3.8183	3.5911	3.4210	3.2884	3.1818	3.0941
27	7.6767	5.4881	4.6009	4.1056	3.7848	3.5580	3.3882	3.2558	3.1494	3.0618
28	7.6356	5.4529	4.5681	4.0740	3.7539	3.5276	3.3581	3.2259	3.1195	3.0320
29	7.5977	5.4204	4.5378	4.0449	3.7254	3.4995	3.3303	3.1982	3.0920	3.0045
30	7.5625	5.3903	4.5097	4.0179	3.6990	3.4735	3.3045	3.1726	3.0665	2.9791
35	7.4191	5.2679	4.3957	3.9082	3.5919	3.3679	3.2000	3.0687	2.9630	2.8758
40	7.3141	5.1785	4.3126	3.8283	3.5138	3.2910	3.1238	2.9930	2.8876	2.8005
45	7.2339	5.1103	4.2492	3.7674	3.4544	3.2325	3.0658	2.9353	2.8301	2.7432
50	7.1706	5.0566	4.1993	3.7195	3.4077	3.1864	3.0202	2.8900	2.7850	2.6981
60	7.0771	4.9774	4.1259	3.6490	3.3389	3.1187	2.9530	2.8233	2.7185	2.6318
70	7.0114	4.9219	4.0744	3.5996	3.2907	3.0712	2.9060	2.7765	2.6719	2.5852
80	6.9627	4.8807	4.0363	3.5631	3.2550	3.0361	2.8713	2.7420	2.6374	2.5508
90	6.9251	4.8491	4.0070	3.5350	3.2276	3.0091	2.8445	2.7154	2.6109	2.5243
100	6.8953	4.8239	3.9837	3.5127	3.2059	2.9877	2.8233	2.6943	2.5898	2.5033
Infinity	6.6349	4.6052	3.7816	3.3192	3.0173	2.8020	2.6393	2.5113	2.4073	2.3209

Figure 5.6: The critical values at various numerator/denominator degrees of freedom of the F -distribution at $\alpha = 0.01$. Each entry in the table shows the critical value of a random variable drawn from the F -distribution for which the upper-tail probability is $\alpha = 0.01$. For example, the entry 12.0600 in the table for numerator degrees of freedom set to 3 and denominator degrees of freedom set to 5 means that $P(F > 12.0600) \approx 0.01$ when $F \sim F_{dist}(3, 5)$.

5.8 Machine Learning Applications of Hypothesis Testing

Hypothesis testing has numerous applications in machine learning, beginning from performance evaluation of machine learning algorithms to extending to feature selection and classification. In the following exposition, an overview of some of the common applications of hypothesis testing will be provided, although this is not an exhaustive list.

5.8.1 Evaluating the Performance of a Single Classifier

As discussed in Chapter 1, a classification algorithm learns to predict labels from pairs of multidimensional training instances and labels, and then predicts the labels on a test set \mathcal{T} of unlabeled instances. A common measure of classifier performance is the *classification accuracy*, which measures the fraction of test instances for which the correct label is predicted. One problem with robustly evaluating classifier accuracy is that it varies from one test set to another, because the test set \mathcal{T} is assumed to be a sample from the population. The degree of variation varies heavily with the size m of the test sample \mathcal{T} . For example, when using only $m = 5$ test instances, a difference between an accuracy of 0.6 and 0.8 is simply a difference of predicting one more instance correctly. Therefore, it will be common to see this type of variation between test samples of this size. It would be hard to make a judgement of where the true accuracy of the classifier lies. On the other hand, if the test set contains one million instances, the degree of variation from one test sample to another will be quite small. The main problem is that the number of test sample is constrained by external factors involving practical limitations on data collection.

The aforementioned observation suggests that it is useful to not only present the accuracy value but also quantify the uncertainty in classification accuracy by providing a range in which the accuracy will lie most of the time. A natural way of doing this is with the use of confidence intervals. It is common to present the confidence intervals at a level of significance $\alpha = 0.05$, although one can use any value. One useful property of classification accuracy is that it can be expressed as an average of Bernoulli random variables, which greatly simplifies the creation of confidence intervals. Let A_i be a Bernoulli random variable that takes on the value of 1 when the i th test instance is classified correctly, and 0, otherwise. Then, the random variable C indicating the classification accuracy can be expressed as the mean of the different values of A_i over the m test instances:

$$C = \frac{\sum_{i=1}^m A_i}{m}$$

Note that if we knew the true classification accuracy $p \in (0, 1)$ at the population level, one can use this value to compute the variance of A_i as $p(1 - p)$. This variance can be used to compute the standard error of C as $\sqrt{p(1 - p)/m}$. Although the distribution of C is binomial, it can be approximated with a normal distribution with a mean of p and standard deviation of $\sqrt{p(1 - p)/m}$. One issue is that the true value of p is not known and only its estimated value \hat{p} (from the test sample) is known. Nevertheless, if the value of m is even modestly large (say, 100), one can substitute the sample estimate \hat{p} in lieu of p . Therefore, the 95% confidence intervals for the classification accuracy may be presented as follows:

$$\left[\hat{p} - 1.96\sqrt{\hat{p}(1 - \hat{p})/m}, \hat{p} + 1.96\sqrt{\hat{p}(1 - \hat{p})/m} \right]$$

When the sample sizes are small, it is appropriate to use a t -distribution rather than the standard-normal distribution. In such a case, the confidence interval at the level of signifi-

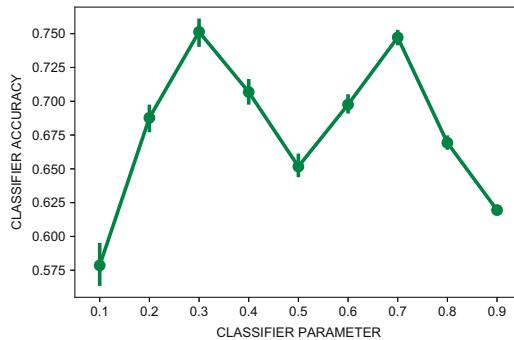


Figure 5.7: A plot of classifier performance with 95% confidence intervals

cance α is as follows:

$$\left[\hat{p} - t_{\alpha/2} \sqrt{\hat{p}(1 - \hat{p})/m}, \hat{p} + t_{\alpha/2} \sqrt{\hat{p}(1 - \hat{p})/m} \right]$$

These confidence intervals also provide the analyst with insights about the range in which the classification accuracy lies. It is also common to present the results graphically with the use of uncertainty bars in the plots. An example of such a plot is illustrated in Figure 5.7 in which the classifier performance variation is shown with respect to some hypothetical parameter affecting classifier accuracy. The analyst would like to determine the best range of parameters for their algorithm. The 95% confidence intervals are shown as vertical lines in Figure 5.7. One immediate observation is that there is greater variation of the algorithmic performance at smaller values of the parameter than at larger values of the parameter. This gives the analyst useful information about the robustness of the algorithm at different parameter values. It is also possible to present confidence intervals in the form of box plots. Some types of box plots in statistical software allow the box and whisker values set at specific confidence values instead of at multiples of inter-quartile ranges.

5.8.2 Comparing Two Classifiers

The discussion of the previous section shows that it is advisable to present the classification accuracy in the form of confidence intervals. Generally, the classifiers are compared using the same test sample. In such a case, there is a dependence between the accuracy of the two algorithms on the same test instance. In such cases, the paired t -test is more appropriate (cf. section 5.5.3).

Consider the case where the random variable defining whether algorithm 1 is accurate on the i th test instance is $A_1(i)$ and the algorithm 2 is accurate on the i th test instance is $A_2(i)$. Note that $A_1(i)$ and $A_2(i)$ are not independent of one another. This is because if a test instance is incorrectly labeled, most algorithms will perform poorly on it. Therefore, a new random variable Y_i corresponding to the difference in accuracy on the i th test instance is defined:

$$Y_i = A_1(i) - A_2(i)$$

Note that Y_i takes on the value of either 1, 0, or -1, depending on where or not either of the two algorithms perform correctly on the i th test instance. Even though $A_1(i)$ and $A_2(i)$ are not independent of one another, the random variables Y_1, Y_2, \dots, Y_m are independent of

one another. Then, the average difference in accuracy D over the m instances is defined as follows:

$$D = \frac{\sum_{i=1}^m Y_i}{m}$$

The random variable D corresponds to the difference in accuracy between the two algorithms. In the case of comparing two algorithms, one can use either a two-tailed test or a one-tailed test. The two-tailed test identifies whether the two algorithms perform differently in a significant way (along with an identification of the direction of the difference). The one-tailed test identifies whether one specific algorithm is better than the other in a significant way. In the two-tailed test, the null hypothesis and the alternate hypothesis are as follows:

$$\begin{aligned} H_0 &: D = 0 \\ H_1 &: D \neq 0 \end{aligned}$$

The procedure for testing this hypothesis is discussed in section 5.5.3.

5.8.3 χ^2 -Statistic for Feature Selection in Text

The χ^2 -Statistic is a useful tool for feature selection of categorical data. Text data is sometimes represented using the presence or absence of words in documents. For a classification problem, one can represent the combination of binary feature presence/absence with the class label as a contingency table. This type of contingency table allows the computation of the χ^2 -statistic, which is used to identify features that are discriminatory for classification.

Consider a classification scenario in which the class label corresponds to whether or not a user bought a specific item. The features correspond to the presence or absence of specific keywords (e.g., “phone”) in the item’s description. Therefore, the data contains documents in which binary features correspond to the presence of words (terms) in the documents. The documents are labeled with information about whether or not the user bought them. In order to elucidate this point, we will use a specific numerical example where frequencies of presence of term-label pairs are observed. Assume that the user has bought about 10% of the items in the collection, and the word w occurs in about 20% of the descriptions. Assume that the total number of items (and corresponding documents) in the collection is 1000. Then, the *expected* number of occurrences of each possible combination of word occurrence and class contingency is as follows:

	Term occurs in description	Term does not occur
User bought item	$1000 * 0.1 * 0.2 = 20$	$1000 * 0.1 * 0.8 = 80$
User did not buy item	$1000 * 0.9 * 0.2 = 180$	$1000 * 0.9 * 0.8 = 720$

The aforementioned expected values are computed under the assumption that the occurrence of the term in the description and the user interest in the corresponding item are independent of one another. If these two quantities are independent, then clearly the term will be irrelevant to the learning process. However, in practice, the item may be highly related to the item at hand. For example, consider a scenario where the contingency table deviates from expected values and the user is very likely to buy the item containing the term. In such a case, the contingency table may appear as follows:

	Term occurs in description	Term does not occur
User bought item	$O_1 = 60$	$O_2 = 40$
User did not buy item	$O_3 = 140$	$O_4 = 760$

The χ^2 -statistic measures the normalized deviation between observed and expected values across the various cells of the contingency table. In this case, the contingency table contains $p = 2 \times 2 = 4$ cells. Let O_i be the observed value of the i th cell and E_i be the expected value of the i th cell. Then, the χ^2 -statistic is computed as follows:

$$t = \sum_{i=1}^p \frac{(O_i - E_i)^2}{E_i} \quad (5.3)$$

Therefore, in the particular example of this table, the χ^2 -statistic evaluates to the following:

$$\begin{aligned} t &= \frac{(60 - 20)^2}{20} + \frac{(40 - 80)^2}{80} + \frac{(140 - 180)^2}{180} + \frac{(760 - 720)^2}{720} \\ &= 80 + 20 + 8.89 + 2.22 = 111.11 \end{aligned}$$

The afore-mentioned test-statistic is a χ^2 -distribution with $(2 - 1) * (2 - 1) = 1$ degree of freedom. Therefore, the critical values in Figure 4.18 can be used to identify the level of significance of the feature. Note that the expected value of a random variable drawn from the χ^2 -distribution with one degree of freedom is 1. However, this is too low a threshold, since one wants to select features with much greater informative power. As a practical matter, features with values greater than 10.83 already meet a level of confidence greater than 99.9% (see Figure 4.18). Therefore, the hypothesis of independence can be rejected at the 99.9% level. Any off-the-shelf classification algorithm can be used with the reduced set of features. A reduced set of features helps remove the effect of random variations in training data and also improves the efficiency of the algorithm.

It is also possible to compute the χ^2 -statistic as a function of the observed values in the contingency table without explicitly computing expected values. This is possible because the expected values are functions of the aggregate observed values across rows and columns. A simple arithmetic formula to compute the χ^2 -statistic in a 2×2 contingency table is as follows (see Exercise 11):

$$t = \frac{(O_1 + O_2 + O_3 + O_4) \cdot (O_1 O_4 - O_2 O_3)^2}{(O_1 + O_2) \cdot (O_3 + O_4) \cdot (O_1 + O_3) \cdot (O_2 + O_4)} \quad (5.4)$$

Here, $O_1 \dots O_4$ are the observed frequencies according to the table above. It is easy to verify that this formula yields the same χ^2 -statistic of 111.11.

Problem 5.16 Among the 200 articles in the sports category of a newspaper, the word “baseball” occurred in 36 of them. Among the 10000 articles in categories other than sports, the word “baseball” occurred in 300 of them. Compute a χ^2 -test of independence between the membership of an article in the sports category and the occurrence of the word “baseball” at the 99%-level of confidence.

5.8.4 Fisher Discriminant Index for Feature Selection

The Fisher discriminant index leverages the ANOVA method introduced in section 5.7. The ANOVA method was proposed by Ronald Fisher [23] who is also credited with the

Fisher discriminant [22]. One difference between the two methods is that ANOVA treats the grouping as the independent variable and the numerical values as the dependent variable, whereas the Fisher discriminant does the converse for problems like classification. Nevertheless, a scaled version of the F -value is computed using exactly the same approach and is directly leveraged for feature selection in classification. The numerical features that have a low F -value are not useful for distinguishing between different groups (classes), and can therefore be dropped. When used in machine learning, the F -value is modified to skip the normalization of the numerator and denominator by the number of degrees of freedom, and this quantity is referred to as the *Fisher discriminant index*. Recall from Equation 5.2 that the F-value is computed as follows:

$$f = \frac{BSS/(k-1)}{WSS/(n-k)}$$

On the other hand, The Fisher discriminant index f' is computed as follows:

$$f' = \frac{BSS}{WSS}$$

It is easy to see that the F-value is maximized whenever the Fisher discriminant index is maximized and vice versa. The two quantities differ by a constant factor of $(n-k)/(k-1)$. Feature selection algorithms for classification retain those features that have the largest value of the Fisher discriminant index.

Consider a set of d -dimensional point-label pairs $(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_n, y_n)$, which correspond to the n training points for classification. Each \vec{x}_i contains d numerical features (represented as a row vector), and y_i contains the class (group) identifier. The feature selection process for the training set is as follows:

1. For each of the d dimensions in \vec{x}_i , compute the Fisher discriminant index. Let the corresponding value for the i th dimension be denoted by f'_i based on the ratio of BSS to WSS for that numerical feature.
2. Select the top $r \leq d$ features with the largest value of f'_i in order to use for the classification algorithm.

Any off-the-shelf classification algorithm can be used with the reduced set of features. A reduced set of features helps remove the effect of random variations in training data and also improves the efficiency of the algorithm.

Example 5.18 Suppose that you have a standardized data set of 300 points in which 100 points belong to each of the three classes Blue, Green, and Red, respectively. You are trying to compare a pair of features in terms of their class discriminatory power. The mean of the Blue and Green classes are 0.2 and 0.1, respectively, for feature 1. The mean of the Blue and Green classes are 0.25 and -0.15, respectively, for feature 2. Calculate the Fisher discriminant index of each feature. Which feature is more discriminatory?

Solution: Since the data set is standardized, the variance is 1 for each feature and the TSS is $299 * 1 = 299$. Standardization also implies that the global means of the two features are 0. Since the classes contain an equal number of points, their means must sum to 0. This implies that the mean of the Red class is -0.3 and -0.1,

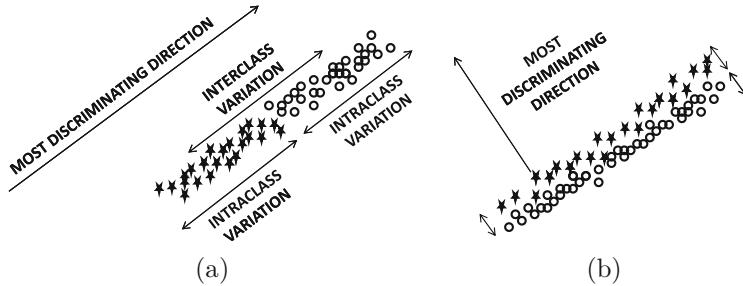


Figure 5.8: Different ways in which the most discriminating Fisher direction may be aligned

respectively for the features 1 and 2. Then, we can calculate the BSS for features 1 and 2 as follows:

$$\begin{aligned} BSS_1 &= 100 * [(0.2 - 0)^2 + (0.1 - 0)^2 + (-0.3 - 0)^2] = 14 \\ BSS_2 &= 100 * [(0.25 - 0)^2 + (-0.15 - 0)^2 + (-0.1 - 0)^2] = 9.5 \end{aligned}$$

One can compute WSS_i by subtracting BSS_i from $TSS = 299$ in each case:

$$\begin{aligned} WSS_1 &= 299 - 14 = 285 \\ WSS_2 &= 299 - 9.5 = 289.5 \end{aligned}$$

Therefore, the Fisher discriminant index for feature 1 is $14/285 \approx 0.0491$. The Fisher index for feature 2 is $9.5/289.5 \approx 0.0328$. The first feature is more discriminatory because it has the greater index. ■

5.8.5 Fisher Discriminant Index for Classification (*)

The discussion in section 5.8.4 uses the Fisher discriminant to find the r features that have the greatest discriminatory power in terms of maximizing the ratio of inter-class variation to intra-class variation. However, the r features still need to be used in combination with an off-the-shelf classification algorithm. The Fisher discriminant can also be used directly for classification by finding a single d -dimensional column-vector direction \vec{w} along which the classes are discriminated best — in other words, if the d -dimensional data points $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$ are projected into the new 1-dimensional numerical features $\vec{w} \cdot \vec{x}_1^T, \vec{w} \cdot \vec{x}_2^T, \dots, \vec{w} \cdot \vec{x}_n^T$, then the ratio of the inter-group variance to the intra-group variance with respect to this numerical feature is maximized. In other words, one wants to find \vec{w} that maximizes the following:

$$\text{Maximize}_{\vec{w}} \frac{BSS(\vec{w})}{WSS(\vec{w})}$$

Here, $BSS(\vec{w})$ is the inter-group (inter-class) scatter computed with respect to the new numerical feature values $\vec{w} \cdot \vec{x}_1^T, \vec{w} \cdot \vec{x}_2^T, \dots, \vec{w} \cdot \vec{x}_n^T$ and the notation $WSS(\vec{w})$ is the corresponding intra-group (intra-class) scatter.

In order to intuitively illustrate the importance of the direction \vec{w} , two examples have been shown in Figures 5.8(a) and (b) corresponding to data in $d = 2$ dimensions. The two figures illustrate different patterns of distribution of the two classes along with the

corresponding discriminating direction \vec{w} . In both cases, it is evident that projecting the data along the discriminating direction using the dot product of the data point with \vec{w} will lead to a 1-dimensional numerical representation of the data set. However, these directions are not equally good in terms of discriminatory power. The ratio of inter-class variation to intra-class variation is maximized in a direction of data “elongation” in the case of Figure 5.8(a) but not in the case of Figure 5.8(b). In fact, the most elongated direction is the least discriminating direction in Figure 5.8(b). Data analysts often use principal component analysis (i.e., elongated directions) to minimize loss of representational accuracy in dimensionality reduction. This example shows how such reduction methods can often lead to inadvertent loss in critical information in supervised problems.

By finding discriminatory directions in the data, the entire data set can be represented along a single discriminating direction. Representing the data in a single dimension leads to a particularly simple classification algorithm in which a point is classified to the label of the closest class mean based on its 1-dimensional numerical representation. Other variations of this approach exist in which thresholds on the value of the numerical feature are chosen to predict a class. The thresholds are chosen in order to maximize classification accuracy on the training data (so that this accuracy is also generalized to the test data). First, we will discuss the derivation of the most discriminating direction in both the two-class and the multi-class case.

5.8.5.1 Most Discriminating Direction for the Two-Class Case

Consider the case where the class variables y_1, y_2, \dots, y_n are drawn from one of two possibilities. Therefore, there are two groups with labels 0 and 1, respectively. The d -dimensional means of the two classes are $\vec{\mu}_0$ and $\vec{\mu}_1$, respectively, and these row vectors are obtained by averaging the values of \vec{x}_i for the corresponding classes. The squared distance between the means of the two classes along \vec{w} is given by $(\vec{w} \cdot \vec{\mu}_1 - \vec{w} \cdot \vec{\mu}_0)^2$. This quantity is proportional to the inter-class variance (or between-class scatter) $BSS(\vec{w})$:

$$\begin{aligned} BSS(\vec{w}) &\propto n(\vec{w} \cdot (\vec{\mu}_1 - \vec{\mu}_0)^T)^2 \\ &= \vec{w}^T \underbrace{[n(\vec{\mu}_1 - \vec{\mu}_0)^T(\vec{\mu}_1 - \vec{\mu}_0)]}_{d \times d \text{ matrix } S_b \text{ of rank-1}} \vec{w} \\ &= \vec{w}^T S_b \vec{w} \end{aligned}$$

The above relationships introduce an additional notation S_b by replacing the $d \times d$ rank-1 matrix $[n(\vec{\mu}_1 - \vec{\mu}_0)^T(\vec{\mu}_1 - \vec{\mu}_0)]$ with a between-class scatter matrix³ S_b .

Let n_0 and n_1 denote the respective numbers of points in the two classes. In order to compute the scatter *within* each class along direction \vec{w} , we make use of the well-known fact [41] that the scatter of a set of n points along a direction \vec{w} can be expressed in terms of the covariance matrix Σ as $n\vec{w}^T \Sigma \vec{w}$. Then, we leverage class-specific covariance matrices Σ_0 and Σ_1 to compute $WSS(\vec{w})$ as the sum of class-specific scatters:

$$\begin{aligned} WSS(\vec{w}) &= n_1(\vec{w}^T \Sigma_1 \vec{w}) + n_0(\vec{w}^T \Sigma_0 \vec{w}) \\ &= \vec{w}^T \underbrace{(n_1 \Sigma_1 + n_0 \Sigma_0)}_{d \times d \text{ matrix } S_w} \vec{w} = \vec{w}^T S_w \vec{w} \end{aligned}$$

³This two-class variant of the scatter matrix S_b is not exactly the same as defined in the multi-class version S_b of the next section. Nevertheless, all entries in the two matrices are related with the proportionality factor of $\frac{n_1 \cdot n_0}{n^2}$ which turns out to be inconsequential to the *direction* of the Fisher discriminant. In other words, the use of the multi-class formulas on page 239 will yield the same result in the binary case.

An additional notation, S_w , corresponding to the within-class scatter matrix is introduced above. Then, the objective function of the Fisher discriminant maximizes the ratio of the interclass to intra-class scatter along \vec{w} as follows:

$$\text{Maximize } J = \frac{BSS(\vec{w})}{WSS(\vec{w})} = \frac{\vec{w}^T S_b \vec{w}}{\vec{w}^T S_w \vec{w}}$$

Note that only the direction of \vec{w} matters in the above solution, and its scaling (i.e., norm) does not affect J . Therefore, in order to make the optimal solution unique, one can choose a scaling in which the denominator is 1. This creates a constrained optimization problem:

$$\begin{aligned} \text{Maximize } J &= \vec{w}^T S_b \vec{w} \\ \text{subject to:} \\ \vec{w}^T S_w \vec{w} &= 1 \end{aligned}$$

Setting the gradient of the Lagrangian relaxation $\vec{w}^T S_b \vec{w} - \alpha(\vec{w}^T S_w \vec{w} - 1)$ to 0 yields the following generalized eigenvector condition:

$$S_b \vec{w} = \alpha S_w \vec{w}$$

When this condition is satisfied, the objective function can be shown to be equal to α , and therefore choosing the eigenvector corresponding to the largest eigenvalue is important. The above condition can be reduced to an eigenvector condition for large data sets with $n \gg d$ in which the matrix S_w is invertible:

$$S_w^{-1} S_b \vec{w} = \alpha \vec{w}$$

A salient point is that a rank-1 matrix has only one non-zero eigenvector. Furthermore, the matrix $S_w^{-1} S_b$ can be shown to be positive semi-definite with only non-negative eigenvalues. Since the optimal objective function J is equal to the eigenvalue, it makes sense to choose \vec{w} as the only positive eigenvector of the rank-1 matrix $S_w^{-1} S_b$.

It is possible to solve the above eigenvector condition by expressing $S_b \vec{w}$ as follows:

$$\begin{aligned} S_b \vec{w} &= [n(\vec{\mu}_1 - \vec{\mu}_0)^T (\vec{\mu}_1 - \vec{\mu}_0)] \vec{w} \\ &= (\vec{\mu}_1 - \vec{\mu}_0)^T \underbrace{[n(\vec{\mu}_1 - \vec{\mu}_0) \vec{w}]}_{\text{Scalar}} \end{aligned}$$

Therefore, $S_b \vec{w}$ always points in the direction of $(\vec{\mu}_1 - \vec{\mu}_0)^T$, and it follows from the eigenvector condition that $S_w \vec{w} \propto (\vec{\mu}_1 - \vec{\mu}_0)^T$. Therefore, we have the following:

$$\vec{w} \propto S_w^{-1} (\vec{\mu}_1 - \vec{\mu}_0)^T \tag{5.5}$$

$$= (n_1 \Sigma_1 + n_0 \Sigma_0)^{-1} (\vec{\mu}_1 - \vec{\mu}_0)^T \tag{5.6}$$

It is also common to use a variant of this methodology in which a parameter γ is introduced to give differential weight to the various classes:

$$\vec{w} \propto (\Sigma_1 + \gamma \Sigma_0)^{-1} (\vec{\mu}_1 - \vec{\mu}_0)^T \tag{5.7}$$

One can choose γ by optimizing a desired cost function on held out portion of the data. Equal weighting to the classes irrespective of their relative population is achieved by setting $\gamma = 1$. However, the “official” Fisher discriminant is defined only by Equation 5.6.

How is the Fisher discriminant direction used for classification? It is common to learn a scalar bias variable b (which is referred to as a threshold) and classify the test instance \vec{z} to class 1 when the sign of $\vec{w} \cdot \vec{z}^T - b$ matches the sign of $\vec{w} \cdot (\vec{\mu}_1 - \vec{\mu}_0)^T$. The value of b is chosen by trying all n possible values $\vec{w} \cdot \vec{x}_i^T$ as candidates for b , and selecting the value that maximizes accuracy on the training data. It is possible to hold out a part of the training data as a validation set, which is not used for computing the direction \vec{w} . Using the validation set for determining b helps minimize overfitting to the training data.

Note that difference $(\vec{\mu}_1 - \vec{\mu}_0)^T$ in means is usually a good discriminating direction in most data sets. This is because the difference in means results in high inter-class variance. The purpose of multiplication with the matrix $(n_1\Sigma_1 + n_0\Sigma_0)^{-1}$ is to account for the intra-class variance because certain directions may be “elongated,” which changes the optimal direction of discrimination. We use an example to illustrate this case.

Example 5.19 Consider the case where class 0 contains the following set of four points $\{(9, 0), (-9, 0), (0, 1), (0, -1)\}$. This class distribution is relatively elongated in the X -direction, although covariances are 0 (to keep the problem simple). The instances for class 1 are obtained by adding $(10, 1)$ to each point of class 0, resulting in the data set $\{(19, 1), (1, 1), (10, 2), (10, 0)\}$ for class 1. Does projection along $(\vec{\mu}_1 - \vec{\mu}_0)^T = [10, 1]^T$ using the dot product separate the points in the two classes? Find the Fisher direction and project the points along that direction. Does the Fisher direction separate the points in the two classes? Comment on the result.

Solution: Projection along the vector $[10, 1]^T$ using the dot product yields $\{90, -90, 1, -1\}$ for class 0, whereas it yields $\{191, 11, 102, 100\}$ for class 1. These values are not separable because the value 90 in class 0 is greater than the value 11 in class 1, preventing the use of a threshold so that points in class 0 and class 1 are separated by a threshold.

The 2×2 matrix $(n_1\Sigma_1 + n_0\Sigma_0)$ can be shown to be diagonal with entries of $8*9^2 = 648$ and 8 along the diagonal. The Fisher direction is the following:

$$\vec{w} \propto \begin{bmatrix} 648 & 0 \\ 0 & 8 \end{bmatrix}^{-1} \begin{bmatrix} 10 \\ 1 \end{bmatrix} = \frac{1}{8} \begin{bmatrix} \frac{10}{81} \\ 1 \end{bmatrix}$$

The Fisher direction is therefore $[10/81, 1]^T$. The Fisher direction is quite different from the simple difference in means used above for projection — it points in the less elongated Y -direction to a greater degree because it adjusts for elongation. Projection along the vector $[10/81, 1]^T$ using the dot product yields $\{90/81, -90/81, 1, -1\}$ for class 0, whereas it yields $\{272/81, 91/81, 262/81, 100/81\}$ for class 1. Note that all values in class 1 are greater than $90/81$, which is the largest value in class 0. As a result, a threshold of $90/81$ yields perfect classification with the Fisher discriminant. This improvement over using the difference in means is because of the adjustment for directions of elongation in the Fisher discriminant. ■

The example above uses class data sets with uncorrelated attributes for simplicity. The Fisher discriminant can also adjust to arbitrary directions of elongation (beyond X - and Y -directions), when there are nonzero covariances among attributes.

5.8.5.2 Most Discriminating Direction for Multiple Classes

The aforementioned solution can be generalized to multiple classes in two ways. One can perform the classification using a one-against-all approach in which one class is selected as the positive class and the remaining classes are selected as the negative classes. This process is repeated k times, and the most confident prediction is returned for a test instance. Although the approach can be reasonably used for prediction, a more powerful approach is to use all the classes simultaneously to derive the $k - 1$ directions.

First, we need to compute the scatter matrices S_w and S_b for the multi-class setting. Let Σ_i be the covariance matrix of the i th class, so that the (j, k) th entry of Σ_i is equal to the covariance between the j th and k th dimensions in the i th class. Let n_i be the number of points in the i th class, and $n = \sum_i n_i$ be the total number of points. Let $\vec{\mu}$ be the d -dimensional row vector representing the mean of the entire data set, and $\vec{\mu}_i$ be the d -dimensional row vector representing the mean of the i th class. Then, the $d \times d$ within-class scatter matrix is defined as follows:

$$S_w = \sum_{i=1}^k n_i \Sigma_i \quad (5.8)$$

The $d \times d$ between-class scatter matrix⁴ is defined as the sum of the following rank-1 matrices:

$$S_b = \sum_{i=1}^k n_i (\vec{\mu}_i - \vec{\mu})^T (\vec{\mu}_i - \vec{\mu}) \quad (5.9)$$

Note that each product above is the product of a $d \times 1$ matrix with a $1 \times d$ matrix, which results in a $d \times d$ matrix. The matrix S_b is n times the covariance matrix of a data set containing the means of the classes in which the mean of the i th class is repeated n_i times. Then, the top eigenvector of the rank- $(k - 1)$ matrix $S_w^{-1} S_b$ provides the 1-dimensional direction along which the data is discriminated best.

For the multiclass case, the classification of test instances is done somewhat differently from the binary case. All training points are projected along the discriminating direction \vec{w} using the dot product $\vec{w} \cdot \vec{x}_i^T$. We also project the test instance \vec{z} along this direction as $\vec{w} \cdot \vec{z}^T$. Then, the t nearest training points to the test instance are determined using this projection (based on the minimum values of $\|\vec{w} \cdot \vec{x}_i^T - \vec{w} \cdot \vec{z}^T\|$). The dominant label among the t nearest points is reported as the class label for the test instance.

5.9 Summary

This chapter introduces hypothesis testing and confidence intervals, which are two of the fundamental principles underlying the scientific method. Hypothesis testing and confidence intervals can help quantify the level of uncertainty in estimating a statistical variable. It can also help in estimating whether one variable is truly larger than the other on average by creating confidence intervals on the difference in means. Multiple groups can be compared with the ANOVA method, which provides the mathematical intuition behind the well-known Fisher discriminant method used in machine learning. Hypothesis testing can be extended to testing other statistics such as variances, for testing the goodness of fit to a distribution, and for testing the independence of attributes in contingency tables.

⁴Note that this matrix is different from the one introduced for the two-class case only by a proportionality factor, which does not affect the final solution.

The two most common distributions used for hypothesis testing are the normal distribution and the t -distribution. However, the χ^2 -distribution can also be used for a number of independence tests and goodness-of-fit tests. It can also be used to derive confidence intervals on the standard deviation of a distribution. Hypothesis testing has numerous applications to various machine learning settings, such as classifier evaluation, feature selection, and the design of classification methods with the Fisher discriminant.

5.10 Further Reading

A formal introduction to hypothesis testing may be found in [62]. A somewhat less formal introduction that is written in intuitive style is given in [26]. A theoretical discussion of some of these statistical methods in the context of machine learning may be found in [33]; a practical discussion with R implementations may be found in [31]. The method of ANOVA was introduced by Ronald Fisher [23], and detailed discussions may be found in [16, 56]. The ANOVA method is closely related to Fisher's linear discriminant, which finds a direction that attempts to maximize the ratio of between-class variance to within-class variance. The Fisher discriminant method is also used for discriminant adaptive nearest neighbor classification [29], which is also discussed in detail in [3]. The use of χ^2 -goodness-of-fit for discovering associations and feature relationships is discussed in several data mining and machine learning books [1–3, 32]. The use of methods based on the χ^2 -statistic and the Fisher discriminant index for feature selection is discussed in [1–3].

5.11 Exercises

1. Consider a factory producing widgets in which the number of defects per widget is known to have a standard deviation of 0.4. A sample of 100 widgets is drawn from the factory production floor and is found to have an average of 1.1 defects per widget (i.e., 110 defects over all widgets). What is the p -value of the null hypothesis that factory widgets have one defect on average? The alternate hypothesis is that the mean number of defects per widget is different from 1. Would you reject this hypothesis at (i) the 95% confidence level, and (ii) the 99% confidence level?
2. Consider a factory manufacturing widgets in which the number of defects per widget is known to have a standard deviation of 0.4. A sample of 100 widgets is drawn from the factory production floor and is found to have an average of 1.1 defects per widget (i.e., 110 defects over all widgets). What is the two-sided confidence interval of the number of defects per widget at (i) the 95% confidence level, and (ii) the 99% confidence level?
Suppose you make the hypothesis that the mean number of defects per widget is 1. Would the hypothesis be rejected at (i) the 95% confidence level, and (ii) the 99% confidence level? Answer the last part using only the confidence interval (without using the p -value).
3. Consider a factory producing widgets in which the standard deviation of the number of defects per widget is not known. A sample of 30 widgets is drawn from the factory production floor and is found to have an average of 1.1 defects per widget (i.e., 33 defects over all widgets) and a standard deviation of 0.4 defects per widget. What is the p -value of the null hypothesis that factory widgets have one defect on average. The alternate hypothesis is that the mean number of defects per widget is different

from 1. Would you reject this hypothesis at (i) the 95% confidence level, and (ii) the 99% confidence level?

- 4.** Consider a factory manufacturing widgets in which the standard deviation of the number of defects per widget is not known. A sample of 30 widgets is drawn from the factory production floor and is found to have an average of 1.1 defects per widget (i.e., 33 defects over all widgets) and a standard deviation of 0.4 defects per widget. What is the confidence interval of the number of defects per widget at (i) the 95% confidence level, and (ii) the 99% confidence level?

Suppose you make the null hypothesis that the mean number of defects per widget is 1. Would the hypothesis be rejected at (i) the 95% confidence level, and (ii) the 99% confidence level? Answer the last part using only the confidence interval you calculated in this problem (without using the p -value).

- 5.** Consider a factory producing widgets in which the number of defects per widget is known to have a standard deviation of 0.4. A sample of 100 widgets is drawn from the factory production floor and is found to have an average of 1.1 defects per widget (i.e., 110 defects over all widgets). Consider a hypothesis that the mean number of defects per widget in the widget population is one. The alternative hypothesis is that the mean number of defects per widget in the widget population is greater than one. Would this one-tailed hypothesis be rejected at (i) the 95% confidence level, and (ii) the 99% confidence level?

- 6.** Find the one-sided 95%-confidence interval for the preceding problem.
- 7.** Consider two factories that produce widgets with some defects. Samples of size 30 are drawn from each factory and found to have an average of 1.1 and 1.3 defects per widget. The standard deviation of the number of defects per widget was found to be 0.3 and 0.4 from each factory floor. What is the p -value of the null hypothesis that the widgets produced by the two factories have the same mean number of defects? You may use the unequal variance t -test, and the number of degrees of freedom can be set in a relaxed way as $\min\{n_1 - 1, n_2 - 1\}$ for two samples of size n_1 and n_2 .
- 8.** For Exercise 7, compute the two-sided confidence interval on the difference in the mean number of defects at the 95%-level of confidence. Use this confidence interval to determine whether or not one should reject the null hypothesis that the expected number of defects in the widgets produced by the two factories are the same (at the 95% confidence level).
- 9.** Repeat Exercises 7 and 8 with the unequal variance t -test and tightened degrees of freedom.
- 10.** Repeat Exercises 7 and 8 under the equal-variance assumption.

- 11.** The χ^2 -statistic is defined in the chapter as follows:

$$\chi^2 = \sum_{i=1}^p \frac{(O_i - E_i)^2}{E_i}$$

Show that for a 2×2 contingency table, the aforementioned formula can be rewritten as follows:

$$\chi^2 = \frac{(O_1 + O_2 + O_3 + O_4) \cdot (O_1 O_4 - O_2 O_3)^2}{(O_1 + O_2) \cdot (O_3 + O_4) \cdot (O_1 + O_3) \cdot (O_2 + O_4)}$$

Here, $O_1 \dots O_4$ are defined in the same way as in the tabular example on page 233.

12. Consider the t -value for the paired t -test computed in Equation 5.1. Instead of computing $\hat{\sigma}_Y$ directly from samples of the differences, Jim sets $\hat{\sigma}_Y$ to $\sqrt{\hat{\sigma}_1^2 + \hat{\sigma}_2^2}$, where $\hat{\sigma}_1$ and $\hat{\sigma}_2$ are the standard deviations corresponding to the two sets of paired samples. This approach results in an identical statistic to the unequal variance t -test. Discuss why it is possible for Jim to get a drastically incorrect result using this approach.
13. The number of pigeons observed per minute by a bird watcher is hypothesized to be a Poisson distribution with $\lambda = 5$. The bird-watcher reports having seen 3, 4, 6, 5, 5, 7, 2, and 5 birds on successive minutes. Use a χ^2 -goodness-of-fit test to evaluate the null hypothesis that the distribution is Poisson. Use buckets of size 1 for variable values up to 8 and a single bucket for all values greater than or equal to 9.
14. The sample variance in SAT scores of 100 students (30 from Boston, 30 from Denver, and 40 from Albany) is 765.3. The mean SAT scores of the samples from Boston, Denver, and Albany are 643.2, 610.1, and 582.3, respectively. Evaluate the null hypothesis that the three groups have the same score means at the $\alpha = 0.05$ significance level.
15. Consider a classification problem with three class A, B, and C, containing 25, 25, and 50 points, respectively. You want to compare the discriminatory power of features 1 and 2. For feature 1, its global variance is 1.1. The means of the three classes A, B, and C along feature 1 are 0.1, 0.6, and 1.3, respectively. For feature 2, its global variance is 2.0. The means of the three classes A, B, and C along feature 2 are 0.15, 1.0, and 2.1, respectively. Calculate the Fisher discriminant index of both features. Which feature is more discriminative?
16. Suppose that you draw the data points 2, 5, 7, 10 from an unknown distribution. You want to test the null hypothesis that the standard deviation of the population is 2.0. Can the hypothesis be rejected at the $\alpha = 0.05$ significance level? Based on the data, formulate the 95% confidence interval for the population standard deviation.
17. You have a corpus of documents in labeled categories of sports, politics, and finance. These categories contain 35, 71, and 82 documents, respectively. The word “challenge” occurs in 9, 13, and 11 documents in the respective categories. Calculate the χ^2 -statistic relating the category membership to the occurrence of the word “challenge.” Is the hypothesis of independence rejected at the $\alpha = 0.05$ significance level?
18. Consider the case where class 0 contains the following set of four points $\{(1, 1), (1, 0), (0, 1), (0.5, 0.5)\}$. Class 1 contains $\{(-1, -1), (-2, 0), (-2, -1), (-3, -1)\}$. Find the Fisher direction and project the points along that direction.
19. The χ^2 -hypothesis test for variance in the text assumes that the population mean is not known. Suppose that the population mean μ_0 is known. How would this knowledge change the design of the hypothesis test? Assume that the given sample is x_1, \dots, x_n .
20. A binary classification algorithm classifies 71 out of 96 instances correctly. The null hypothesis is that the classification algorithm has 80% accuracy and the alternate hypothesis is that the accuracy is less than 80%. Compute the p -value.
21. Compute the one-sided confidence interval on the number of correct classifications for Exercise 20. Use it to evaluate the null hypothesis at the $\alpha = 0.05$ significance level.

- 22.** Consider a 2-dimensional binary classification data set. The variances of the individual dimensions are 1 and 3 for class 0. The covariance of the two dimensions in class 0 is 0.5. The variances of the individual dimensions are 2 and 1 in class 1. The covariance of the two dimensions in class 1 is -0.25 . Class 1 contains twice the number of points in class 0. Find the Fisher discriminant direction.



Chapter 6

Reconstructing Probability Distributions from Data

“With four parameters I can fit an elephant, and with five, I can make him wiggle his trunk.” — John von Neumann

6.1 Introduction

Machine learning applications often assume that the observed data is sampled from probability distributions. How can these probability distributions be reverse engineered from observed data? The main challenge is that the data analyst only has access to observed data but no a priori knowledge of the shape of the underlying probability distribution. In fact, the generating processes in the real world are so complex that the simplified distributions of Chapter 4 are woefully inadequate to model observed data. Therefore, one must model *simplified versions* of these distributions as approximations of the intractable complexities in the real world. The choice of the specific distribution family to use for modeling requires experience and insight from the analyst. In many cases, the data can be visually explored in order to create a first “guess” of the distribution shape (e.g., mixture of Gaussian distributions), while treating the parameters (e.g., Gaussian distribution mean and variance) as unknown quantities. The distribution parameters are estimated so that they can be used in order to make predictions about unknown properties of individual data instances. A specific example of how a distribution can be used to make predictions is the knowledge-based Bayes classifier in section 3.8.3 of Chapter 3. While the exposition in Chapter 3 assumes that the distribution parameters are already available from a knowledgeable expert, the common approach in machine learning is to *learn these parameters in a data-driven manner*. It is this process of learning parameters to reconstruct distributions that is the focus of this chapter.

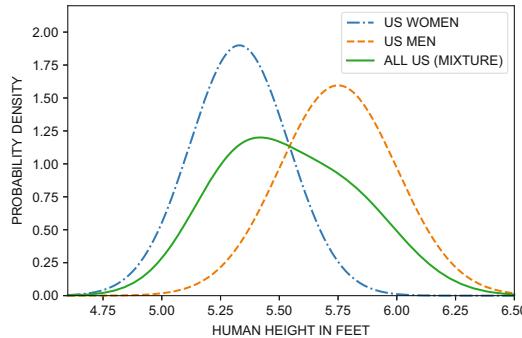


Figure 6.1: Revisiting Figure 4.20: The mixture distribution for US population height

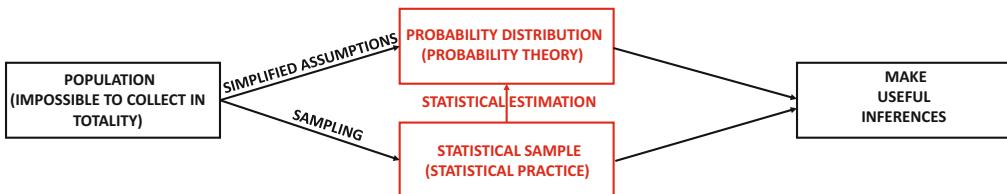


Figure 6.2: Revisiting Figure 1.6: The portion in red corresponds to the estimation of parameters of simplified distributions of observed data.

In order to understand this point, we will revisit an example of US male and female heights in Chapter 4. The mixture model representing US population heights is illustrated in Figure 6.1. Creating such a mixture model is the task of the analyst with the use of domain knowledge that a normal distribution should be used for each biological sex and that the entire distribution is a mixture of these distributions. The methodology of generating data from a mixture of two simplified distributions is referred to as a *generating process*, which reflects the analyst's insight and experience. Then, the observed data can be used in order to create estimations of the parameters (i.e., mean and variance of each normal distribution). Once the parameters of the mixture components have been estimated, they can be used to classify instances of height for which we do not know the biological sex. This process is identical to the inference portion of the knowledge-based Bayes classifier in section 3.8.3 of Chapter 3.

This chapter discusses the estimation of distribution parameters from observed data, once the analyst has selected the appropriate family of generating distributions along with a generating process. This important part of the machine learning process is shown in Figure 6.2. The overall process of reconstructing a distribution from observed data is also referred to as *fitting* a probability distribution to observed data. Fitting observed data to a family of distributions creates a model that is often used for making predictions in parts of the data space where observed data were not available. Therefore, moving from discrete instances to continuous distributions creates the ability to *generalize* predictions from known instances to previously unseen instances.

A desirable property of estimators is that they should be *unbiased* — the expected value of the estimated parameter (over random choices of data samples) must equal the true value of the parameters (assuming that the data was indeed generated from that distribution).

In other words, an unbiased estimate is a “correct” estimate of the parameters in terms of statistical expectation, although the specific parameters estimated from a particular sample may still be different from the true value because of estimation variance. Various techniques exist in statistics to estimate the parameters of a distribution from observed data, such as *minimum variance estimation*, *maximum likelihood estimation*, and *maximum a posteriori estimation*. The first of these methods finds estimates of parameters so that varying the specific sample of the observed data changes the estimates of the parameters as little as possible (i.e., the estimation of the parameter has the least variance with respect to choice of observed data sample). However, it is rarely used in machine learning. Our primary focus in this chapter will be on the other two techniques of maximum likelihood estimation and maximum a posteriori estimation.

It is noteworthy that optimizing the parameters of a distribution family is not the only approach for reconstructing distributions. A more general approach is kernel density estimation, which can be considered a smooth generalization of histograms in which points are replaced by smooth “bumps” and added together to create a smooth histogram. These bumps are referred to as kernels. This approach is a non-parametric way of reconstructing a probability distribution from data. One advantage of this approach is that it does not require any assumptions on the nature of the data distribution from the analyst up front. However, it might require more data for robust estimation, particularly when the number of attributes in the data is large.

6.1.1 Chapter Organization

This chapter is organized as follows. The next section introduces the methodology of maximum likelihood estimation. This approach is applied to specific distribution families in section 6.3. A more challenging case is one in which the parameters of a mixture of distributions need to be estimated. The expectation-maximization algorithm is discussed in section 6.4 in order to address this case. Kernel density estimation methods for reconstructing distributions are discussed in section 6.5. These methods can reconstruct distributions of arbitrary shape (unlike parametric models). The reduction of reconstruction variance is discussed in section 6.6. The bias-variance trade-off is introduced in section 6.7. Distributions that are popularly used for Bayesian analysis are discussed in section 6.8. A summary is given in section 6.9.

6.2 Maximum Likelihood Estimation

Maximum likelihood estimation (MLE) is a method for estimating the parameters of a probability distribution, so that the probability of the observed data being generated by that probability distribution is maximized. All the possible values of the parameters of a probability distribution are viewed as *hypotheses*, and we would like to select the specific hypothesis (i.e., parameter value(s)) that maximizes the probability of the observed data. The notion of *likelihood* was first introduced in section 3.6.1 of Chapter 3. Stated simply, a likelihood is a probability or probability-like quantification (e.g., probability density) of the observed data, conditional on a particular hypothesis. The probability of an entire data set of observed data points being generated is quantified with a *likelihood function*, which uses the both the parameters of the probability distribution and the observed data points in its arguments. The values of the parameters that maximize the likelihood of the observed data being generated by the model are referred to as the *maximum likelihood estimates*. One can

use methods from differential calculus in order to create optimality conditions for likelihood functions, so that the best-fitting values of the parameters may be estimated.

First, we will describe the construction of a likelihood function from a discrete distribution, because it is easier to interpret probability mass functions rather than probability density functions. However, the principles for probability density functions are exactly similar. Consider a data set with n (vector) instances denoted by $\vec{x}_1, \dots, \vec{x}_n$. The probability mass function of this discrete distribution for instance \vec{x}_i is $p_{\vec{X}}(\vec{x}_i)$. In maximum likelihood estimation, we treat the parameters $\vec{\Theta}$ of the distribution as random variables, and therefore we have a joint distribution of the data (examples of which are observed) and parameters (which are not observed). Our approach will be to use the PMF of the data after putting the distribution parameters in the conditional. Putting specific values $\vec{\Theta} = \vec{\theta}$ of the parameters in the conditional of the probability mass function allows us to treat the specific values $\vec{\theta}$ as deterministic values that can be optimized with calculus. Therefore, one can express the PMF of the random variable corresponding to the observed data vectors \vec{x}_i as $p_{\vec{X}|\vec{\Theta}=\vec{\theta}}(\vec{x}_i)$. Since the instances are assumed to be independently generated from the probability distribution, the overall probability of the entire data set being generated from the distribution is the product of these probabilities. Assume that the parameter vector of the probability distribution is denoted by $\vec{\theta}$, and therefore the probability mass function as well as the likelihood function $\mathcal{L}(\cdot)$ is expressed and the instances $\vec{x}_1 \dots \vec{x}_n$ as follows:

$$\mathcal{L}(\vec{x}_1 \dots \vec{x}_n, \vec{\theta}) = \prod_{i=1}^n p_{\vec{X}|\vec{\Theta}=\vec{\theta}}(\vec{x}_i)$$

We include $\vec{\theta}$ in the argument of the likelihood function¹ consistently in the text, as it needs to be optimized in terms of $\vec{\theta}$. Note that the generic notation $\vec{\theta}$ in the argument of the likelihood function is replaced with the parameters of the specific distribution at hand. For example, for the uniform distribution $U(a, b)$, the notation $\vec{\theta}$ is replaced with two arguments containing a and b . For the Bernoulli distribution $\text{Bernoulli}(p)$, the notation $\vec{\theta}$ is replaced with p .

The goal of maximum likelihood estimation is to find the value of the parameter vector $\vec{\theta}$ that maximizes the probability of the observed data — in other words, the probability of the observations given the parameters (choice of hypothesis) has been maximized:

$$\vec{\theta}^* = \operatorname{argmax}_{\vec{\theta}} \mathcal{L}(\vec{x}_1 \dots \vec{x}_n, \vec{\theta})$$

The optimal value $\vec{\theta}^*$ of the parameters is referred to as the *maximum likelihood estimate*. We also refer to the optimally estimated parameters with a circumflex like $\hat{\vec{\theta}}$ or as $\vec{\theta}^*$.

One issue with this optimization is that the likelihood function is in product-wise form, which is not computationally convenient for optimization with differentiation. One way to deal with this problem is to use the negative logarithm of the likelihood function in order to create the *log-likelihood function*. First, note that the logarithm is an increasing function, and therefore applying the negative of the logarithm changes the maximization problem into a minimization problem. Secondly, the use of the logarithm changes the product-wise form to a summation form, which enables easy differentiation:

$$\mathcal{LL}(\vec{x}_1 \dots \vec{x}_n, \vec{\theta}) = -\ln [\mathcal{L}(\vec{x}_1 \dots \vec{x}_n, \vec{\theta})] = -\sum_{i=1}^n \ln (p_{\vec{X}|\vec{\Theta}=\vec{\theta}}(\vec{x}_i))$$

¹In some textbooks, the notation for the likelihood function is $\mathcal{L}(\vec{\theta}|\vec{x}_1 \dots \vec{x}_n)$.

It is easy to differentiate the separable sum of point-wise functions, as the derivative simply gets distributed over the various functions. One can estimate the parameter vector by solving for the following (necessary) optimality condition² from differential calculus:

$$\frac{\partial \mathcal{L}(\vec{x}_1 \dots \vec{x}_n, \vec{\theta})}{\partial \vec{\theta}} = \vec{0}$$

Computing this derivative yields the following:

$$\sum_{j=1}^n \frac{1}{p_{\vec{X}|\vec{\Theta}=\vec{\theta}}(\vec{x}_j)} \frac{\partial p_{\vec{X}|\vec{\Theta}=\vec{\theta}}(\vec{x}_j)}{\partial \vec{\theta}} = \vec{0} \quad (6.1)$$

The above equation uses vector calculus notation in denominator layout, where the derivative of a scalar with respect to a vector $\vec{\theta}$ yields a vector of the same shape as $\vec{\theta}$; each element of the derivative vector is the differentiation of the scalar with the corresponding element of $\vec{\theta}$. There are exactly as many equations in the above vector of conditions as the number of variables in the parameter vector. In some cases, solving the above set of conditions yields a closed-form solution to the parameter vector. Examples of such closed-form solutions will be provided in the next section.

In general cases, a closed-form solution may not exist and iterative methods need to be used to solve Equation 6.1. The *expectation-maximization algorithm* for mixture distributions (cf. section 6.4) is a specific example of an iterative approach. Another popular approach that is used in some cases is to perform gradient descent updates with respect to the parameter vector $\vec{\theta}$ as follows:

$$\theta \leftarrow \theta - \alpha \frac{\partial \mathcal{L}(\vec{x}_1 \dots \vec{x}_n, \vec{\theta})}{\partial \vec{\theta}}$$

Here, $\alpha > 0$ is the learning rate.

The discussion thus far has focused on the use of probability mass functions, which are applicable to distributions of discrete variables. One can also define a similar likelihood function for continuous-variable distributions by simply replacing the probability mass function $p_{\vec{X}}(\vec{x}_i)$ with the probability density function $f_{\vec{X}}(\vec{x}_i)$ in the above expression for the likelihood function:

$$\mathcal{L}(\vec{x}_1 \dots \vec{x}_n, \vec{\theta}) = \prod_{i=1}^n f_{\vec{X}|\vec{\Theta}=\vec{\theta}}(\vec{x}_i)$$

All other steps for creating the negative log-likelihood function and optimizing it remain the same. Differentiating the negative log-likelihood function and setting it to 0 yields the following optimality condition (which is identical to Equation 6.1 except that the probability mass function has been replaced with the probability density function):

$$\sum_{j=1}^n \frac{1}{f_{\vec{X}|\vec{\Theta}=\vec{\theta}}(\vec{x}_j)} \frac{\partial f_{\vec{X}|\vec{\Theta}=\vec{\theta}}(\vec{x}_j)}{\partial \vec{\theta}} = \vec{0} \quad (6.2)$$

The quality of the negative log-likelihood fit depends heavily on whether an analyst chooses a reasonable distribution family to represent the data. For example, consider the

²These conditions are necessary but not sufficient, as they represent *critical points*, which could be maxima, minima, or inflection points. Adding second-derivative conditions can help ensure that these critical points are local or global minima. A detailed discussion of continuous optimization methods is given in [6].

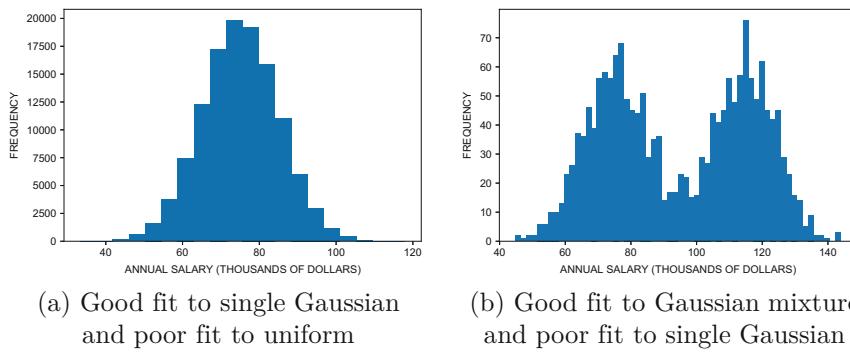


Figure 6.3: Good selection of distribution family is essential for good fit.

1-dimensional data represented by the histogram in Figure 6.3(a). In such a case, it makes sense for the analyst to choose the Gaussian distribution rather than the uniform distribution to model the data. If the uniform distribution is chosen, the quality of the optimal likelihood fit will be poor because of the mismatch between the shapes of the uniform distribution and the histogram at hand. Similarly, a mixture model is more appropriate for the case of Figure 6.3(b) as compared to a Gaussian distribution. These examples show that even though maximum likelihood estimation provides the optimal estimation of distribution parameters, it is not very helpful if the analyst makes a poor selection of the distribution family up front. This selection may require a phase of data exploration and visualization.

Example 6.1 Consider a family of density functions $f_{X|\Theta=\theta}(x) = \theta x^{\theta-1}$, where $\theta > 0$ is the distribution parameter and $x \in (0, 1)$. Find the maximum likelihood estimate $\hat{\theta} = \hat{\theta}^*$ in terms of observed data points $x_1 \dots x_n$ from this distribution. What is the numerical value of $\hat{\theta}^*$ for $n = 1$ and a single observed point x_1 when (a) $x_1 = 0.1$, (b) $x_1 = 0.5$, and (c) $x_1 = 0.9$. Comment on the trend.

Solution: The negative log-likelihood function is given by the following:

$$\mathcal{LL}(x_1 \dots x_n, \theta) = - \sum_{i=1}^n \ln(f_{X|\Theta=\theta}(x_i)) = -n\ln(\theta) - (\theta - 1)(\sum_{i=1}^n \ln(x_i))$$

On differentiating and setting to 0, one obtains the following:

$$\frac{n}{\theta} = - \sum_{i=1}^n \ln(x_i)$$

In other words, the optimal parameter $\hat{\theta}^*$ is as follows:

$$\hat{\theta}^* = - \frac{n}{\sum_{i=1}^n \ln(x_i)}$$

On substituting $x_1 = 0.1$ based on case (a), we obtain $\hat{\theta}^* = -1/\ln(0.1) \approx 0.434$. On substituting $x_1 = 0.5$ based on case (b), we obtain $\hat{\theta}^* = -1/\ln(0.5) \approx 1.443$. On substituting $\theta = 0.9$ for case (c), we obtain $\hat{\theta}^* = -1/\ln(0.9) \approx 9.491$.

The optimal estimate of θ^* increases with the value of the observed data point. This is because large values of θ result in distributions with greater probability mass on the right. ■

Example 6.2 Two biased coins with heads probabilities p_1 and p_2 , respectively, are flipped simultaneously. The payoff is equal to the square of the number of heads. The experiment is repeated n times and pay-offs of 0, 1, and 4 are observed f_0 , f_1 and f_2 times. Set up the equation(s) to find the maximum likelihood estimates of p_1 and p_2 . Can the equation(s) be solved in closed form? Now find the maximum likelihood estimate for a single experiment and (a) reward of 0, (b) reward of 1, and (c) reward of 4.

Solution: The payoff values can only be 0, 1, and 2², since the maximum number of heads is 2. We first set up the PMF of the payoff as follows:

$$p_{X|\vec{\Theta}=[p_1, p_2]}(x) = \begin{cases} (1-p_1)(1-p_2) & \text{if } x=0 \\ p_1(1-p_2) + p_2(1-p_1) & \text{if } x=1 \\ p_1p_2 & \text{if } x=4 \end{cases}$$

The negative log-likelihood of the reward is set up using the PMF:

$$\begin{aligned} \mathcal{LL}(f_0, f_1, f_2, p_1, p_2) &= -\sum_{i=1}^n \ln(p_{X|\vec{\Theta}=[p_1, p_2]}(x_i)) \\ &= -f_0 \ln((1-p_1)(1-p_2)) - f_1 \ln(p_1 + p_2 - 2p_1p_2) - f_2 \ln(p_1p_2) \end{aligned}$$

The partial derivatives with respect to p_1 and p_2 yield the following:

$$\begin{aligned} \frac{f_0}{1-p_1} - \frac{f_1(1-2p_2)}{p_1 + p_2 - 2p_1p_2} - \frac{f_2}{p_1} &= 0 \\ \frac{f_0}{1-p_2} - \frac{f_1(1-2p_1)}{p_1 + p_2 - 2p_1p_2} - \frac{f_2}{p_2} &= 0 \end{aligned}$$

This relationships can be shown to be third-order polynomial equations, for which a closed-form solution does not always exist. In general, numerical methods must be used to solve such systems. Furthermore, since the values p_1, p_2 lie in $(0, 1)$, it is necessary to check the interval end points for possible optimality.

Now considering $[f_0, f_1, f_2] = [1, 0, 0]$ for case (a), we obtain the optimality conditions $1/(1-p_1) = 0$ and $1/(1-p_2) = 0$. These optimality conditions do not yield valid probabilities because the partial derivatives are always positive in $(0, 1)$. For a monotonically increasing function, the left end points of the interval $(0, 1)$ are optimal solutions, corresponding to $p_0 = p_1 = 0$. In other words, both biased coins always yield tails, and we deterministically receive zero reward.

Considering $[f_0, f_1, f_2] = [0, 1, 0]$ for case (b), we obtain the optimality conditions $1-2p_1 = 1-2p_2 = 0$, corresponding to $p_1 = p_2 = 0.5$. The negative log-likelihood at $p_1 = p_2 = 0.5$ is also lower than at the interval end-points $p_1, p_2 \in \{0, 1\}$. Therefore, both coins are fair, and the observed data tells us that one of them turned up heads.

Considering $[f_0, f_1, f_2] = [0, 0, 1]$ for case (c), the partial derivatives are always negative in $(0, 1)$. For a monotonically decreasing function, the right end points of the interval $(0, 1)$ are optimal solutions, corresponding to $p_0 = p_1 = 1$. Therefore, both biased coins always yield heads, and we deterministically receive a reward of 4. ■

6.2.1 Comparing Likelihoods with Posteriors

The notion of maximum likelihood can be interpreted as the probability of the observed data, given the parameters. However, since we are optimizing the parameters in $\vec{\theta}$, a natural question arises as to whether we should instead maximize the *posterior* distribution $f_{\vec{\Theta}|\vec{X}_1=\vec{x}_1, \dots, \vec{X}_n=\vec{x}_n}(\vec{\theta})$, given the data. According to Bayes rule, the posterior is proportional to the product of the likelihood and the prior density function $f_{\vec{\Theta}}(\vec{\theta})$. This approach is referred to as *maximum a posteriori estimation* (MAP). One advantage of MAP estimation is that it allows the analyst to express their prior belief about how $\vec{\Theta}$ is distributed using $f_{\vec{\Theta}}(\vec{\theta})$. The simplest (and laziest) choice is to assume that $\vec{\Theta}$ is uniformly distributed over the parameter space. In such cases, MAP estimation is equivalent to maximum likelihood estimation since uniform priors do not bias the estimation towards any specific part of the parameter space. Therefore, maximum likelihood estimation is a special case of maximum a posteriori (MAP) estimation. MAP estimation is discussed in section 6.6.2.

MAP estimation is generally used in the presence of limited data, when optimizing over a large parameter space is difficult. In such cases, prior assumptions can provide helpful guidance by favoring smaller regions of the parameter space. For example, consider a case where the underlying prior assumption is that the parameters in $\vec{\Theta}$ are *concise*; in other words, values of $\|\vec{\Theta}\|^2$ that are small are preferred by the prior (as a *bias*). One way of imposing such a bias is to assume that $\vec{\Theta}$ has a prior distribution that is Gaussian with zero mean. Such an assumption causes the parameters to be biased towards the zero mean of the Gaussian, thereby resulting in a parameter vector with a smaller distance from the origin (i.e., smaller squared norm). These types of MAP methods lead to *regularized* models in machine learning. Maximum likelihood estimation is considered a *frequentist approach* to statistical learning, whereas maximum a posteriori estimation is considered a *Bayesian approach* to statistical learning.

6.3 Reconstructing Common Distributions from Data

In this section, we will discuss how to reconstruct the parameters of common families of distributions from observations with the use of maximum likelihood estimation.

6.3.1 The Uniform Distribution

As discussed in section 4.2 of Chapter 4, the uniform distribution has the following probability density function:

$$f_{X|\vec{\Theta}=[a,b]}(x) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

The density function is non-differentiable at $x = a$ and $x = b$. Furthermore, it is uninformative everywhere else with zero gradient. As a result, methods from differential calculus

cannot be used. However, since the likelihood function turns out to have a simple form, it is easy to optimize without using calculus.

Consider the case where the data points $x_1 \dots x_n$ have been observed. One can assume that a and b satisfy the following constraints:

$$\begin{aligned} a &\leq \min\{x_1, \dots, x_n\} \\ b &\geq \max\{x_1, \dots, x_n\} \end{aligned}$$

The density of each x_i is exactly $1/(b-a)$. Therefore, the corresponding likelihood function is obtained by raising the point-specific likelihood to the n th power:

$$\mathcal{L}(x_1, \dots, x_n, a, b) = \frac{1}{(b-a)^n}$$

This likelihood function is maximized when $(b-a)$ is as small as possible—therefore, a is as large as possible and b is as small as possible. Based on the aforementioned constraints, the following estimates are obtained:

$$\begin{aligned} \hat{a} &= \min\{x_1, \dots, x_n\} \\ \hat{b} &= \max\{x_1, \dots, x_n\} \end{aligned}$$

The non-smooth nature of the uniform distribution can exacerbate the effect of any mismatch of the true generating distribution from the uniform distribution. For example, consider the case where the true generating distribution is close to uniform but it also occasionally generates a few extreme-valued points. In such a case, the over-simplified uniform distribution (leveraged for modeling) will estimate the values of a and b in a grossly incorrect way by using the extreme-valued points to define these parameters. In most practical distributions like the Gaussian distribution, a few anomalies do not significantly affect the modeling as they are absorbed by the smooth tails of the distribution.

Example 6.3 Suppose that you observe the points 0.1, 1.3, 2.1, 3.5, 4.0, 5.3, 6.2, 7.8, 7.9, 9.2 that are generated by a uniform distribution. Compute the maximum likelihood estimate of the distribution parameters. Suppose a data entry error causes the entry 7.9 to be recorded as 79. Now recalculate the parameter estimates. Comment on your results.

Solution: Without the entry error, the maximum likelihood estimate of the lower and upper bounds are 0.1 and 9.2. Therefore, the data is distributed uniformly in [0.1, 9.2] according to the maximum likelihood estimate. The data error causes the upper bound to increase to 79. The newly estimated distribution range is [0.1, 79]. The results show how brittle the estimation of the uniform distribution is to errors. ■

Problem 6.1 You sample 1.7, 8.2, 3.5, 6.6 from a uniform distribution. What are the maximum likelihood estimates of the lower and upper bounds of this uniform distribution?

6.3.2 The Bernoulli Distribution

The Bernoulli distribution is one of the simplest discrete distributions, and its parameter estimation process is important because it is very similar to that of the categorical, binomial, and multinomial distributions. All these distributions are used frequently in machine

learning. The Bernoulli distribution is discussed in section 4.3 of Chapter 4. The generating process of the Bernoulli distribution can be thought of in terms of a biased coin toss, in which the side that is considered the “success” side has probability p . The success side maps to a value of 1 and the other side maps to a value of 0. The probability mass function for the Bernoulli distribution may be defined as follows:

$$p_{X|\Theta=p}(x) = \begin{cases} p & \text{if } x = 1 \\ (1-p) & \text{if } x = 0 \end{cases} = p^x(1-p)^{(1-x)}$$

In this case, maximum likelihood estimation needs to estimate the success probability p . Now consider the case where one has n observations $x_1 \dots x_n$, each of which is drawn from $\{0, 1\}$. The corresponding probability of that observation is $p^{x_i}(1-p)^{(1-x_i)}$. Therefore, the likelihood function may be expressed as follows:

$$\begin{aligned} \mathcal{L}(x_1, \dots, x_n, p) &= \prod_{i=1}^n p_{X|\Theta=p}(x_i) = \prod_{i=1}^n p^{x_i}(1-p)^{(1-x_i)} \\ &= p^{\sum_i x_i} (1-p)^{(n - \sum_i x_i)} \end{aligned}$$

The corresponding (negative) log-likelihood function is as follows:

$$\mathcal{LL}(x_1, \dots, x_n, p) = -\ln(p) \sum_{i=1}^n x_i - \ln(1-p)(n - \sum_{i=1}^n x_i)$$

On differentiating the negative log-likelihood function with respect to the parameter p and setting it to 0, one obtains the following:

$$\frac{\partial \mathcal{LL}(x_1 \dots x_n, p)}{\partial p} = -\frac{\sum_{i=1}^n x_i}{p} + \frac{n - \sum_{i=1}^n x_i}{1-p} = 0$$

On simplifying the above expression, one obtains the following estimate \hat{p} :

$$\hat{p} = \frac{\sum_{i=1}^n x_i}{n} \tag{6.3}$$

In other words, the estimated value of the Bernoulli success probability p is the fraction of successful trials. This estimation clearly makes intuitive sense. As we will see later, this general principle carries over to the geometric, binomial, categorical, and multinomial distributions, where the frequencies of alternative outcomes are counted over multiple trials.

Example 6.4 You toss a coin 20 times and it shows up as heads 8 times. What is the maximum likelihood estimate of the probability of a heads?

Solution: Using the aforementioned formula, the maximum likelihood estimate of the Bernoulli parameter for success (heads) is $8/20 = 0.4$. ■

Problem 6.2 (Categorical Distribution) Consider a categorical distribution with outcome probabilities $p_1 \dots p_d$. Samples are drawn repeatedly from the distribution. Show that the maximum likelihood estimate of p_r is the sample fraction of instances belonging to the r th outcome.

6.3.3 The Geometric Distribution

The geometric distribution (cf. section 4.5) is constructed on top of the Bernoulli distribution, in which the Bernoulli trials are performed in an independent and identically distributed manner until the outcome is a success. The total number of trials (including the final successful trial) is reported as the value of the random variable. The success probability of each Bernoulli trial is p . The probability mass function of the geometric distribution is as follows:

$$p_{X|\Theta=p}(x) = (1-p)^{(x-1)}p \quad \forall x \in \{1, 2, \dots, \infty\}$$

Now, consider the case where the data points $x_1 \dots x_n$ are observed as a result of n independent geometric processes. The corresponding likelihood function is constructed using the aforementioned PMF:

$$\begin{aligned} \mathcal{L}(x_1, \dots, x_n, p) &= \prod_{i=1}^n p_{X|\Theta=p}(x_i) = \prod_{i=1}^n (1-p)^{(x_i-1)}p \\ &= (1-p)^{(\sum_i x_i - n)} p^n \end{aligned}$$

The corresponding (negative) log-likelihood function is as follows:

$$\mathcal{LL}(x_1, \dots, x_n, p) = -\ln(1-p) \left(\sum_{i=1}^n x_i - n \right) - n \ln(p)$$

On differentiating the negative log-likelihood function with respect to parameter p and setting it to zero, one obtains the following:

$$\frac{\partial \mathcal{LL}(x_1 \dots x_n, p)}{\partial p} = \frac{\sum_{i=1}^n x_i - n}{1-p} - \frac{n}{p} = 0$$

On simplifying the above expression, one obtains the following:

$$p \left(\sum_{i=1}^n x_i \right) = n \tag{6.4}$$

Therefore, the estimated value \hat{p} of the parameter p is as follows:

$$\hat{p} = \frac{n}{\sum_{i=1}^n x_i} \tag{6.5}$$

Here, the numerator n is the number of terminations (success events), whereas the denominator is the total number of Bernoulli trials. The estimated parameter is intuitively similar to that in the Bernoulli distribution, because *it is equal to the fraction of successful Bernoulli trials.*

Example 6.5 Samples from the geometric distribution yield values of 1, 1, 1, 2, 2, 3, 3, 4, 6. Find the maximum likelihood estimate of the success parameter.

Solution: A total of $n = 9$ samples are taken, and the value of $\sum_i x_i$ is 23. Therefore, the MLE of the success parameter is $9/23$. ■

Problem 6.3 You roll a biased die until it shows a 6 and count the number of rolls (including the roll of 6). You repeat this process five times and find that you require 1, 8, 11, 5, and 9 rolls. What is the maximum likelihood estimate of the probability of getting a 6?

6.3.4 The Binomial Distribution

The binomial distribution quantifies the probability of x successes in m Bernoulli trials with probability p . Even though the binomial distribution has two parameters m and p , it is generally only the success parameter p that needs to be estimated in most application-centric settings. The probability of x successes in m trials is as follows (cf. section 4.6):

$$p_{X|\Theta=p}(x) = \binom{m}{x} p^x (1-p)^{(m-x)}$$

Now consider the case where one has a total of n different sequences of Bernoulli trials in which the numbers of successes are denoted by $x_1 \dots x_n$. Assume that the i th sequence is achieved by performing m_i trials, and therefore we are allowing for binomial distributions with varying values of the observable parameter m_i corresponding to the number of trials. The likelihood function is therefore computed as follows:

$$\begin{aligned} \mathcal{L}(x_1, \dots, x_n, p) &= \prod_{i=1}^n p_{X|\Theta=p}(x_i) = \prod_{i=1}^n \binom{m_i}{x_i} p^{x_i} (1-p)^{(m_i-x_i)} \\ &= \left[\prod_{i=1}^n \binom{m_i}{x_i} \right] \left[p^{\sum_i x_i} (1-p)^{\sum_i (m_i-x_i)} \right] \end{aligned}$$

The number of trials $m_1 \dots m_n$ are treated as known constants in the likelihood function (and not included in its argument). The only parameter of interest is the success probability p . The corresponding (negative) log-likelihood function is as follows:

$$\mathcal{LL}(x_1, \dots, x_n, p) = -\ln \left[\prod_{i=1}^n \binom{m_i}{x_i} \right] - \ln(p) \sum_{i=1}^n x_i - \ln(1-p) \sum_{i=1}^n (m_i - x_i)$$

On differentiating the negative log-likelihood function and setting it to 0, one obtains the following:

$$\frac{\partial \mathcal{LL}(x_1 \dots x_n, p)}{\partial p} = -\frac{\sum_{i=1}^n x_i}{p} + \frac{\sum_{i=1}^n (m_i - x_i)}{1-p} = 0$$

On simplifying the above expression, one obtains the following:

$$p \left(\sum_{i=1}^n m_i \right) = \sum_{i=1}^n x_i \tag{6.6}$$

Therefore, the estimated value \hat{p} of the parameter p is as follows:

$$\hat{p} = \frac{\sum_{i=1}^n x_i}{\sum_{i=1}^n m_i} \tag{6.7}$$

This parameter estimate is similar to the case of the Bernoulli and geometric distributions because it is the fraction of successful Bernoulli trials (over all binomial experiments).

Example 6.6 You toss a (possibly) biased coin 5 times and you obtain heads each time. What is the maximum likelihood estimate of the probability of heads? Is this answer reasonable?

Solution: The maximum likelihood estimate of the probability of heads is $5/5 = 1$. The answer does not seem reasonable. The reason is that the number of tosses is too few and the estimate of the probability may vary significantly if the entire experiment is repeated. This problem is that of excessive *variance* of estimation. The estimates of the probabilistic parameters (like success probability) can be stabilized only by having sufficient amount of data (tosses in this case). Alternative methods using MAP estimation are discussed in later sections. ■

6.3.5 The Multinomial Distribution

The multinomial distribution is a joint probability distribution of the frequencies of categorical outcomes (cf. section 4.7). Consider an experiment with d possible outcomes (e.g., rolling a biased die) in which the probabilities of the d outcomes are $p_1 \dots p_d$. These probabilities must sum to 1. The die is rolled a total of m times and the number of times each face shows up is counted. The frequencies of the d faces are given by $x_1 \dots x_d$, so that we have $\sum_i x_i = n$. The PMF of the multinomial distribution defines the probability of the outcome vector $\vec{x} = [x_1 \dots x_d]$ as follows:

$$p_{\vec{X}|\Theta=[p_1, \dots, p_d]}(x_1, x_2, \dots, x_d) = \frac{n!}{\prod_{i=1}^d x_i!} \prod_{i=1}^d p_i^{x_i}$$

This is a joint probability distribution and its marginal distribution on each attribute can be shown to be the binomial distribution. This particular distribution is rather important in machine learning because it is often used to model the frequencies of words in documents.

Suppose that we have observed data vectors $\vec{x}_1 \dots \vec{x}_n$, so that each \vec{x}_i is the d -dimensional numerical vector $[x_{i1}, x_{i2}, \dots, x_{id}]$. One can view each of these vectors as the frequencies of the d die faces, when the d -sided die is rolled a total of m_i times. Since the frequencies of the different faces must sum to the number of die rolls, the following constraint must hold:

$$\sum_{j=1}^d x_{ij} = m_i \quad \forall i \in \{1 \dots n\} \quad (6.8)$$

The likelihood function of the n observations is as follows:

$$\mathcal{L}(\vec{x}_1, \dots, \vec{x}_n, p_1, \dots, p_d) = \prod_{i=1}^n \left(\frac{m_i!}{\prod_{j=1}^d x_{ij}} \prod_{j=1}^d p_j^{x_{ij}} \right)$$

As in the case of the binomial distribution, the numbers of trials $m_1 \dots m_n$ are treated as constants (rather than as parameters to be optimized). The corresponding (negative) log-likelihood function is as follows:

$$\mathcal{LL}(\vec{x}_1, \dots, \vec{x}_n, p_1, \dots, p_d) = -\ln \left[\prod_{i=1}^n \frac{m_i!}{\prod_{j=1}^d x_{ij}} \right] - \sum_{i=1}^n \sum_{j=1}^d x_{ij} \ln(p_j) \quad (6.9)$$

This particular setting is a *constrained* optimization problem, where the additional constraint is $\sum_{j=1}^d p_j = 1$. This type of constraint is referred to as a *convex constraint*, and it

can be shown that using the method of *Lagrangian relaxation* results in setting the derivative of the log-likelihood function to the Lagrangian parameter α rather than 0:

$$\frac{\partial \mathcal{LL}(\vec{x}_1, \dots, \vec{x}_n, p_1, \dots, p_d)}{\partial p_j} = -\frac{\sum_{i=1}^n x_{ij}}{p_j} = \alpha \quad \forall j \in \{1 \dots d\}$$

One can use the above expression to show the following:

$$p_j \propto \sum_{i=1}^n x_{ij}$$

Since $\sum_j p_j = 1$, the estimated value of p_j is the following:

$$\hat{p}_j = \frac{\sum_{i=1}^n x_{ij}}{\sum_{j=1}^d \sum_{i=1}^n x_{ij}} = \frac{\sum_{i=1}^n x_{ij}}{\sum_{i=1}^n m_i} \quad \forall j \quad (6.10)$$

The maximum likelihood estimate of p_j is the fraction of the trials resulting in outcome j .

Problem 6.4 You roll a biased die 35 times and you obtain the frequencies 5, 8, 8, 6, 3, 5 for the six sides numbered from 1 to 6. What are the maximum likelihood estimates for the outcome probabilities of the six sides?

6.3.6 The Exponential Distribution

The exponential distribution (cf. section 4.8 of Chapter 4) has the following probability density function:

$$f_{T|\Theta=\lambda}(t) = \lambda \exp(-\lambda t)$$

The parameter λ is referred to as the arrival rate. It is natural to think of the random variable T in terms of time units. Consider the situation where n observations of the exponential variable T are t_1, t_2, \dots, t_n . Using the aforementioned probability density function, the likelihood function is as follows:

$$\mathcal{L}(t_1, \dots, t_n, \lambda) = \prod_{i=1}^n f_{T|\Theta=\lambda}(t_i) = \lambda^n \exp\left(-\lambda \sum_i t_i\right)$$

The corresponding (negative) log-likelihood function is as follows:

$$\mathcal{LL}(t_1, \dots, t_n, \lambda) = -n \cdot \ln(\lambda) + \lambda \sum_{i=1}^n t_i \quad (6.11)$$

On differentiating the negative log-likelihood function with respect to the parameter λ and setting it to zero, one obtains the following:

$$\frac{\partial \mathcal{LL}(t_1 \dots t_n, \lambda)}{\partial \lambda} = -\frac{n}{\lambda} + \sum_{i=1}^n t_i = 0$$

On simplifying the above expression, one obtains the following estimated value $\hat{\lambda}$:

$$\hat{\lambda} = \frac{n}{\sum_{i=1}^n t_i} \quad (6.12)$$

One can also write the above estimate as follows:

$$\hat{\lambda} = \frac{1}{\sum_{i=1}^n t_i/n} = 1/t_{avg} \quad (6.13)$$

Here, $t_{avg} = \sum_{i=1}^n t_i/n$ is the average of the n samples of the exponential variable. In other words, the estimated value of the arrival-rate λ is the inverse of the average arrival time (which seems intuitively reasonable).

Example 6.7 Buses arrive at a stop according to a recurring exponential process. You start waiting at 8:00 AM, and see buses at 8:07, 8:10, 8:11, and 8:13. You finally board the 8:14 bus, which arrives next. Find the probability density function of the waiting time of the first bus starting at 8 : 00.

Solution: The waits for successive buses are 7, 3, 1, 2, and 1 minutes. The average arrival time is $14/5 = 2.8$ minutes. Therefore, the MLE of the exponential arrival rate is $1/2.8$ per minute. Therefore, the exponential distribution is as follows:

$$f_{T|\Theta=1/2.8}(t) = \frac{1}{2.8} \exp(-t/2.8)$$

■

Problem 6.5 You sample from the exponential distribution 6 times and obtain successive arrival times of 6.1, 5.3, 7.2, 1.3, 9.1, and 3.5. What is the maximum likelihood estimate of the arrival-rate parameter λ of the exponential distribution?

6.3.7 The Poisson Distribution

The Poisson distribution is closely related to the exponential distribution (cf. section 4.9 of Chapter 4). The Poisson process counts the number of arrivals in a fixed time-window of one unit, where the time between successive arrivals is defined by an exponential distribution with rate parameter λ . The probability mass function quantifying the probability of x arrivals is as follows:

$$p_{X|\Theta=\lambda}(x) = \left[\frac{\lambda^x}{x!} \right] \exp(-\lambda)$$

Assume that we have n observations $x_1 \dots x_n$ corresponding to the number of arrivals for each of n experiments. The corresponding likelihood function is as follows:

$$\mathcal{L}(x_1, \dots, x_n, \lambda) = \prod_{i=1}^n p_{X|\Theta=\lambda}(x_i) = \exp(-n\lambda) \prod_{i=1}^n \left[\frac{\lambda^{x_i}}{x_i!} \right]$$

One can then derive the negative log-likelihood function as follows:

$$\mathcal{LL}(x_1, \dots, x_n, \lambda) = n\lambda - \ln(\lambda)(\sum_{i=1}^n x_i) + \sum_{i=1}^n \ln(x_i!) \quad (6.14)$$

On computing the derivative of the negative log-likelihood function and setting it to zero, one obtains the following:

$$n - \frac{\sum_{i=1}^n x_i}{\lambda} = 0$$

On re-arranging, one obtains the estimated value $\hat{\lambda}$ of the parameter λ as follows:

$$\hat{\lambda} = \frac{\sum_{i=1}^n x_i}{n}$$

In other words, the maximum likelihood estimate of the parameter λ is equal to the inverse of the average number of arrivals per unit time in a fixed time interval. This estimate is very similar to the case of the exponential distribution. However, there are subtle differences in the estimates. The Poisson process is a continuous process in which exponential arrivals repeatedly occur. In the MLE of the exponential distribution, one inverts the average waiting time of a fixed *number of arrivals* to obtain $\hat{\lambda}$, whereas in the MLE of the Poisson distribution, one waits for a fixed *amount of time* and counts the number of arrivals to obtain $\hat{\lambda}$. This difference is because the MLE of different distributions are optimized in the two cases.

Example 6.8 Buses arrive at a stop according to the exponential distribution. You start waiting at 8:00 AM, and see buses at 8:07, 8:10, 8:11, 8:13. You are about to board the 8:14 bus, which arrives next, but you change your mind and decide to wait a little more for a less crowded bus. You wait till 8:20 and no bus arrives. Discouraged, you walk away. Note that the only difference from Example 6.7 is that you do not board the 8:14 bus. Find the exponential probability density function of the waiting time (in minutes) as well as the Poisson probability mass function.

Solution: A total of 5 arrivals were recorded in 20 minutes. Therefore, the MLE estimate of the exponential arrival rate is $\lambda = 5/20 = 1/4$ per minute. Therefore, the probability density function of the exponential distribution of the arrival time (in minutes) is as follows:

$$f_{T|\Theta=\lambda}(t) = \frac{1}{4} \exp(-t/4)$$

The difference in the estimation of the arrival parameter from the case of Example 6.7 is noteworthy. This is because the estimate is done differently by waiting for a fixed time, rather than terminating the process after a fixed number of arrivals.

The number of arrivals per minute is a Poisson distribution with $\lambda = 1/4$. The probability mass function of the number of arrivals X is as follows:

$$p_{X|\Theta=1/4}(x) = \frac{1}{4^x} \frac{\exp(-x/4)}{x!}$$

■

Problem 6.6 Consider a situation where the Poisson process (with the same inter-arrival rate) is performed five times, each over a window of 1 minute. The number of arrivals over the five trials is 4, 6, 4, 7, and 6. What is the maximum likelihood estimate of the inter-arrival rate in arrivals per minute?

6.3.8 The Normal Distribution

The normal distribution (cf. section 4.10 of Chapter 4) is defined using the mean μ and the variance σ^2 as follows:

$$f_{X|\vec{\Theta}=[\mu,\sigma^2]}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (6.15)$$

The likelihood function over n observations $x_1 \dots x_n$ is obtained using the product of the point-specific density functions:

$$\mathcal{L}(x_1, \dots, x_n, \mu, \sigma) = \prod_{i=1}^n f_{X|\vec{\Theta}=[\mu, \sigma^2]}(x_i) = \frac{1}{(2\pi)^{n/2}\sigma^n} \exp\left(-\frac{\sum_i(x_i - \mu)^2}{2\sigma^2}\right)$$

The corresponding (negative) log-likelihood function is as follows:

$$\mathcal{LL}(x_1, \dots, x_n, \mu, \sigma) = \frac{n\ln(2\pi)}{2} + n \cdot \ln(\sigma) + \frac{\sum_{i=1}^n(x_i - \mu)^2}{2\sigma^2}$$

There are two parameters μ and σ that need to be optimized. Therefore, the partial derivative with respect to each of these parameters is set to 0. Setting the partial derivative with respect to μ to zero, one obtains the following:

$$\frac{\sum_{i=1}^n(x_i - \mu)}{\sigma^2} = 0$$

Therefore, we obtain the following estimated value $\hat{\mu}$ of the mean parameter:

$$\hat{\mu} = \frac{\sum_{i=1}^n x_i}{n}$$

In other words, the maximum likelihood estimate of μ is the sample average $\hat{\mu}_X$ of the different observations, which seems like a logical result. For estimating σ , the partial derivative of the negative log-likelihood function with respect to σ is set to 0:

$$\frac{n}{\sigma} - \frac{\sum_{i=1}^n(x_i - \mu)^2}{\sigma^3} = 0$$

On simplifying the above expression, one obtains the estimated value of the parameter σ :

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^n(x_i - \hat{\mu})^2}{n}$$

In other words, the maximum likelihood estimate of the variance is equal to the sample variance without Bessel correction. Therefore, the maximum likelihood estimate is *biased* in this case, although the bias is typically quite small.

Example 6.9 Consider the following mean-centered data set in two dimensions:

$$\{(4, 4), (3, 3), (2, 2), (1, 1), (-1, -1), (-2, -2), (-3, -3), (-4, -4), (1, -1), (-1, 1)\}$$

You want to model this data with a Gaussian distribution. Compute the 1-dimensional marginal distribution of the data set along each dimension.

Solution: Assume that $\vec{\Theta}_1 = [\mu_1, \sigma_1^2]$ and $\vec{\Theta}_2 = [\mu_2, \sigma_2^2]$ be the distribution parameters along the two dimensions. The 1-dimensional marginal distribution along the x_1 dimension can be computed using the mean and (Bessel uncorrected) sample variance along those dimensions, which are 0 and 6.2, respectively. Plugging these values into

the Gaussian density function of Equation 6.15, we obtain the following:

$$f_{X_1|\vec{\Theta}_1=[0,6.2]}(x_1) = \frac{1}{\sqrt{12.4\pi}} \exp\left(-\frac{x_1^2}{12.4}\right)$$

One can use a similar argument to show that the distribution along x_2 is the following:

$$f_{X_2|\vec{\Theta}_2=[0,6.2]}(x_2) = \frac{1}{\sqrt{12.4\pi}} \exp\left(-\frac{x_2^2}{12.4}\right)$$

■

Problem 6.7 Consider the data samples 1.2, 2.5, 2.5, 3.1, which are generated from a normal distribution. Derive the maximum likelihood estimate of the distribution parameters and the density function of the corresponding normal distribution.

Problem 6.8 Repeat the same problem as above, but for the data samples 1.2, 1.2, 2.5, 3.1. Compare the two sets of points and explain why you get a higher or lower likelihood fit in one case.

6.3.9 Multivariate Distributions with Dimension Independence

Most of the distributions discussed thus far are 1-dimensional distributions. However, it is common for real-world data sets to have joint distributions over multiple variables. In practice, it can be a huge challenge to estimate the probability density of distributions if the attributes are not independent. This is because such distributions can potentially have complexity that increases exponentially with the number of dimensions; in other words, the amount of information contained in the joint distribution requires space that increases exponentially with the number of dimensions. This observation implies that the number of parameters to capture a joint distribution could potentially increase exponentially with the number of dimensions, which proportionally increases the number of points needed to estimate these parameters as well. However, in many models, simplifying assumptions are made on the nature of the dependence among dimensions. This greatly simplifies the modeling and reduces the number of parameters required to capture the distribution. The simplest case is one in which the d components of the random variable vector $\vec{X} = [X_1, \dots, X_d]$ are independent of one another.

Consider the case where the i th data instance is $\vec{x}_i = [x_{i1}, \dots, x_{id}]$. Here, x_{ij} is the j th dimension of the i th data instance \vec{x}_i . The (marginal) probability density function of the i th observation along the j th dimension is denoted by $f_{X_j|\vec{\Theta}_j=\vec{\theta}_j}(x_{ij})$. Subsequently, the joint distribution $f_{\vec{X}|\vec{\Theta}=\vec{\theta}}(\vec{x}_i)$ is estimated as the product of the marginal distributions along the d dimensions:

$$f_{\vec{X}|\vec{\Theta}=\vec{\theta}}(\vec{x}_i) = \prod_{j=1}^d f_{X_j|\vec{\Theta}_j=\vec{\theta}_j}(x_{ij})$$

A key observation is that the negative log-likelihood estimate of the joint distribution is the sum of the negative log-likelihood estimates along the d dimensions *because of the product-wise form of the joint distribution* in case of dimension independence. In other words, one

can write the algebraic expression for negative log-likelihood estimation as follows:

$$-\ln(f_{\vec{X}|\vec{\Theta}=\vec{\theta}}(\vec{x}_i)) = -\ln\left(\prod_{j=1}^d f_{X_j|\vec{\Theta}_j=\vec{\theta}_j}(x_{ij})\right) = -\sum_{j=1}^d \ln(f_{X_j|\vec{\Theta}_j=\vec{\theta}_j}(x_{ij}))$$

Here, $\vec{\Theta}$ is broken up into $[\vec{\Theta}_1, \dots, \vec{\Theta}_d]$, so that $\vec{\Theta}_j$ corresponds to the parameters of the j th marginal distribution. This type of separable sum makes log-likelihood optimization very simple, because differentiating with respect the parameters of one distribution allow the parameters of the other distributions to drop out (while computing partial derivatives). Therefore, one can independently estimate the parameters of each marginal distribution using the sample statistics along that dimension. We summarize this result as follows:

Lemma 6.1 (MLE with Independent Dimensions) *One can independently estimate the parameters of each marginal distribution in maximum likelihood estimation using the sample statistics along that dimension, and the joint distribution is expressed as the product of marginal distributions.*

In the following, an example is provided for the case of the Gaussian distribution. The probability density $f_{X_j|\vec{\theta}_j=[\mu_j, \sigma_j^2]}(x_{ij})$ for the j th dimension of the i th observation has its own mean μ_j and standard deviation σ_j . Therefore, it is expressed as follows:

$$f_{X_j|\vec{\Theta}_j=[\mu_j, \sigma_j^2]}(x_{ij}) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(x_{ij} - \mu_j)^2}{2\sigma_j^2}\right) \quad (6.16)$$

On differentiating the negative log-likelihood and setting it to zero, one again obtains a similar form of the maximum likelihood estimate, wherein the parameter μ_j is estimated as the sample mean along the j th dimension, and the parameter σ_j is estimated as the sample standard deviation along the j th dimension (without Bessel's correction).

Example 6.10 *For the case of Example 6.9, use MLE to find the joint distribution with independent dimensions.*

Solution: The joint distribution is the product of the marginal distributions already derived in Example 6.9. This is because of the dimension independence assumption. The two means are 0 and the two variances are each 6.2. The corresponding probability density function with parameter vector $[0, 0, 6.2, 6.2]$ is as follows:

$$f_{\vec{X}|\vec{\Theta}=[0,0,6.2,6.2]}(x_1, x_2) = \frac{1}{12.4\pi} \exp\left(-\frac{x_1^2 + x_2^2}{12.4}\right)$$

Problem 6.9 *Consider the following sample of three points $(1, 2)$, $(2, 3)$, and $(5, 3)$. Assume that the dimensions are independent of one another. Derive the joint distributions under the assumptions of geometric, Poisson, and Gaussian random variables.*

6.3.10 Gaussian Distribution with Dimension Dependence

The Gaussian distribution is one of the simpler cases involving dimension dependence because the entire distribution can be captured using n means and $\binom{n}{2}$ covariances. Therefore,

the multivariate Gaussian only captures second-order information about the relations among attributes, which is much simpler than the (potentially) exponential complexity of joint distributions. The probability density function of the multivariate Gaussian for the row vector $\vec{x} = [x_1, \dots, x_d]$ is as follows:

$$f_{\vec{X}|\vec{\Theta}=[\vec{\mu}, C]}(\vec{x}) = \frac{1}{(2\pi)^{d/2}|C|^{1/2}} \exp\left(-\frac{[\vec{x} - \vec{\mu}]C^{-1}[\vec{x} - \vec{\mu}]^T}{2}\right)$$

Here, $\vec{\mu}$ is the d -dimensional row vector of means, and C is the $d \times d$ covariance matrix with determinant $|C|$. We have n vector observations denoted by $\vec{x}_1 \dots \vec{x}_n$ from which the parameters need to be estimated. The negative log-likelihood function is as follows:

$$\begin{aligned} \mathcal{LL}(\vec{x}^1, \dots, \vec{x}_n, \vec{\mu}, C) &= -\sum_{i=1}^n \ln(f_{\vec{X}|\vec{\theta}=[\vec{\mu}, C]}(\vec{x}_i)) \\ &= \frac{n}{2} \ln(2\pi) + \frac{n}{2} \ln(|C|) + \frac{1}{2} \sum_{i=1}^n [\vec{x}_i - \vec{\mu}]C^{-1}[\vec{x}_i - \vec{\mu}]^T \end{aligned}$$

On computing the vector derivative of the log-likelihood function with respect to the vector $\vec{\mu}$ and setting it to the zero vector, one obtains the following optimality condition using matrix calculus:

$$\left[\sum_{i=1}^n [\vec{x}_i - \vec{\mu}] \right] C^{-1} = \vec{0}$$

The above vector derivative is expressed in denominator layout, and therefore the result is a row vector. Here, $\vec{0}$ is a d -dimensional row vector of zeros. Post-multiplying both sides with C and simplifying, one obtains the following:

$$\hat{\vec{\mu}} = \frac{\sum_{i=1}^n \vec{x}_i}{n}$$

Therefore, the maximum likelihood estimate of the mean is the sample estimate of the mean. It remains to compute the covariance matrix. It is more challenging to compute the derivatives of the negative log-likelihood with respect to the entries of C because of the presence of the determinant $|C|$ in the log-likelihood expression. We will omit these details (which require more complex ideas from matrix calculus) and state that the resulting maximum likelihood estimate of the covariance matrix is simply the sample covariance matrix (without Bessel's correction):

$$\hat{C} = \frac{\sum_{i=1}^n [\vec{x}_i - \hat{\vec{\mu}}]^T [\vec{x}_i - \hat{\vec{\mu}}]}{n}$$

Note that \vec{x}_i and $\vec{\mu}$ are row vectors. Therefore, the expression on the right multiplies d -dimensional column vectors with d -dimensional row vectors to create $d \times d$ matrices, which are then averaged to create the estimate of the covariance matrix. Note that the value of the mean vector $\vec{\mu}$ used in the above expression is the sample mean $\hat{\vec{\mu}}$ (which is also its maximum likelihood estimate).

Example 6.11 We revisit the mean-centered data set in Examples 6.9 and 6.10, which is the following:

$$\{(4, 4), (3, 3), (2, 2), (1, 1), (-1, -1), (-2, -2), (-3, -3), (-4, -4), (1, -1), (-1, 1)\}$$

Model this data with a general Gaussian distribution in two dimensions. In other words, it is not assumed that the dimensions are independent, as in Example 6.10.

Solution: The covariance matrix of this data set and its inverse can be shown to be the following:

$$\hat{C} = \begin{bmatrix} 6.2 & 5.8 \\ 5.8 & 6.2 \end{bmatrix} \quad C^{-1} = \frac{1}{24} \begin{bmatrix} 31 & -29 \\ -29 & 31 \end{bmatrix}$$

The density function of the zero-centered Gaussian distribution can be expressed in terms of $\vec{x} = [x_1, x_2]$ and the covariance matrix as follows:

$$f_{\vec{X}|\vec{\Theta}=[\vec{0}, \hat{C}]}(x_1, x_2) = \frac{1}{2\pi|\hat{C}|^{0.5}} \exp\left(-\frac{(\vec{x} - \vec{0})\hat{C}^{-1}(\vec{x} - \vec{0})^T}{2}\right)$$

The determinant $|\hat{C}|$ of the covariance matrix is $24/5 = 4.8$. On plugging in the values of C^{-1} and $|C|$, one obtains the following Gaussian distribution:

$$f_{\vec{X}|\vec{\Theta}=[\vec{0}, \hat{C}]}(x_1, x_2) = \frac{1}{2\pi\sqrt{4.8}} \exp\left(-\frac{31x_1^2 - 58x_1x_2 + 31x_2^2}{48}\right)$$

Note that the term in x_1x_2 is the interaction term, which gives the Gaussian distribution its correlated behavior. This Gaussian distribution is more refined than the one obtained in Example 6.10 using the assumption of dimension independence. ■

Problem 6.10 Consider a data set in two dimensions with the following points:

$$\{(5, -2), (4, -1), (3, 0), (2, 1), (0, 3), (-1, 4), (-2, 5), (-3, 6), (2, 3), (0, 1)\}$$

This data set is not mean centered. Model this data with a general Gaussian distribution in two dimensions.

Problem 6.11 Suppose that the three samples in Problem 6.9 are modeled with a multivariate Gaussian distribution. Derive the density function of this distribution with maximum likelihood estimation.

6.4 Mixture of Distributions: The EM Algorithm

The reconstruction methods discussed thus far have been rather simplistic in the sense that they assume that the entire data set is generated from a distribution described by a single closed-form probability mass (or density) function. As discussed in section 4.13, mixtures are natural in settings where multiple generating processes create the data depending on the case at hand. For example, the heights of males and females are naturally generated from different distributions (cf. Figure 6.1). One can also envisage a similar situation for multidimensional data (cf. Figure 6.4(a)), wherein different clusters are created by different

generating processes. The corresponding probability density is illustrated in Figure 6.4(b). The probability distribution associated with each mixture component is referred to as its conditional distribution. The set of points generated by any of these conditional distributions is referred to as a *mixture component*. The term “mixture component” is loosely referred to as not only the points generated by the conditional distribution but also the probabilistic model/parameters associated with it. In supervised applications, the data points are tagged with the identifier of the mixture component generating them, whereas in unsupervised applications the data points are not tagged with these identifiers. When the data points are not tagged with their component identifiers, the reconstruction problem becomes more difficult because one cannot deterministically identify membership of observed data to mixture components. For example, if one is given untagged data corresponding to human heights (cf. Figure 6.1), a height of 5.5 feet could either be male or female with roughly similar probability. This uncertainty makes parameter estimation of the densities of individual mixture components more difficult.

In the mixture model, it is assumed that the data are created by k different generating processes, each of which has its own distribution (and all distributions are *typically* from the same family of probability distributions but with different parameters). For example, a common setting for numeric multidimensional data is to assume that each of the k mixture components is a different Gaussian distribution. The data for human heights (cf. Figure 6.1) is generated using this approach in one dimension. The different generating processes may have different relative frequencies in terms of contributing to the observed data. In the case of human heights, the two mixture components may have equal probability, but this may not be case in another application. The choices of the number of mixture components and the distribution family corresponding to each component are made by the analyst (and therefore require practical skill and application-specific insight). For example, choosing an exponential distribution to model male or female human height will inevitably lead to poor results.

After the analyst has modeled the number and type of mixture components, the parameters of the mixture model need to be estimated in a data-driven manner. The parameters of the mixture model include (a) the relative frequency of each mixture component in generating points, and (b) the parameters of each distribution corresponding to a mixture component. The estimation process is more challenging than in the cases discussed thus far (with a single closed-form distribution), because one only has access to the observed data but it is unknown which points are generating by which mixture component. For example, which of the components generated a height of 5.5 feet (cf. Figure 6.1)? Another example is the case of Figure 6.4(a), in which each cluster corresponds to a mixture component. The points in the overlapping regions of different clusters of Figure 6.4(a) could easily have been created by any of the overlapping mixture components. In other words, *membership of observed data points to different mixture components can only be probabilistically estimated because each point typically has a nonzero probability of being generated by each mixture component*.

This situation leads to a circular inter-dependency between the estimation of mixture component parameters and membership estimation:

1. If we knew the parameters of the probability distributions of the different mixture components and the relative frequencies of mixture components (prior probabilities of point membership), we could use them to estimate the posterior probabilities of membership of points in different mixture components. This is done using the *Bayes rule*.

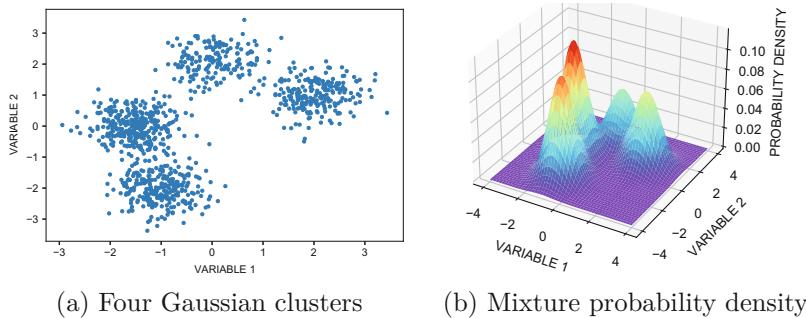


Figure 6.4: The probability density of a mixture of four Gaussian clusters

2. If we knew the posterior probabilities of membership of points in different mixture components, we could “cut up” the point and assign a point with a fractional weight to each mixture component. Then, we could use the single-distribution-based maximum likelihood estimation with each mixture component independently to determine the optimal values of the parameters.

This interdependency between point membership and parameter estimation suggests an iterative approach, which starts by assigning random values of the parameters and then repeating the following steps:

- *Expectation step:* Compute the (posterior) probability of membership of each observed data point to each mixture component, assuming that the current parameters of different distributions are optimal. Thus, the *expected* membership is computed in this step. The Bayes rule is used to achieve this goal.
- *Maximization step:* Assuming that the current probabilistic assignments of points to mixture components are correct, divvy up each point into different mixture components with weight equal to assignment probability. Perform maximum likelihood estimation independently for each mixture component with its own set of weighted points (to derive the parameters for the different conditional distributions). Although the derivations of maximum likelihood estimation seen thus far do not use fractional weights, the extension to this case is straightforward.

The expectation step is also referred to as the *E-step* and the maximization step is also referred to as the *M-step*. These two steps are iterated to convergence. The above description already provides an informal description of the *expectation-maximization algorithm*, which is also referred to as the *EM-algorithm*. Therefore, the parameters of the generative process and membership probabilities are refined iteratively. Next, we will provide a more formal description with notations. We first recap the generating process of data created from a mixture distribution given in section 4.13 of Chapter 4.

We assume that the different mixture components are denoted by $\mathcal{G}_1, \dots, \mathcal{G}_k$, with corresponding conditional probability distributions denoted by $f_{\vec{X}|\mathcal{G}_1}(\vec{x}), f_{\vec{X}|\mathcal{G}_2}(\vec{x}), \dots, f_{\vec{X}|\mathcal{G}_k}(\vec{x})$. Therefore, there are a total of k mixture components. In the case when the data are discrete, one can use probability mass functions $p_{\vec{X}|\mathcal{G}_1}(\vec{x}), p_{\vec{X}|\mathcal{G}_2}(\vec{x}), \dots, p_{\vec{X}|\mathcal{G}_k}(\vec{x})$ instead of probability density functions. For greater generality, we will work with probability density functions rather than probability mass functions. The set of parameters associated with the density function $f_{\vec{X}|\mathcal{G}_j}(\vec{x})$ is denoted by the vector $\vec{\theta}_j$. For some distributions like the Bernoulli

distribution in a single dimension, this vector could contain only one element (Bernoulli success probability parameter), which makes it a scalar. Note that the value of k is 2 in the example of data containing male and female heights (cf. Figure 6.1), which refers to the two distinct generative processes for male and female heights; the corresponding distributions $f_{X|\mathcal{G}_1}(x)$ and $f_{X|\mathcal{G}_2}(x)$ are univariate normal distributions and $\vec{\theta}_j$ is the parameter set for the j th normal distribution containing its mean and variance. The probability that the j th generative process is selected for a particular point is given by $\alpha_j = P(\mathcal{G}_j)$. Therefore, we have:

$$\sum_{j=1}^k P(\mathcal{G}_j) = \sum_{j=1}^k \alpha_j = 1$$

The probability $P(\mathcal{G}_j)$ is the *prior* probability of an observed data point belonging to mixture component j , because it refers to the probability that we would predict an observation to belong to mixture component j without knowing anything about the data point. This prior probability simply reflects the relative frequencies of the different mixture components. In the example where the observed data contains heights of males and females, both prior probabilities are 0.5, since males and females are presumed to have equal frequency in the population. However, in mixture modeling, this fact is not known up front and therefore the vector of prior probabilities is treated as the k -dimensional parameter vector $\vec{\alpha} = [\alpha_1, \dots, \alpha_k]$, which needs to be estimated during the execution of the EM-algorithm. The basic generative process for a single observation in the data is as follows:

1. Roll a biased die whose k sides have probabilities $P(\mathcal{G}_1) \dots P(\mathcal{G}_k)$. Let the outcome of the die roll be the side j , which provides the identity of the mixture component from which the observation is generated.
2. Sample the data point \vec{x} from the probability distribution $f_{\vec{X}|\mathcal{G}_j}(\vec{x})$. The point \vec{x} is the output of one iteration of the generative model.

The aforementioned generative process is repeated to create a sample of observations, which reflects the observed data set. Since it is assumed that the data set contains n points, the above generative process is applied n times.

The unconditional density function for the generated data is given by the total probability rule of Chapter 3:

$$f_{\vec{X}}(\vec{x}) = \sum_{j=1}^k P(\mathcal{G}_j) f_{\vec{X}|\mathcal{G}_j}(\vec{x})$$

The negative log-likelihood function is therefore given by the following:

$$\mathcal{LL}(\vec{x}_1, \dots, \vec{x}_n, \vec{\theta}_1, \dots, \vec{\theta}_k, \vec{\alpha}) = - \sum_{i=1}^n \ln \left(\sum_{j=1}^k \alpha_j f_{\vec{X}|\mathcal{G}_j}(\vec{x}_i) \right)$$

Differentiating the negative log-likelihood function with respect to $\vec{\theta}_r$ in vector-calculus notation and setting to zero, one obtains the following (while recognizing that only the density function for the j th mixture component depends on $\vec{\theta}_j$):

$$\sum_{i=1}^n \underbrace{\frac{\alpha_j f_{\vec{X}|\mathcal{G}_j}(\vec{x}_i)}{\sum_{r=1}^k \alpha_r f_{\vec{X}|\mathcal{G}_r}(\vec{x}_i)}}_{\text{Bayes form}} \frac{1}{f_{\vec{X}|\mathcal{G}_j}(\vec{x}_i)} \frac{\partial f_{\vec{X}|\mathcal{G}_j}(\vec{x}_i)}{\partial \vec{\theta}_j} = \vec{0}$$

Here, a key observation is that one part of this expression is simply the posterior probability $\gamma_{ji} = P(\mathcal{G}_j | \vec{x}_i)$ based on the Bayes rule. Therefore, one can rewrite the above expression as follows:

$$\sum_{i=1}^n \gamma_{ji} \frac{1}{f_{\vec{X}|\mathcal{G}_j}(\vec{x}_i)} \frac{\partial f_{\vec{X}|\mathcal{G}_j}(\vec{x}_i)}{\partial \theta_j} = \vec{0} \quad (6.17)$$

Note that this form of the optimality condition is a generalization of the condition for optimality with a single mixture distribution (cf. Equation 6.2); in this case, the j th mixture distribution masquerades as a single unconditional distribution, but the i th point has a weight equal to that of its posterior probability. Of course, since the posterior probability also depends on the distribution parameters, one cannot solve the above equation in closed form. The EM algorithm gets around this problem with an iterative procedure of fixing the weights (posterior probabilities γ_{ji}) using the parameters in the last iteration and then using these posterior probabilities in the above equation to recompute the optimal values of these parameters in closed form. This provides the mathematical justification of why individual points are replaced by weighted points in the EM algorithm. The parameter estimation procedure for each mixture component in the maximization step is independent and boils down to a weighted version of the parameter estimation process for a single distribution.

Next, we describe the expectation and maximization steps formally with notations. The purpose of the expectation step is to compute the posterior probability γ_{ji} of the membership of point i in component j by using the Bayes rule. The maximization component then uses γ_{ji} as weights to recompute all parameters with maximum likelihood estimation, including the prior probabilities and the parameters of mixture components. The algorithm starts with setting the parameters to random values and then iterates through the following steps:

- *Expectation step:* Compute the probability of membership of each observed data point to each mixture component using the Bayes rule:

$$\gamma_{ji} = P(\mathcal{G}_j | \vec{x}_i) = \frac{P(\mathcal{G}_j) \cdot f_{\vec{X}|\mathcal{G}_j}(\vec{x}_i)}{\sum_{r=1}^k P(\mathcal{G}_r) \cdot f_{\vec{X}|\mathcal{G}_r}(\vec{x}_i)} \quad \forall i, j$$

Recall that the posterior probability of membership of point i to mixture component j is denoted concisely by γ_{ji} .

- *Maximization step:* First, note that the prior probability of each cluster is a parameter of a categorical distribution. As in the case of the Bernoulli distribution, the maximum likelihood estimate of the probability of each category is derived as the fraction of points assigned to mixture component j . Since points have fractional memberships to different mixture components, this probability is computed as follows:

$$\alpha_j = P(\mathcal{G}_j) = \frac{\sum_{i=1}^n \gamma_{ji}}{n} \quad \forall j$$

Next, we discuss the derivation of the parameters of the different mixture components. Each mixture component is assumed to contain the all the n points in the data set, albeit with different fractional weights that sum to 1. The optimality condition for the i th mixture component is contained in Equation 6.17, which is simply a posterior-weighted version of the optimality condition for a data set generated only by a single mixture component. The key point is that the expectation step *has helped decompose the problem into k independent problems*, one for each mixture component. Depending

Table 6.1: The posterior-weighted parameter estimations of mixture component j . The notations such as x_i , m_i , and n are the same as used in earlier sections on estimations of individual distributions.

Distribution	MLE from section 6.3	MLE for mixture component j
Bernoulli	$\hat{p} = (\sum_{i=1}^n x_i) / n$	$(\sum_{i=1}^n \gamma_{ji} x_i) / (\sum_{i=1}^n \gamma_{ji})$
Geometric	$\hat{p} = n / \sum_{i=1}^n x_i$	$(\sum_{i=1}^n \gamma_{ji}) / (\sum_{i=1}^n \gamma_{ji} x_i)$
Binomial	$\hat{p} = (\sum_{i=1}^n x_i) / (\sum_{i=1}^n m_i)$	$(\sum_{i=1}^n \gamma_{ji} x_i) / (\sum_{i=1}^n \gamma_{ji} m_i)$
Multinomial	$\hat{p}_r = (\sum_{i=1}^n x_{ir}) / (\sum_{i=1}^n m_i)$	$(\sum_{i=1}^n \gamma_{jix_{ir}}) / (\sum_{i=1}^n \gamma_{jim_i})$
Exponential	$\hat{\lambda} = n / (\sum_{i=1}^n t_i)$	$(\sum_{i=1}^n \gamma_{ji}) / (\sum_{i=1}^n \gamma_{jiti})$
Poisson	$\hat{\lambda} = (\sum_{i=1}^n x_i) / n$	$(\sum_{i=1}^n \gamma_{ji} x_i) / (\sum_{i=1}^n \gamma_{ji})$
Normal (1-dim.)	$\hat{\mu} = \sum_{i=1}^n x_i / n$ $\hat{\sigma}^2 = \sum_{i=1}^n (x_i - \hat{\mu})^2 / n$	$(\sum_{i=1}^n \gamma_{ji} x_i) / (\sum_{i=1}^n \gamma_{ji})$ $(\sum_{i=1}^n \gamma_{ji} (x_i - \hat{\mu})^2) / (\sum_{i=1}^n \gamma_{ji})$
Gaussian (dep. attrib.)	$\hat{\mu} = \frac{\sum_{i=1}^n \vec{x}_i}{n}$ $\hat{C} = \frac{\sum_{i=1}^n [\vec{x}_i - \hat{\mu}]^T [\vec{x}_i - \hat{\mu}]}{n}$	$\frac{\sum_{i=1}^n \gamma_{ji} \vec{x}_i}{\sum_{i=1}^n \gamma_{ji}}$ $\frac{\sum_{i=1}^n \gamma_{ji} [\vec{x}_i - \hat{\mu}]^T [\vec{x}_i - \hat{\mu}]}{\sum_{i=1}^n \gamma_{ji}}$

on the specific choice of the distribution selected by the analyst for each mixture component, any of the maximum likelihood estimation methods discussed in Table 6.1 can be used. Note that the parameter estimations in Table 6.1 are weighted versions of the maximum likelihood estimations discussed earlier in this chapter; the weights are simply the fractional memberships γ_{ji} of points.

As a specific example of how the expectation-maximization algorithm might work in a real-world setting, consider the case where you were given a 1-dimensional data set of heights of men and women in the US population (cf. Figure 6.1), and you wanted to create a mixture model with two Gaussian components. The rows of the data set do not contain any information about the biological sex. Assume that you don't know the parameters of these two Gaussian distributions or the relative number of items in the population belonging to these two components. In such a case, the EM-algorithm can be used to estimate the mixture-component distribution parameters, the relative frequencies of the points in the mixture components, and the probabilities of points belonging to the two components. Note that some insight would still be needed from the analyst up front to "guess" the fact that it is reasonable to model this distribution with two Gaussian components. In this context, the analyst may find it helpful to visualize a histogram of the data set, which provides helpful hints about the number and shape of the underlying mixture components. Indeed exploratory data analysis at the beginning of the process may often help make wise choices that significantly improve the underlying models.

Specific examples of the maximization step with different types of mixture components (along with the details of the descriptions of the corresponding EM algorithms) are also provided in section 9.2 of Chapter 9. Note that the algorithms in section 9.2 are relatively straightforward applications of the procedure discussed in this section, and even the maximization procedure is based on formulas in Table 6.1.

Example 6.12 You have a data set of seven heights of gorillas, which are 3.9, 4.0, 4.1, 5.0, 6.0, 6.5, 7.0 ft in ascending order. You model the heights with two mix-

ture components $\mathcal{G}_1, \mathcal{G}_2$ with normal distributions, where your assumption is that \mathcal{G}_1 corresponds to the female component. You initialize prior probabilities to 0.5, both standard deviations to 1 foot, and the means of females and males to $\mu_1 = 4.8$ feet and $\mu_2 = 6.2$ feet, respectively. It can be shown that an iteration of the E-step yields (female component) posterior probabilities $\gamma_{1i} = P(\mathcal{G}_1|x_i)$ to be 0.90, 0.89, 0.88, 0.67, 0.33, 0.20, and 0.11, respectively.

- Estimate the prior probabilities and mixture distribution parameters in the first M-step. Use the appropriate formulas from Table 6.1.
- Recompute the posterior probability $P(\mathcal{G}_1|x_4)$ of the height $x_4 = 5.0$ ft belonging to the first (female) mixture component in the next E-step.

Solution:

- The prior probability α_1 of females is $\alpha_1 = \sum_i \gamma_{1i}/n = (0.9+0.89+0.88+0.67+0.33+0.20+0.11)/7 = 0.57$. The prior probability α_2 of males is therefore $1 - 0.57 = 0.43$. The new estimation of the (female component) normal distribution parameter $\hat{\mu}_1 = \sum_i \gamma_{1i}x_i / \sum_i \gamma_{1i}$ is as follows:

$$\hat{\mu}_1 = \frac{3.9(0.9) + 4.0(0.89) + 4.1(0.88) + 5.0(0.67) + 6.0(0.33) + 6.5(0.2) + 7.0(0.11)}{0.90 + 0.89 + 0.88 + 0.67 + 0.33 + 0.20 + 0.11}$$

The above value of $\hat{\mu}_1$ evaluates to 4.54 feet. Using a similar approach, the mean height $\hat{\mu}_2$ of the normal distribution for males is estimated to be 6.1 feet. Next, the normal distribution standard deviation parameter $\hat{\sigma}_1$ of the female component is computed as follows:

$$\hat{\sigma}_1^2 = \frac{\sum_i \gamma_{1i}(x_i - 4.8)^2}{\sum_i \gamma_{1i}} \approx 0.92^2$$

Note that the value of $\hat{\mu}_1 = 4.8$ from the previous iteration is used in the above estimation (rather than the latest value of 4.54) because parameters from the previous iteration are consistently used. Making the same calculation for the male component, the estimated value of $\hat{\sigma}_2$ is as follows:

$$\hat{\sigma}_2^2 = \frac{\sum_i \gamma_{2i}(x_i - 6.2)^2}{\sum_i \gamma_{2i}} \approx 0.95^2$$

- In the next iteration of the E-step, the posterior probability of the height of $x_4 = 5.0$ feet being female is as follows:

$$\gamma_{14} = \frac{0.53 \frac{1}{0.92} \exp\left(-\frac{(5.0-4.54)^2}{2*0.92^2}\right)}{0.53 \frac{1}{0.92} \exp\left(-\frac{(5.0-4.54)^2}{2*0.92^2}\right) + 0.47 \frac{1}{0.95} \exp\left(-\frac{(5.0-6.1)^2}{2*0.95^2}\right)} \approx 0.73$$

The factor $1/\sqrt{2\pi}$ in the density function of the normal distribution was ignored in both the numerator and denominator. The corresponding posterior probability for the second (male) component is $1 - 0.73 = 0.27$.

■

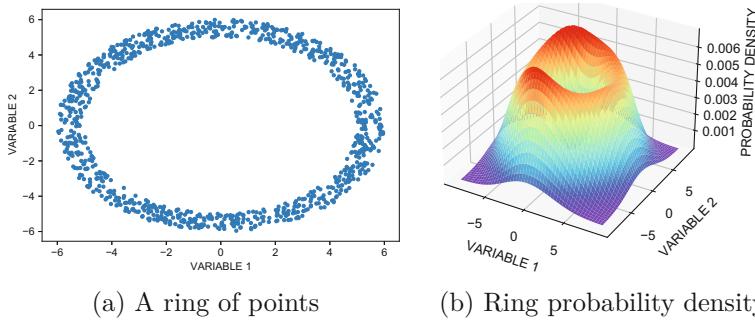


Figure 6.5: An example of a data set whose probability density is hard to capture with parametric reconstruction with maximum likelihood estimation. The density in (b) was generated using kernel density estimation.

6.5 Kernel Density Estimation

All the methods discussed thus far reconstruct distributions by assuming that the distribution belongs to a particular family of models, which pre-defines the shape (e.g., bell curve) of the distribution based on analyst insight; subsequently, the parameters define the details of the particular distribution at hand (e.g., mean and variance of the normal distribution). However, by selecting a particular distribution *a priori*, there is a risk that its shape may not be consistent with the observed data. For example, if one chooses the normal distribution to represent a particular data set, but the histogram of the data set shows a bimodal distribution (cf. Figure 6.3(b)), the resulting distribution will fit the data very poorly. The main problem with these types of parametric approaches is that they depend heavily on the analyst to choose the correct family of distributions up front. Even when a mixture model is used, the analyst may sometimes make errors in choosing the correct number of mixture components or the specific family of distributions that are relevant for the mixture components. If the analyst does not have a good idea of the distribution shape up front and chooses the incorrect family of distributions, both the quality of fit and the prediction accuracies of downstream applications will be poor. It is possible for some real-world distributions to not conform to the shapes of popularly used distributions. For example, consider the case of a data set containing a ring of points. An appropriate probability distribution for the data set is illustrated in Figure 6.5. This type of distribution cannot be generated easily by using either the individual models discussed earlier in this chapter or even by using a mixture of these distributions. Kernel density estimation is an appropriate approach to generate such distributions (and is in fact how the visualization in Figure 6.5(b) was generated).

Kernel density estimation is a *non-parametric method* for reconstructing distributions, and it has the ability to reconstruct distributions of arbitrary shape from the underlying data without making any prior assumptions. The basic idea of kernel density estimation is to replace each point in the data with a smooth “bump,” which is referred to as a *kernel*. The data point serves as the mean of this kernel. A typical example is the Gaussian kernel with independent attributes and standard-deviation h_i along the i th dimension. In other words, a kernel is a point-specific Gaussian probability density function with mean centered at that point and dimension-specific standard deviation defined by the bandwidth h_i . Although the Gaussian kernel is the most popular choice, other forms of the kernel provide results of high quality. Therefore, each point in the data set contributes to the density of every point in the

space, albeit with a contribution decreasing with distance from the point. Consider the case where the set of observations in a data set are given by $\vec{x}_1 \dots \vec{x}_n$, where $\vec{x}_i = [x_{i1}, \dots, x_{id}]$. The contribution of \vec{x}_i to a probability density estimation at \vec{z} is given by the following:

$$G(\vec{z}, \vec{x}_i) = G(z_1, \dots, z_d, x_{i1}, \dots, x_{id}) = \prod_{j=1}^d \frac{1}{\sqrt{2\pi}h_j} \exp\left(-\frac{(z_j - x_{ij})^2}{2 \cdot h_j^2}\right)$$

Then, the probability density $f_{\vec{X}}(\vec{z})$ at \vec{z} is estimated as follows:

$$\hat{f}_{\vec{X}}(\vec{z}) = \frac{1}{n} \sum_{i=1}^n G(\vec{z}, \vec{x}_i) \quad (6.18)$$

The bandwidth h_j controls the smoothness of the distribution for the j th dimension. Choosing a bandwidth that is excessively large would lead to a similar result for almost any data set, which is a very gently sloping hill with its peak near the center of the data set. Choosing an extremely small bandwidth will lead to a density function that contains a bunch of spikes and is not very smooth. The best choice of the bandwidth is given by Silverman's approximation rule, which uses the dimension-specific standard deviation σ_j and the number of points n in the data set:

$$h_j = \frac{1.06\sigma_j}{n^{1/5}}$$

Note that the bandwidth reduces with an increasing number of points, because the nearest-neighbor distances between points tend to reduce. Therefore, the "bumps" from closely spaced points tend to merge at smaller bandwidths (which is what we want). This choice tends to provide excellent and smooth approximations of the probability density function. It turns out that other choices of the kernel, such as a cone, can also provide good approximations, although the Gaussian kernel tends to provide the best results in practice. It can be shown that the kernel density estimate converges to the true density, as the number of points approaches infinity.

The probability density of the ring of points in Figure 6.5(a) is reconstructed in Figure 6.5(b) with kernel density estimation. Note the complex volcano-like reconstruction of the probability density, which cannot be easily represented even with a mixture of standard distributions. Furthermore, one can also generate the probability density function of a mixture of distributions (cf. Figure 6.4(b)) with kernel density estimation without making any prior assumptions on the number of clusters. The prior assumptions required by mixture models can sometimes serve as a strait-jacket that prevents fits of high quality.

It is noteworthy that *kernel density estimations are closely related to histograms in the sense that they aggregate frequencies, albeit in a smoothed way with kernels*. Like kernel-density estimates, histograms often show the true distribution structure of the data; the main difference is that histograms summarize the data up-front with bins, whereas kernel-density estimates need to be computed at a specific point with aggregation-based computations. In a histogram, data points contribute fixed amounts to the densities of bins, whereas data points contribute varying amounts in kernel density estimation (depending on the choice of kernel). Therefore, unlike the step-wise histograms, kernel density estimation provides a smooth density function even when the number of data points is limited.

The advantages of kernel density estimation come with some computational costs. The complexity of data distributions can increase exponentially with increasing data dimensionality. Therefore, the number of points required for accurate estimation increases exponentially as well. This type of complexity would be shown even by a uniform distribution,

where an exponentially increasing number of points would be required to avoid spurious peaks and valleys. In practice, however, since the different attributes are correlated with one another, there are clear patterns in the data that are reinforced by far fewer points. Therefore, for most real-world distributions, high-quality results can be achieved with far fewer points. The most attractive aspect of kernel density estimation is that it makes no prior assumption about the shape of the underlying data distribution, and is usually able to provide a density estimate that reflects the shape of the true distribution quite well.

Although it is not widely used, kernel-density estimation can be used in probabilistic classifiers (like the Bayes classifier) and in clustering. However, its use is not very widespread largely because of the high computational complexity associated with density estimation (at prediction time). The parametric methods for probability density estimation front-load the work involved in the maximum likelihood estimation process. Once the parameters have been learned, the (closed-form) probability density can be computed very rapidly in constant time. On the other hand, the kernel density method does not require any estimation up front, but when the density of a specific point is needed during prediction, it requires time that is proportional to the number of points in the data (cf. Equation 6.18). This property often makes kernel density estimation harder to use, because prediction settings tend to be more time critical. This is because the estimation of parameters is often done offline by an analyst in a preprocessing phase, whereas prediction algorithms are often leveraged by decision-makers and end-users in response to specific needs. As a result, slow responses are more likely to be frowned upon than slow training. Therefore, kernel density estimates are not used very frequently in clustering and classification (although their potential is very significant). A number of techniques exist for improving the speed of kernel density estimation. Pointers are provided in the bibliographic notes.

6.6 Reducing Reconstruction Variance

One of the challenges with reconstructing distributions is that the reconstruction heavily depends on the sample at hand, and there may be wide variability between one sample and the next. When the sample sizes are small the reconstructed distributions may depend so heavily on the random nuances of the underlying sample that the reconstructed distribution from a particular sample may no longer faithfully represent the distribution of the underlying *population*. Furthermore, changing the sample might drastically change the reconstructed distribution. In other words, there is a *variance* in the reconstructed distribution, which depends heavily on the (unimportant) vagaries of the specific sample at hand. This is not desirable for downstream applications like classification, which start *overfitting* to unimportant artifacts of the observed data. As a result, the same instance may get predicted differently when different samples are used for building models. This type of inconsistency reduces the accuracy of prediction.

In order to understand this point, we create two different samples of data from a mixture model of four Gaussian clusters. The scatter plot of a large sample of 1000 points is illustrated in Figure 6.6(a), whereas the scatter plot of a sample of 20 points from exactly the same distribution is illustrated in Figure 6.6(c). The corresponding reconstructed probability density functions are illustrated in Figures 6.6(b) and (d), respectively. It is evident that the two reconstructed distributions are different even in terms of the relative frequencies of points in clusters (which is reflected in the varying patterns of relative heights of the probability density profiles in the two cases). A key point is that the sample of 20 points could easily correspond to frequencies that do not reflect the true distribution. Furthermore,

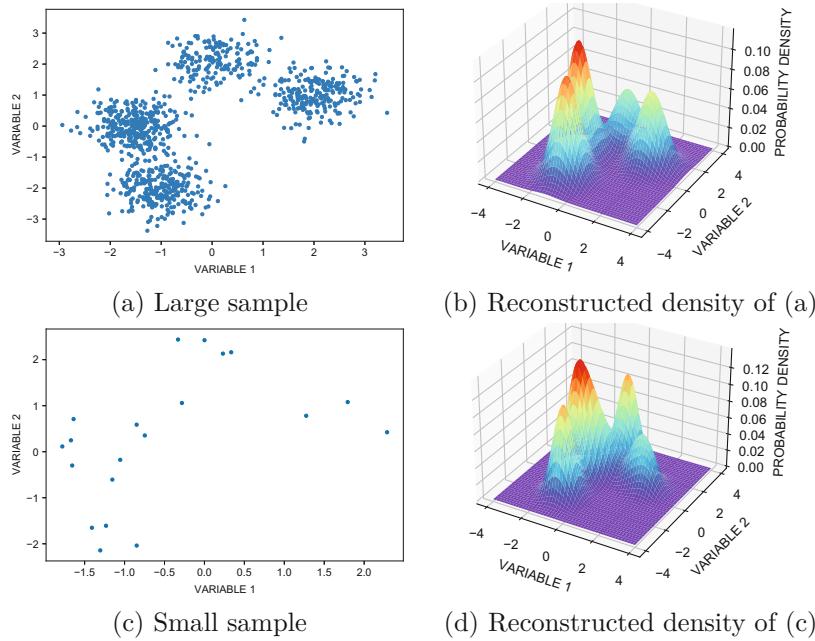


Figure 6.6: The reconstructed probability densities of a mixture of four Gaussian clusters with different samples are different. Larger samples lead to better estimations because of reduced variance in the reconstructed density values.

a specific sample of 3 points from one of the clusters may pick those points from a border region, which is not reflective of the true location of the cluster. Using a different sample of 20 points could very easily result in a very different density profile. Clearly, both profiles could not be reflective of the true (population) distribution. This error is *therefore* a natural *variance* in the probability density estimate, and is one of the important sources of the overall error in estimation. It is noteworthy that all the parametric and non-parametric methods for density estimation use some kind of averaging of functions over individual sample points in order to estimate sample parameters or densities. This averaging has the effect of reducing variance (cf. Theorem 3.3 of Chapter 3). Therefore, large samples will naturally reflect the population distribution in a stable and accurate way.

The error in the reconstruction of probability density has a detrimental effect on downstream (predictive) applications in which probability density is used. For example, if the input probability density to a probabilistic classifier is inaccurate, it will result in inaccurate class predictions of the underlying points. This pipeline can be summarized as follows:

$$\text{Sample variability} \Rightarrow \text{Variance in Distribution Reconstruction} \Rightarrow \text{Variance in Prediction}$$

Note that variance in prediction means that if we reconstruct the distributions of male and female heights with a small amount of data (and then use it to predict whether person X is male or female), then changing the sample used for reconstruction might change the prediction of X's biological sex as well. This is a problem, because one of the two predictions must be incorrect. Therefore, a high level of variance has a significantly detrimental effect on the accuracy of applications that use some form of distribution reconstruction.

Table 6.2: The variance of parameter estimations of different distributions

Distribution	MLE from section 6.3	Variance
Bernoulli	$\hat{p} = \sum_{i=1}^n x_i / n$	$p(1-p)/n$
Geometric	$\hat{p} = n / \sum_{i=1}^n x_i$	$\approx p^2(1-p)/n$
Binomial	$\hat{p} = \sum_{i=1}^n x_i / \sum_{i=1}^n m_i$	$p(1-p) / \sum_{i=1}^n m_i$
Multinomial	$\hat{p}_r = \sum_{i=1}^n x_{ir} / \sum_{i=1}^n m_i$	$p_r(1-p_r) / \sum_{i=1}^n m_i$
Exponential	$\hat{\lambda} = n / \sum_{i=1}^n t_i$	$\approx \lambda^2/n$
Poisson	$\hat{\lambda} = \sum_{i=1}^n x_i / n$	λ/n
Normal (univariate)	$\hat{\mu} = \sum_{i=1}^n x_i / n$ $\hat{\sigma}^2 = \sum_{i=1}^n (x_i - \hat{\mu})^2 / n$	σ_X^2 / n $2(n-1)\sigma_X^4 / n^2$
Gaussian (multivariate)	$\hat{\mu} = \frac{\sum_{i=1}^n \vec{x}_i}{n}$ $\hat{C} = \frac{\sum_{i=1}^n [\vec{x}_i - \hat{\mu}]^T [\vec{x}_i - \hat{\mu}]}{n}$	$\left[\begin{array}{c} (1/n)[\sigma_{X_1}^2 \dots \sigma_{X_d}^2] \\ \hline \frac{(n-1)(\sigma_{X_i, X_j}^2 + \sigma_{X_i}^2 \sigma_{X_j}^2)}{n^2} \end{array} \right]_{d \times d}$

The variance in distribution reconstruction is estimated indirectly for parametric methods by measuring the variance in parameter estimation. Variance in the parameter estimation automatically maps to variance in density estimation because the probability density function is a function of both the parameters and the sample point at which the density is estimated. Since the sample point could be chosen from one of a potentially infinite number of possibilities for continuous distributions, the variance in density estimation is often measured using the variance of parameter estimation as a surrogate.

6.6.1 Variance in Maximum Likelihood Estimation

In this section, we will give an overview of the notion of variance of parameter estimation. One observation is that the parameter estimations for many distributions are in an additive form with respect to the data points used for estimating these parameters. For example, the success parameter p of the Bernoulli distribution (cf. Table 6.2) can be estimated from the data points x_1, x_2, \dots, x_n as follows:

$$\hat{p} = \frac{\sum_{i=1}^n x_i}{n} \quad (6.19)$$

Let X denote the Bernoulli random variable, of which $x_1 \dots x_n$ are instantiations. In this case, it is clear that one can use Theorem 3.3 to compute the variance of the random variable as follows:

$$\sigma_{\hat{p}}^2 = \frac{\sigma_X^2}{n}$$

Since the value of σ_X for the Bernoulli distribution is $p(1-p)$, it follows that the variance of the estimate is $p(1-p)/n$.

Table 6.2 shows the variance of the maximum likelihood estimation of the parameters of many distributions. It is striking how many of them use the averaging form such as $\sum_i x_i / n$, which is similar to the Bernoulli case (cf. Equation 6.19). Even the binomial and multinomial estimators are roughly in this form (with minor modifications to account for varying values of m_i in different experiments). Therefore, in all such cases, the variance of the estimator can be computed as σ_X^2 / n (cf. Equation 6.6.1), where σ_X will depend on

the specific distribution being considered. This observation has been used in order to fill in the variance estimates for the Bernoulli, binomial, multinomial, Poisson, and Gaussian distributions. It is noteworthy that the variance of the parameter estimate reduces with the number of samples in each case (which is intuitively reasonable to expect). Increasing the amount of available data will naturally reduce the variability of the parameter estimate across different data samples. The variance of the dispersion parameter (i.e., variance of variance!) of the normal distribution requires the computation of the fourth moment. For the case of the normal distribution, approximate values of these variance estimates are included in Table 6.2 without proof.

The estimations of the variance of the estimate of the parameters of the geometric and exponential random variables presents challenges because the denominator of the estimator contains an aggregation of samples in each case. Specifically, the random variable corresponding to the estimator is of the form $1/\bar{X}$, where \bar{X} is a random variable containing the mean arrival times of n success/arrival events. A key point is that the variance of the reciprocal $1/\bar{X}$ of a random variable \bar{X} is hard to estimate exactly. A direct computation of the variance $E[(1/\bar{X} - \mu_{1/\bar{X}})^2]$ with integral calculus does not lead to a closed-form expression. In fact, even the expected value $\mu_{1/\bar{X}}$ of the inverse of the mean arrival time is hard to estimate. However, some approximations exist that work well when a sufficient number of samples n are used for creating the averaged random variable \bar{X} .

Consider a random variable \bar{X} representing sample averages with mean $\mu_{\bar{X}}$ and standard deviation $\sigma_{\bar{X}}$. One can use the Taylor expansion to express $1/\bar{X}$ in terms of \bar{X} , if the variable \bar{X} is “tightly” distributed around the mean $\mu_{\bar{X}}$ (which tends to happen to averaged random variables for larger values of n). This is because the ratio of the standard-deviation of the averaged sample to its mean reduces with an increasing number n of samples — the mean does not change with increasing n , whereas $\sigma_{\bar{X}} \propto 1/\sqrt{n}$. This implies that $\bar{X} - \mu_{\bar{X}}$ will be small in magnitude in relation to $\mu_{\bar{X}}$. Therefore, we have the following:

$$\frac{1}{\bar{X}} = \left(\frac{1}{\mu_{\bar{X}}} \right) \left(\frac{1}{1 + \frac{\bar{X} - \mu_{\bar{X}}}{\mu_{\bar{X}}}} \right) \approx \frac{1}{\mu_{\bar{X}}} \left(1 - \frac{\bar{X} - \mu_{\bar{X}}}{\mu_{\bar{X}}} \right)$$

The above approximation is valid only when the absolute magnitude of $(\bar{X} - \mu_{\bar{X}})$ is small in relation to that of $\mu_{\bar{X}}$. This is because we are essentially using the first-order Taylor expansion in order to approximate $1/(1+z)$ by $(1-z)$, which is valid only when $|z| \ll 1$. This occurs when \bar{X} is tightly distributed around $\mu_{\bar{X}}$. The main point of the approximation is to bring the random variable \bar{X} to the numerator as an affine function, which enables easy estimation of the variance. Using this approximation of $1/\bar{X}$, one can show the following estimate of the variance of $1/\bar{X}$ using the scaling and translation law of variances (cf. Theorem 3.11):

$$\sigma_{1/\bar{X}}^2 \approx \frac{\sigma_{\bar{X}}^2}{\mu_{\bar{X}}^4}$$

Note that the geometric distribution uses the average estimates the parameter p as $1/x_{avg}$, where x_{avg} can be viewed as a random variable that is the average of n instantiations from the geometric distribution. A single instantiation of a geometric variable has an expected value of $1/p$ and variance of $(1-p)/p^2$. The average of n independent instantiations has the same expected value of $1/p$ but a variance that is $(1-p)/(np^2)$ (cf. Theorem 3.3). On using the aforementioned formula to estimate the variance, one obtains an estimate of $p^2(1-p)/n$. One can use a similar approach to show that the variance of estimation of the

exponential random variable is roughly λ^2/n . As discussed in Chapter 4, the exponential random variable is the continuous version of the geometric distribution, and the principles involved in estimating the variance are similar. We leave the proof of this result as an exercise:

Problem 6.12 (Variance of MLE of Exponential Distribution) *Show that the variance of the maximum likelihood estimate of the parameter λ of the exponential random variable is roughly λ^2/n for large values of n where the average of n exponential random variables is tightly distributed around the mean.*

The variance increases when the data are limited, which can cause problems in terms of the quality of estimation. For example, in the case of the Bernoulli distribution, if one only has a few observations, it is possible to estimate the probability parameter to 0 or 1. For example, it is possible to not obtain any heads (i.e., success) one-eighth of the time, if a fair coin is tossed only three times. In such a case, the success parameter may be estimated to be 0. A different set of three tosses may result in all three heads, which will cause the success parameter p to be estimated as 1. Such instability is often indicative of parameter estimation error, and it is a direct manifestation of variance; such instability may cause disastrously inaccurate predictions in downstream machine learning applications that work with these reconstructed distributions. As a related data-centric example to Bernoulli estimation, if we have a sparse, binary data set (e.g., user clicks), a lack of clicks on a specific item from a small data set will cause that attribute to become irrelevant for the model. This type of scenario is often indicative of the problems caused by increased variance in the data (and can present significant problems for various applications). One approach for dealing with this problem is to incorporate a *prior belief* about the value of the parameter, which makes the estimation more stable. Such an approach is the topic of discussion of the next section.

Problem 6.13 *Compute the variance of the parameter estimation of a Bernoulli distribution with 10 trials and $p = 0.5$. Repeat the same exercise for 10 trials and $p = 0.05$. You will find that the lower variance is achieved at $p = 0.05$. In spite of this fact, discuss why the case of $p = 0.05$ is more problematic for downstream applications when the estimation variance is compared to the magnitude of p .*

6.6.2 Prior Beliefs with Maximum A Posteriori (MAP) Estimation

The high variance of the MLE estimates can sometimes pose a challenge when the amount of data is limited. One of the methods that is used in order to reduce variance is to assume that the parameters of the data distribution are themselves random variables with a *prior probability distribution*. The goal is to compute parameters that maximize the *posterior densities* (i.e., conditional densities) of the parameters given the data (rather than maximizing the likelihood of the data given the parameters). Subsequently, the conditional density based on the optimum parameter value can be used to reconstruct the distribution. Before proceeding further, we recommend the reader to revisit section 3.10 of Chapter 3 on compound distributions.

Recall that the likelihood function maximizes the following:

$$\mathcal{L}(\vec{x}_1 \dots \vec{x}_n, \vec{\theta}) = \prod_{i=1}^n p_{\vec{X}|\vec{\Theta}=\vec{\theta}}(\vec{x}_i)$$

The product on the right-hand side is the same as maximizing the probability of the data, given the parameters. However, if the parameters are viewed as samples drawn from a

probability distribution, we could also maximize the probability density of the parameter vector $\vec{\theta}$ given the data $\vec{x}_1, \dots, \vec{x}_n$. Stated formally in mathematical notation, we want to maximize $f_{\vec{\Theta}|\vec{X}_1=\vec{x}_1 \dots \vec{X}_n=\vec{x}_n}(\vec{\theta})$ instead, which puts the observed data in the conditional rather than putting the parameters in the conditional.

Why would one want to maximize the posterior density of the parameters rather than the likelihood of the data? One reason is that doing so allows us to incorporate our *prior assumptions* about the parameters when the data is very limited. The Bayes rule can be used to combine the prior assumptions with the data likelihood to create the posterior density of the data. If we make no prior assumptions, the data likelihood is a proxy for the posterior density of the parameters with the (implicit) assumption that the prior distribution of the parameters is uniform. In other words, maximum likelihood estimation makes the implicit assumption of giving equal importance to all parts of the parameter space; such an approach may cause problems when the data is limited and the parameter space is large. Therefore, a more natural approach is to impose a prior distribution on the parameters, which causes the parameter optimization to be *biased* towards specific regions of the space. This bias has a positive effect on the quality of parameter estimation (and the accuracy of downstream applications), particularly when the bias is imposed through the prior in a judicious way. The theory of regularization in machine learning can be attributed in large part to this basic principle.

Note that one can use the Bayes rule in order to compute the posterior density of the parameters given the data:

$$f_{\vec{\Theta}|\vec{X}_1=\vec{x}_1 \dots \vec{X}_n=\vec{x}_n}(\vec{\theta}) = \frac{f_{\vec{\Theta}}(\vec{\theta}) \prod_{i=1}^n p_{\vec{X}|\vec{\Theta}=\vec{\theta}}(\vec{x}_i)}{p_{\vec{X}}(\vec{x}_1, \dots, \vec{x}_n)} \propto \underbrace{f_{\vec{\Theta}}(\vec{\theta})}_{\text{Prior}} \underbrace{\prod_{i=1}^n p_{\vec{X}|\vec{\Theta}=\vec{\theta}}(\vec{x}_i)}_{\text{Likelihood}}$$

The denominator in the Bayes rule has been dropped since it is independent of the parameters that need to be optimized. Therefore, one is left only with the product of the prior and the same likelihood function that was used in maximum likelihood estimation. Note that the above formula is an application of the continuous version of the Bayes rule (cf. section 3.10 of Chapter 3).

Taking the negative logarithm of the posterior function, and calling it the negative *log-posterior* $\mathcal{LP}(\cdot)$, one obtains the following:

$$\mathcal{LP}(\vec{x}_1 \dots \vec{x}_n, \vec{\theta}) = -\ln(f_{\vec{\Theta}}(\vec{\theta})) + \mathcal{LL}(\vec{x}_1 \dots \vec{x}_n, \vec{\theta}) + C_0 \quad (6.20)$$

Here, C_0 is a constant that accounts for the proportionality factor in the posterior likelihood condition. The notation $\mathcal{LL}(\cdot)$ is inherited from earlier sections in referring to the negative log-likelihood. Therefore, *log-posteriors are different from log-likelihoods only in the addition of a log-prior function*. The derivative with respect to the parameter vector $\vec{\theta}$ is as follows:

$$\frac{1}{f_{\vec{\Theta}}(\vec{\theta})} \frac{\partial f_{\vec{\Theta}}(\vec{\theta})}{\partial \vec{\theta}} + \sum_{i=1}^n \frac{1}{p_{\vec{X}|\vec{\Theta}=\vec{\theta}}(\vec{x}_i)} \frac{\partial p_{\vec{X}|\vec{\Theta}=\vec{\theta}}(\vec{x}_i)}{\partial \vec{\theta}} = \vec{0} \quad (6.21)$$

This condition is very similar to the optimality condition of maximum likelihood estimation (cf. Equation 6.1), except that the maximum a posteriori approach has an additional term for the prior.

The main problem is that this additional term makes the resulting equation much harder to solve in closed form than in the case of maximum likelihood estimation. It is only when

we choose specific types of prior distributions that the above equation can be solved in closed form and the posterior distribution of $\bar{\Theta}$ is from the same family of distributions as the prior. This type of *algebraically convenient* prior is referred to as a *conjugate prior* distribution. The choice of conjugate prior distribution depends on the distribution of the likelihood. For example, when the likelihoods of data points follow the Bernoulli distribution, the appropriate conjugate prior of the probability p is the *beta distribution*. It is noteworthy that using the (algebraically convenient) conjugate prior is not necessary — we could use any type of prior that is appropriate for the problem domain and then solve the above equation using numerical methods. However, since numerical methods are cumbersome, there is generally a strong preference towards using conjugate priors. Numerically expressed distributions are also not interpretable. Conjugate prior distributions have the advantage of providing a clear view of how the likelihood function updates the prior distribution of the parameters to a posterior distribution.

6.6.2.1 Example: Laplacian Smoothing

As an example, we consider the MAP estimation of the Bernoulli distribution, where using the beta-distribution as the prior leads to a well-known technique called *Laplacian smoothing*. The beta distribution is defined as the prior on the Bernoulli success parameter p as follows:

$$f_{\Theta}(p) = B p^{\alpha-1} (1-p)^{\beta-1}$$

Here, $\alpha > 0$ and $\beta > 0$ are beta distribution parameters, and it is important to use $\alpha, \beta > 1$ to encourage p to take on “middling” values rather than extreme values like 0 or 1. The constant B is chosen so that the density function integrates to 1:

$$B = \frac{1}{\int_0^1 x^{\alpha-1} (1-x)^{\beta-1} dx}$$

One can re-parameterize this distribution using $\lambda > 0$ and $q > 0$ satisfying $\alpha = \lambda + 1$ and $\beta = \lambda q + 1$. As we will see later, these parameters are intuitively more interpretable:

$$f_{\Theta}(p) = B [p(1-p)^q]^{\lambda}$$

The mode of this distribution can be shown to be $1/(1+q)$, which biases the MAP estimation of p towards this value. The value of the exponent $\lambda > 0$ in the probability density function controls the strength of the bias — an exponent value λ that is close to zero results in a prior density function that does not change much over $p \in (0, 1)$; such a choice is equivalent to using a uniform prior with no bias. The negative log-posterior is obtained by adding a constant term and $-\lambda \ln[p(1-p)^q]$ to the negative log-likelihood:

$$\mathcal{LP}(x_1 \dots x_n, p) = -\lambda \ln(p) - \lambda q \ln(1-p) - \left(\sum_i x_i \right) \ln(p) - \left(n - \sum_i x_i \right) \ln(1-p) + C_1$$

Here, C_1 absorbs the constant B in the prior as well as the proportionality factor relating the posterior to the product of the prior and the likelihood. The notation x_i denotes the i th binary observation out of n observations. Differentiating the above equation with respect to p , setting it to zero, and simplifying with cross-multiplication yields the following:

$$(\lambda + q\lambda + n)p - \lambda - \sum_{i=1}^n x_i = 0$$

The MAP estimate can be shown to be the following:

$$\hat{p} = \frac{\lambda + \sum_{i=1}^n x_i}{\lambda + q\lambda + n} \quad (6.22)$$

This optimal solution is in exactly the same form given by maximum likelihood estimation of the Bernoulli distribution, except that λ *fake successes and λq fake failures have been added to the data*. These fake trials reflect our prior opinion of the ratio q of failures to successes. Increasing λ increases the influence of fake trials, corresponding to greater bias. In the case of the Bernoulli distribution, it makes sense to set $q = 1$ (corresponding to equal successes and failures) when there is no prior knowledge³ about the relative presence of successes and failures. This choice yields the following posterior estimate:

$$\hat{p} = \frac{\sum_{i=1}^n x_i + \lambda}{n + 2\lambda}$$

In the absence of any data, the estimated parameter \hat{p} will be 0.5, which is the bias caused only by fake trials. In the case of $\lambda = 0$, the MAP estimate is $\hat{p} = \sum_{i=1}^n x_i/n$, which is exactly the same as the maximum likelihood estimate. This situation corresponds to a uniform prior in which no prior opinion has been expressed about the ratio of successes and failures. Laplacian smoothing is particularly helpful in the presence of limited data because it avoids estimations where Bernoulli probabilities are set to values such as 0 or 1 — such choices can have disastrous effects in downstream applications.

It is common to incorporate a similar type of prior for the categorical and multinomial distributions. For a categorical or multinomial distribution with k outcomes, the value of q is chosen to be $(k - 1)$, which reflects the prior belief that each category has a default probability of $1/k$. Let n_r be the number of outcomes for the r th category out of n outcomes. In such a case the estimate of the probability \hat{p}_r of the r th category is as follows:

$$\hat{p}_r = \frac{n_r + \lambda}{n + k\lambda} \quad (6.23)$$

The nature of the bias here is similar to that of the Bernoulli distribution. The word “bias” alludes to the fact that these assumptions may not always be correct but they substitute well for cases where enough data is not available to estimate parameters meaningfully.

Example 6.13 You have a data set containing 10 individuals with their demographic information. There are a total of 6 possible races and the prior assumption is that all races are equally likely. You find that your data does not contain any Native American person. Find a MAP estimation for the fraction of Native Americans in the population in terms of the Laplacian smoothing parameter λ . What are the estimations for $\lambda = 0.01$ and $\lambda = 100$?

Solution: Assume that the first category is Native American. Using Equation 6.23 with $n_1 = 9$, $n = 10$, and $k = 6$, the following MAP estimation is obtained:

$$\hat{p}_1 = \frac{n_1 + \lambda}{n + k\lambda} = \frac{\lambda}{10 + 6\lambda}$$

³If one has prior knowledge that the ratio of successes to failures is often 1 : 2 in similar data sets, then one can set $q = 2$. This choice defines the nature of the bias on the basis of prior information.

Using $\lambda = 0.01$ results in a probability estimation of 0.001, which is close to the maximum likelihood value of 0. In other words, the effect of smoothing is limited. Using $\lambda = 100$ results in an estimation of 10/61, which is close to the prior value of 1/6. The results show that increasing λ causes increasing influence of the prior on the final estimation. ■

Example 6.14 We revisit Example 6.1 with the use of maximum a posteriori estimation. Consider a family of density functions of the form $f_{X|\Theta=\theta}(x) = \theta x^{\theta-1}$, where $\theta > 0$ is the distribution parameter and $x \in (0, 1)$. Domain knowledge about the parameter θ tells you that it has an exponential prior distribution $f_\Theta(\theta) = (1/2)\exp(-\theta/2)$. You have the observed data points $x_1 \dots x_n$ from $(0, 1)$ that were generated from this distribution. Find the MAP estimate of θ in terms of $x_1 \dots x_n$. Consider the case where $n = 1$ and you have a single observed point x_1 . Find the optimal numerical value of θ for (a) $x_1 = 0.1$, (b) $x_1 = 0.5$, and (c) $x_1 = 0.9$. Comment on the results in comparison with Example 6.1.

Solution: Since the posterior function will contain the prior function as an additional factor, the negative log posterior will contain $-\ln(0.5\exp(-\theta/2)) = \theta/2 + \ln(2)$ as an additive term. Therefore, the negative log-posterior term in the solution to Example 6.1 is modified by adding $\theta/2 + \ln(2)$:

$$\begin{aligned}\mathcal{LP}(x_1 \dots x_n, \theta) &= \theta/2 + \ln(2) - \sum_{i=1}^n \ln(f_X(x_i)) \\ &= \theta/2 + \ln(2) - n\ln(\theta) - (\theta - 1)(\sum_{i=1}^n \ln(x_i))\end{aligned}$$

On differentiating the negative log-posterior and setting it to 0, one obtains the following:

$$\frac{n}{\theta} = 0.5 - \sum_{i=1}^n \ln(x_i)$$

In other words, the optimal parameter θ^* is as follows:

$$\theta^* = \frac{n}{0.5 - \sum_{i=1}^n \ln(x_i)}$$

On substituting $x_1 = 0.1$ based on case (a), we obtain $\theta^* = 1/(0.5 - \ln(0.1)) \approx 0.357$. On substituting $x_1 = 0.5$ based on case (b), we obtain $\theta^* = 1/(0.5 - \ln(0.5)) \approx 0.838$. On substituting $\theta = 0.9$ for case (c), we obtain $\theta^* = -1/(0.5 - \ln(0.9)) \approx 1.652$.

The estimates of the parameters reduce in each case compared to the likelihood estimates in Example 6.1. The effect is particularly noticeable for case (c) in which the likelihood-based parameter estimate is 9.492, whereas the MAP-based parameter estimate has a much smaller value of 1.652. Using an exponential prior on a positive parameter like θ is very similar to using a technique called *L₁-regularization* (cf. Chapter 7), which uses the double-exponential (Laplace) prior. The prior biases the parameters to be distributed near the origin, which reduces their magnitude. ■

Example 6.15 We revisit Example 6.2 with maximum a posteriori estimation. Two biased coins with heads probabilities p_1 and p_2 , respectively, are flipped simultaneously. Then, a payoff is made, which is equal to the square of the number of heads. The experiment is repeated n times and pay-offs of 0, 1, and 4 are observed f_0 , f_1 and f_2 times. Set up the equation(s) to find the maximum a posteriori estimates of p_1 and p_2 where the beta distribution $f_\theta(p) \propto [p(1-p)]^\lambda$ with $q = 1$ is used as the prior. Now find the MAP estimate in terms of λ for a single trial and (a) reward of 0, (b) reward of 1, and (c) reward of 4.

Solution: The analysis is exactly identical to that in the solution to Example 6.2, except that the negative log-posterior is of the form $-\lambda[\ln(p_1(1-p_1)) + \ln(p_2(1-p_2))] + \mathcal{LL}$ where \mathcal{LL} is the negative-log likelihood in the solution to Example 6.2. The partial derivatives of these additional terms will modify the optimality conditions in the solution to Example 6.2 as follows:

$$\begin{aligned}\frac{f_0 + \lambda}{1 - p_1} - \frac{f_1(1 - 2p_2)}{p_1 + p_2 - 2p_1p_2} - \frac{f_2 + \lambda}{p_1} &= 0 \\ \frac{f_0 + \lambda}{1 - p_2} - \frac{f_1(1 - 2p_1)}{p_1 + p_2 - 2p_1p_2} - \frac{f_2 + \lambda}{p_2} &= 0\end{aligned}$$

Now considering $[f_0, f_1, f_2] = [1, 0, 0]$ for case (a), we obtain the optimality conditions $(1 + \lambda)/(1 - p_1) = \lambda/p_1$ and $(1 + \lambda)/(1 - p_2) = \lambda/p_2$. These optimality conditions yield $p_1 = p_2 = \lambda/(1 + 2\lambda)$. Note that these probabilities are always less than 0.5 and close to 0 for small λ . This result matches intuition because no reward was received. Considering $[f_0, f_1, f_2] = [0, 1, 0]$ for case (b), we obtain the optimality conditions $1 - 2p_1 = 1 - 2p_2 = 0$, corresponding to $p_1 = p_2 = 0.5$. In other words, both coins are fair, and the observed data tells us that one of them turned up heads.

Considering $[f_0, f_1, f_2] = [0, 0, 1]$ for case (c), it can be shown in a similar way to case (a) that $p_1 = p_2 = (1 + \lambda)/(1 + 2\lambda)$. These probabilities are always greater than 0.5 and close to 1 for small λ . This result matches intuition because a maximum reward of 4 was received. ■

Problem 6.14 Repeat Example 6.14 with exponential priors with (a) parameter value of 0.1, and (b) parameter value of 1. Comment on the effect of prior choice.

Problem 6.15 Suppose you toss a (possibly biased) coin thrice and it turns up heads each time. Find the MLE estimate of the probability of a heads outcome. Compute the Laplacian-smoothed MAP estimate assuming that both outcomes are equally likely from the prior-belief perspective and $\lambda = 1$. What happens to the MAP estimate when you use $\lambda = 100$?

6.6.3 Kernel Density Estimation: Role of Bandwidth

The kernel density estimation technique is a non-parametric method, and therefore it makes sense to directly estimate the variance of the density at a particular point (rather than the values of the parameters of a modeled distribution). While the full analytical results on variance are quite complex for kernel density estimation, we will provide some intuition on the role of the bandwidth in density estimation. Our goal is to show that small values of the bandwidth lead to higher variance and vice versa. We state the 1-dimensional simplification

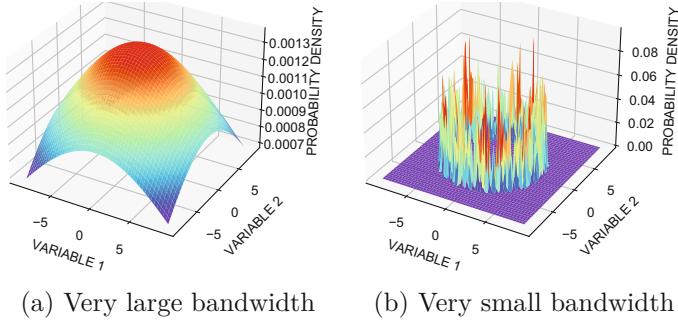


Figure 6.7: The density estimation of the ring of points in Figure 6.5 with very large and very small values of the bandwidth. The estimation is misleading in each case.

of the kernel density estimate, where the set of 1-dimensional observations in a data set are given by $x_1 \dots x_n$. The density at z is estimated as follows:

$$\hat{f}_X(z) = \frac{1}{nh\sqrt{2\pi}} \sum_{i=1}^n \exp\left(-\frac{(z - x_i)^2}{2 \cdot h^2}\right) \quad (6.24)$$

The bandwidth along the single dimension is denoted by h . Here, a key point is that the value of n occurs in the denominator, which reduces the variance of the estimate, whereas the numerator is independent of n . Furthermore, the bandwidth h occurs in the denominator of the estimate as well as the negative exponent. The overall effect of large bandwidth is to reduce the ratio of the variance to the squared magnitude of the density. As a result, large values of h tend to reduce the (relative) variance. On the other hand, using very large values of h increases bias in an excessive manner that washes out any learned knowledge in the data.

As a specific example, we revisit the construction of the kernel density estimate of the ring of points illustrated in Figure 6.5. By increasing the bandwidth by a factor of 10 over that in Figure 6.5, the corresponding density estimate is illustrated in Figure 6.7(a). In this case, the very large spread of the kernel tends to increase the density of central regions of the data even though there are few points there. The natural ring-shaped estimate has clearly been lost. This type of estimate is a result of our bias that a sampled point includes a large amount of random noise whose effect must be accounted for by using a large bandwidth (wide bump) to increase density at all places in the distribution space. Unfortunately, all bumps for different points contribute significantly to the center of the ring (when the bandwidth is large), and therefore central regions end up having the largest density. This is obviously not reflective of the true distribution in which the center of the ring is empty! Using a different data sample from the same distribution may not change the density estimates in an appreciable way, which is reflective of low variance. Although this feature may be considered a desirable one, the bias dominates the shape of the distribution — therefore, a very large bandwidth is not very helpful in learning anything new from the data beyond our prior assumptions.

What happens when a very small value of the bandwidth is used? By reducing the bandwidth of the density estimate in Figure 6.5 by a factor of 10, the corresponding density estimate is illustrated in Figure 6.7(b). In this case, it is evident that the density estimate has not been sufficiently smoothed and therefore the densities correspond to a bunch of spikes. Changing the data set to a different sample of the same distribution will completely change

the density estimate. This is obviously not very helpful either. Silverman's approximation rule for the bandwidth tends to avoid both the extreme situations illustrated in Figure 6.7.

The aforementioned discussion shows that there is a clear trade-off between the bias and the variance in many statistical settings. Bias improves error by reducing variability in results (thereby reducing inconsistency across different samples), but increasing the bias too much may result in consistently wrong answers. On the other hand, increasing variance too much may result in widely varying results, some of which are obviously wrong. This leads to a natural bias-variance trade-off, which is the topic of discussion in the next section.

Example 6.16 Consider a 1-dimensional data set containing 2 points at 0 and 4, respectively. Calculate the density at 0, 1, and 2 using bandwidth $h = 1$. Now repeat the process using $h = 4$. What do you observe?

Solution: Using $h = 1$, the kernel density values are as follows:

$$\begin{aligned} K_h(0) &= \frac{1}{\sqrt{2\pi}} (\exp(-16/2) + \exp(0)) \approx \frac{1}{\sqrt{2\pi}} \\ K_h(1) &= \frac{1}{\sqrt{2\pi}} (\exp(-9/2) + \exp(-1/2)) \approx \frac{0.618}{\sqrt{2\pi}} \\ K_h(2) &= \frac{1}{\sqrt{2\pi}} (\exp(-2) + \exp(-2)) \approx \frac{0.271}{\sqrt{2\pi}} \end{aligned}$$

An observation is that the density values decrease as one approaches the middle of the two points, which are spaced at a distance of 4. This is because the bandwidth defines 1 as a significant distance. As a result, the middle point is considered a sparse region.

Using $h = 4$, the kernel density values are as follows:

$$\begin{aligned} K_h(0) &= \frac{1}{4\sqrt{2\pi}} (\exp(-16/32) + \exp(0)) \approx \frac{0.402}{\sqrt{2\pi}} \\ K_h(1) &= \frac{1}{4\sqrt{2\pi}} (\exp(-9/32) + \exp(-1/32)) \approx \frac{0.431}{\sqrt{2\pi}} \\ K_h(2) &= \frac{1}{4\sqrt{2\pi}} (\exp(-1/8) + \exp(-1/8)) \approx \frac{0.441}{\sqrt{2\pi}} \end{aligned}$$

In this case, the density values increase as one moves to the middle of the two points. The density values also vary in a smooth and slow way. This is because a larger bandwidth is used. Most points are expected to be further away than the bandwidth. Therefore the middle of the two points is seen as a more dense region. ■

6.7 The Bias-Variance Trade-Off

The previous section shows how one can reduce the variance of reconstructed distributions by adding prior beliefs, which is a form of bias. This type of bias help the accuracy of downstream applications when the amount of available data is small, because it provides the statistician to substitute the (lack of) data-driven evidence with reasonable beliefs. However, the approach can also be detrimental in terms of constraining the model, when

the data has sufficient complexity to infer useful evidence. Clearly, there is a sweet spot on the amount of bias that should be incorporated, and we refer to this as the bias-variance trade-off. In this section, this trade-off will be explored in detail.

Although the bias-variance trade-off applies to every stage of the data analysis pipeline (like parameter estimation), it is generally used in the context of downstream applications like prediction. Although prediction has not been addressed so far in this book, we will provide a gentle introduction to prediction in the context of reconstructed distributions. The reconstructed distributions are often used by Bayes classifiers to predict the class labels of observations. For example, by reconstructing the distributions of the heights of males and females, one can probabilistically predict whether the height of a particular observation belongs to a male or female. An example of such a pair of distributions is provided in Figure 6.1. Other forms of prediction such as *regression* predict numeric values rather than binary labels. These predicted values are referred to as *targets*, and the predictions (e.g., probability of male or female) are typically performed on features (e.g., heights) of instances (e.g., people associated with the heights) that are not part of the *training data* used to reconstruct the distributions of males and females. Typically the distributions of males and females are separately reconstructed from gender-tagged training data. Distinct from the training data, these instances are referred to as *test data* (e.g., untagged heights whose gender probability needs to be predicted). Throughout this section, we will use the male-female classification example in order to explain the bias-variance trade-off.

Consider the situation where we have t test instances whose true gender values are $y_1 \dots y_t \in \{0, 1\}$. Assume that $t \rightarrow \infty$, so that the sample-specific analysis here generalizes directly to the corresponding distribution. While evaluating the accuracy of prediction, one can use the observed values $y_1^o \dots y_t^o \in \{0, 1\}$ of the gender, which may be (very occasionally) different from the true values because of errors in data collection. Therefore, we have:

$$y_i^o = y_i + \epsilon_i$$

Note that $\epsilon_i \in \{-1, 0, 1\}$ and it is assumed to be 0 most of the time. Furthermore, ϵ_i is statistically independent of the observed data y_i^o . The term ϵ_i is referred to as the *noise*. Now imagine that the analyst uses some form of distribution reconstruction to predict probability values $\hat{y}_1 \dots \hat{y}_t \in (0, 1)$ for the gender. Clearly, we want these probability values to be as close to the true gender y_i as possible. Therefore, the squared error E of prediction can be computed as follows:

$$E = \frac{\sum_{i=1}^t (y_i - \hat{y}_i)^2}{t}$$

One can substitute $y_i = y_i^o - \epsilon_i$ in order to show the following for the squared error:

$$E = \frac{\sum_{i=1}^t (y_i^o - \hat{y}_i - \epsilon_i)^2}{t} = \frac{\sum_{i=1}^t [(y_i^o - \hat{y}_i)^2 + \epsilon_i^2 - 2\epsilon_i(y_i^o - \hat{y}_i)]}{t}$$

Note that since the predictions \hat{y}_i are made using a model on the observed data, and the observed data is independent of ϵ_i , it follows that $(y_i^o - \hat{y}_i)$ is also statistically independent of ϵ_i . Therefore, the term $\sum_i \epsilon_i(y_i^o - \hat{y}_i)/t$ can be set to the product of the average value of ϵ_i and the average value of $(y_i^o - \hat{y}_i)$. Since the average value of ϵ_i is 0 as $t \rightarrow \infty$, it follows that the value of the term drops to 0. This leads to the following simplification for the mean-squared error (MSE):

$$\begin{aligned} MSE &= \frac{\sum_{i=1}^t [(y_i^o - \hat{y}_i)^2 + \epsilon_i^2]}{t} = \frac{\sum_{i=1}^t (y_i^o - \hat{y}_i)^2}{t} + \frac{\epsilon_i^2}{t} \\ &= \text{Observed MSE} + \text{Noise} \end{aligned}$$

There is not much one can do in terms of addressing the noise component because it is an inherent problem with data quality. We denote the observed MSE by MSE^o and further decompose it into its statistical components.

The key point is that as different samples of the *training* data are taken, the same observation y_i^o will be predicted in different ways. In other words, the value of \hat{y}_i will be different over different distribution reconstructions. Let μ_i be the mean prediction of observation i over a large number of training samples (and corresponding distribution reconstructions). Then, the observed error can be further decomposed as follows:

$$\begin{aligned} MSE^o &= \frac{\sum_{i=1}^t (y_i^o - \hat{y}_i)^2}{t} = \frac{\sum_{i=1}^t ((y_i^o - \mu_i) - (\hat{y}_i - \mu_i))^2}{t} \\ &= \frac{\sum_{i=1}^t (y_i^o - \mu_i)^2}{t} + \frac{\sum_{i=1}^t (\hat{y}_i - \mu_i)^2}{t} - 2 \frac{\sum_{i=1}^t (y_i^o - \mu_i)(\hat{y}_i - \mu_i)}{t} \end{aligned}$$

Then, the expected value of MSE^o over different random choices of training samples (for distribution reconstruction) is denoted by $E_x[MSE^o]$. We use the subscript x in the expectation to indicate that it is the *training observations* that are sampled (whereas the test samples are fixed), and therefore the value of \hat{y}_i will change with the training sample. All other quantities such as y_i^o and μ_i do not depend on the choice of training sample (and can therefore be pulled out of the expectation where they occur as separable factors). It is also noteworthy that μ_i can be interpreted at $E_x[\hat{y}_i]$. Therefore, taking the expectation of MSE^o over training samples we obtain the following:

$$\begin{aligned} E_x[MSE^o] &= \frac{\sum_{i=1}^t (y_i^o - \mu_i)^2}{t} + \frac{\sum_{i=1}^t E_x[(\hat{y}_i - \mu_i)^2]}{t} - 2 \underbrace{\frac{\sum_{i=1}^t (y_i^o - \mu_i) E_x[(\hat{y}_i - \mu_i)]}{t}}_0 \\ &= \underbrace{\frac{\sum_{i=1}^t (y_i^o - \mu_i)^2}{t}}_{\text{Squared Bias}} + \underbrace{\frac{\sum_{i=1}^t E_x[(\hat{y}_i - \mu_i)^2]}{t}}_{\text{Variance}} \end{aligned}$$

It is noteworthy that the first term computes the squared difference between the averaged prediction over many training data sets and the observed prediction. This value is the squared bias because it is indicative of the error of the *averaged prediction* over many training data sets, and is therefore indicative of a consistent direction in error (e.g., consistently classifying a male test instance as female over different choices of training samples). On the other hand, the part $\sum_{i=1}^t E_x[(\hat{y}_i - \mu_i)^2]/t$ is the error caused by the variability (i.e., inconsistency) in distribution reconstruction over different choices of training samples, which we refer to as the *variance*. This part of the error is high when the i th test instance is classified as either male or female with high probability over different choices of training samples (possibly because the reconstructed distributions are very different over different choices of training samples, which causes different predictions of the same test instance). Therefore, one can write the above expression for the expected value of the observed error as follows:

$$E_x[\text{Observed MSE}] = \text{Bias}^2 + \text{Variance}$$

The decomposition of the overall error into bias and variance is important because most design choices of the algorithm lead to a trade-off between the bias and variance. For example, reducing λ leads in increased variance and reduced bias, when the reconstructed Bernoulli distribution is used in a Bayes classifier. While the sum of the two does not vary as significantly with changing λ as either the bias or the variance, it does tend to take on

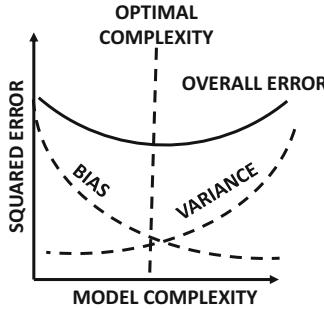


Figure 6.8: The point of optimal model complexity in the bias-variance trade-off

its lowest values at intermediate values of λ . A similar observation can be made for the kernel bandwidth h , where an intermediate value (near a value indicated by Silverman's approximation rule) provides the best results. A key point is that incorporating prior assumptions tends to simplify models while reducing the complexity caused by the nuances of different data distribution. More complex models have greater variance because some of the learned complexity is sample-specific. A point of optimal complexity occurs somewhere in the middle where the most accurate results are obtained. This trade-off between the bias and variance is shown in Figure 6.8 along with the point of optimal model complexity.

At the left end of Figure 6.8, the models tend to be oversimplified. As a result, changing the training sample does not affect the prediction for a downstream application like classification (by incorporating bias). However, this can be a problem if the bias is so overwhelming that it overshadows useful patterns in the training sample. For example, the reconstructed distribution in density estimation for high kernel bandwidth will always show the same shape (cf. Figure 6.7(a)) irrespective of the true data distribution of observed training samples. As a result, the model will not provide good predictions, when such a reconstructed distribution is used for making predictions on test samples. On the other hand, using a very small bandwidth leads to a reconstructed distribution that overfits the specific training data (cf. Figure 6.7(b)). Therefore, if it is used to make predictions on new observations, the results will be poor as well. A large part of the tuning of models in data science is geared towards finding the “sweet spot” in the middle where overall error is minimized. Techniques like MAP estimation try to reach the sweet spot in the middle by adding prior assumptions (which serve the role of bias). On the other hand, pure MLE estimation may have high variance when the amount of observed data is little.

Example 6.17 (Bias-Variance Trade-Off in Laplacian Smoothing) *The Laplacian smoothing of Bernoulli parameter estimation has two hyper-parameters λ and q , and it uses binary data x_1, x_2, \dots, x_n to estimate the Bernoulli parameter as follows (Equation 6.22):*

$$\hat{p} = \frac{\lambda + \sum_{i=1}^n x_i}{\lambda + q\lambda + n}$$

What role do λ and q play in terms of the bias-variance trade-off?

Solution: The value of q defines the nature of the bias, because it defines the analyst's prior knowledge about the ratio of successes to failure ($1 : q$). Laplacian smoothing

Table 6.3: Popular conjugate prior distributions in machine learning

Likelihood distribution parameter	Conjugate Prior	Machine Learning Application(s)
Bernoulli success parameter	beta	Laplacian smoothing
Categorical parameter vector (one-hot conversion)	beta	Laplacian smoothing
Categorical parameter vector (multivariate version)	Dirichlet	Generalized linear models (categorical attributes or targets) [28, 45]
Poisson arrival rate	gamma	Generalized linear models (count targets) [13, 28, 45]
Exponential arrival rate	gamma	Generalized linear models (waiting time) [13, 45]
Multinomial parameter vector	Dirichlet	Latent Dirichlet Allocation (topic models) [12]
Gaussian mean (known variance)	Gaussian	L_2 -regularized regression (Chapter 7)

is equivalent to adding λ fake successes and λq fake failures to the data. Therefore, increasing λ increases the relative importance of bias and moves along the trade-off curve on Figure 6.8 towards the left (simpler model). Often λ is chosen in a manner that maximizes prediction accuracy on downstream algorithms like classification. This provides the sweet spot of the bias-variance trade-off. ■

6.8 Popular Distributions Used as Conjugate Priors (*)

Section 6.6.2 on MAP estimation discusses the example of Laplacian smoothing in which the beta distribution is used as the conjugate prior to the Bernoulli success parameter of the likelihood distribution. Conjugate prior distributions have the property that a closed-form solution exists for MAP estimation. Furthermore, the posterior distribution comes from the same distribution family as the prior distribution. The specific choice of the conjugate prior distribution depends on the likelihood distribution. It is noteworthy that choosing a distribution from the conjugate prior family is only an algebraic convenience, and one can choose any prior distribution if one is willing to live with numerically optimized solutions (that are not in closed form). Choosing conjugate prior distributions also comes with an interpretability advantage in terms of the visible change in the parameters from the prior to the posterior from the same distribution family.

This section provides a working knowledge of how conjugate prior distributions are used in MAP estimation. A reader who is not specifically interested in Bayesian statistics may choose to skip over the material in this section (without loss of continuity). Therefore, this section has been designated as an asterisked (advanced and optional) section. Table 6.3 provides a list of some key likelihood functions together with their conjugate prior distributions. It can be seen that the most popular distributions used as conjugate priors are the gamma, beta, and Dirichlet distributions. These distributions are discussed in the following section (together with examples of MAP estimation).

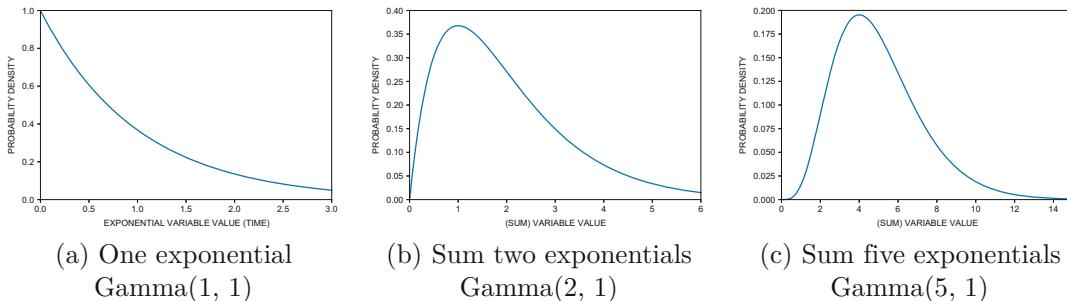


Figure 6.9: Revisiting Figure 4.8: The special case of gamma distributions with integral α is a sum of i.i.d. exponential distributions and is referred to as the Erlang distribution.

6.8.1 Gamma Distribution

The gamma distribution is a generalization of the exponential family of distributions with the following probability density function for $x > 0$:

$$f_X(x) = \frac{1}{\Gamma(\alpha)} \beta^\alpha x^{\alpha-1} \exp(-\beta x) \quad \forall x > 0$$

The parameter $\alpha > 0$ is referred to as the *shape parameter*, whereas the parameter $\beta > 0$ is referred to as the *rate parameter*. Setting α to 1 results in the exponential distribution with arrival rate $\beta > 0$. The above distribution uses a *gamma function* $\Gamma(\alpha)$:

$$\Gamma(\alpha) = \int_{x=0}^{\infty} x^{\alpha-1} \exp(-x) dx$$

When α is a natural number, it can be shown that $\Gamma(\alpha) = (\alpha-1)!$. When the random variable X has a gamma distribution with parameters α and β , it is denoted by the following:

$$X \sim \text{Gamma}(\alpha, \beta)$$

The shape parameter α need not be an integer, but it needs to be positive. For integer values of the shape parameter, the gamma distribution can be interpreted as the sum of α i.i.d. exponential distributions. This special case of the gamma distribution is referred to as the *Erlang distribution*, and is useful for modeling waiting times in several real-world settings. Examples of such gamma distributions for a rate parameter of 1 (but varying integer shape parameter values) are shown in Figure 6.9. Although the characterization of the gamma distribution in terms of i.i.d. exponentials works only for integer α , it is easy to derive expressions for quantities such as the mean, variance, and moment generating function in this special case. It turns out that these expressions happen to apply to non-integer α as well. Therefore, this section will derive these expressions using this simplification.

If T_1, \dots, T_α have i.i.d. exponential distributions with arrival rate β , the Erlang random variable X corresponding to the sum of these α i.i.d. exponential distributions is as follows:

$$X = \sum_{i=1}^{\alpha} T_i$$

The mean, variance, and moment generating function of X can be derived in a simple manner by using the sum of α i.i.d. exponential random variables:

$$\mu_X = \alpha/\beta, \quad \sigma_X^2 = \alpha/\beta^2$$

$$f_X^T(s) = \left(\frac{\beta}{\beta - s} \right)^\alpha$$

When the value of α becomes large, the gamma distribution converges to the normal distribution because of the central limit theorem. The gamma distribution is often used to model the prior distribution of the arrival rate of the Poisson and exponential distributions [13, 28, 45] in MAP estimation.

Example 6.18 (MLE Estimation of Gamma Distribution) *Given the n observed points x_1, \dots, x_n , find the MLE of the gamma distribution when the value of $\alpha = \alpha_0$ is known up front and only β needs to be estimated. This is sufficient in many applications where α is known up front.*

Solution: The likelihood function of the Gamma distribution at fixed $\alpha = \alpha_0$ is expressed as follows:

$$\mathcal{L}(x_1, \dots, x_n, \beta) = \prod_{i=1}^n f_X(x_i) \propto \prod_{i=1}^n \beta^{\alpha_0} \exp(-\beta x_i)$$

The factor $x_i^{\alpha-1}/\Gamma(\alpha)$ has been ignored in each PDF, since it does not depend on β and is irrelevant to the optimization. The corresponding negative log-likelihood function is as follows:

$$\mathcal{LL}(x_1, \dots, x_n, \beta) = \text{Constant} - n\alpha_0 \ln(\beta) + \beta \sum_{i=1}^n x_i$$

On differentiating with respect to β and setting to 0, one obtains the condition $\sum_{i=1}^n x_i - n\alpha_0/\beta = 0$. This condition yields $\hat{\beta} = \alpha_0/\bar{x}$, where \bar{x} is the mean of x_1, \dots, x_n . The second derivative is $n\alpha_0/\beta^2$, which is positive. Therefore, this solution is a minimum of the negative log-likelihood function. ■

Example 6.19 (MAP Estimation of Exponential with Gamma Prior)

Section 6.3.6 discusses the MLE estimation of the exponential rate parameter λ as $n/(\sum_{i=1}^n t_i)$ for n observed time samples t_1, \dots, t_n . Show that if MAP estimation is used with a prior $\text{Gamma}(\alpha, \beta)$ distribution, then the following estimate is obtained for the exponential rate parameter λ :

$$\hat{\lambda} = \frac{n + \alpha - 1}{\sum_{i=1}^n t_i + \beta}$$

Solution: Based on the MLE estimation discussed in section 6.3.6, the negative log-likelihood function is defined by Equation 6.11, which is replicated below:

$$\mathcal{LL}(t_1, \dots, t_n, \lambda) = -n \cdot \ln(\lambda) + \lambda \sum_{i=1}^n t_i$$

The main difference in MAP estimation is that a gamma prior is imposed on λ , which is replicated below:

$$f_\Theta(\lambda) = C\lambda^{\alpha-1}\exp(-\beta\lambda)$$

Here, C is a constant that is independent of λ . According to Equation 6.20, the negative log-likelihood can be converted to the negative log-posterior by adding the negative of the logarithm of the prior (while ignoring constant terms from the optimization perspective). Therefore, the negative log-posterior is as follows:

$$\mathcal{LP}(t_1, \dots, t_n, \lambda) = -n \cdot \ln(\lambda) + \lambda \sum_{i=1}^n t_i - (\alpha - 1)\ln(\lambda) + \beta\lambda$$

On differentiating the above expression with respect to λ and setting it to zero, one obtains the following:

$$-\frac{n}{\lambda} + \sum_{i=1}^n t_i - \frac{\alpha - 1}{\lambda} + \beta = 0$$

On simplifying, one obtains the following parameter estimate:

$$\hat{\lambda} = \frac{n + \alpha - 1}{\sum_{i=1}^n t_i + \beta}$$

■

6.8.2 Beta Distribution

The beta distribution is designed to capture univariate random variables that are distributed in $(0, 1)$. The beta distribution is a conjugate prior when the likelihood functions are the Bernoulli, binomial, and geometric distributions. Therefore, it is used for MAP estimation of the success parameters of these distributions in the form of Laplacian smoothing (cf. page 280).

The beta distribution has the following probability density function:

$$f_X(x) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1}$$

The variable x must lie in $(0, 1)$ and therefore this distribution is ideal for modeling the density function of probability parameters in Bayesian statistics. The density is therefore proportional to $x^{\alpha-1}(1-x)^{\beta-1}$ and the proportionality factor is chosen so that the density function integrates to 1. The reciprocal of the proportionality factor is referred to as the beta function $B(\alpha, \beta)$, and is defined as follows:

$$B(\alpha, \beta) = \int_{x=0}^1 x^{\alpha-1} (1-x)^{\beta-1} dx$$

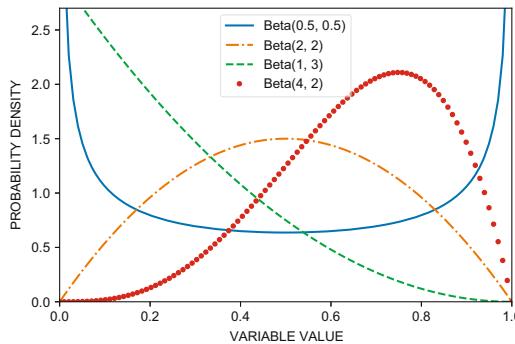


Figure 6.10: Examples of various cases of the beta distribution

The parameters α and β are referred to as shape parameters. Both parameters are greater than 0. Setting $\alpha = \beta = 1$ leads to a uniform distribution in $[0, 1]$. Examples of various cases of the beta distribution (i.e., for different values of α and β) are shown in Figure 6.10. The beta distribution is denoted as follows:

$$X \sim \text{Beta}(\alpha, \beta)$$

The beta distribution is symmetric about a mean value of 0.5 when $\alpha = \beta$. For example, Beta(0.5, 0.5) and Beta(2, 2) have very different shape, but they are both symmetric about the mean value of 0.5 (cf. Figure 6.10).

The mean and variance of the beta distribution are as follows:

$$\mu_X = \frac{\alpha}{\alpha + \beta}$$

$$\sigma_X^2 = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}$$

The moment generating function of a beta distribution takes on the form of an infinite series rather than a concise function in closed form. In particular, the moment generating function of the beta distribution is as follows:

$$f_X^T(s) = 1 + \sum_{k=1}^{\infty} \left(\prod_{r=1}^{k-1} \frac{\alpha + r}{\alpha + \beta + r} \right) \frac{s^k}{k!}$$

We make the following observations about the beta distribution:

1. Choosing α and β to be less than 1 will cause the largest densities at the end points of the domain $[0, 1]$. On the other hand, choosing α and β to be greater than 1 will cause the largest densities somewhere in the middle of the range (depending on the relative values of α and β). In such case, the mode of the distribution can be shown to be $(\alpha - 1)/(\alpha + \beta - 2)$. In the event that α and β are both less than 1, an *anti-mode* (minimum probability density) exists somewhere in the middle of the distribution. When used as a conjugate prior of the Bernoulli distribution, values of α and β greater than 1 are chosen in order to allow the existence of a mode in the middle of the range (which is close to the prior “guess” for the success parameter).

2. Instead of parametrization with α and β , one can parameterize the distribution with λ and q , where $\alpha = 1 + \lambda$ and $\beta = 1 + q\lambda$. In such a case, the probability density function of the beta distribution is as follows:

$$f_X(x) \propto [x(1-x)^q]^\lambda$$

In Laplacian smoothing, the value of q is chosen so that the ratio of successes to failures is assumed to be $1 : q$ as a prior estimate. The mode of this prior distribution is $1/(1+q)$, which is consistent with this estimate. The value of λ controls the strength of the prior bias. Larger values of λ cause the Bernoulli success parameter to be closer to $1/(1+q)$. The beta distribution is also used as a conjugate prior in cases where a k -way categorical attribute is converted to k Bernoulli attributes by mapping each categorical value to a binary attribute (before applying MAP estimation). If all categories are assumed to have similar frequency, the value of q is selected to be $(k-1)$.

3. Choosing one of the values of (α, β) to be greater than 1 and the other to be less than 1 can lead to an asymptotically infinite mode at one end of the distribution and a minimum density value at the other end. In particular, if α is less than 1 and β is greater than 1, the asymptotically infinite mode will be at $x = 0$. This type of setting is rarely used in probabilistic machine learning.

The beta distribution is implicitly used in Bayes classifiers and the expectation-maximization algorithm (Chapters 8 and 9) by virtue of the use of Laplacian smoothing in these algorithms.

6.8.3 Dirichlet Distribution

The Dirichlet distribution is a multivariate generalization of the beta distribution. Therefore, the Dirichlet distribution models a d -dimensional vector $[x_1, x_2, \dots, x_d]$ in which each $x_i \in (0, 1)$ and the sum of different values of x_i is 1:

$$\sum_{i=1}^d x_i = 1$$

An immediate observation is that this vector would be ideal in modeling the probabilities of the faces of a d -sided die. The Dirichlet distribution can be a conjugate prior of the categorical distribution and it is also a conjugate prior of the multinomial distribution. A popular use of the Dirichlet distribution is for topic modeling, where the word-frequencies in each topic are assumed to be drawn from a multinomial distribution. Therefore, it is assumed that we have a lexicon of d words and a topic-specific die is rolled repeatedly in order to obtain the frequencies of the words in that topic. Such an approach results in multinomial word frequencies for each topic (corresponding to the likelihood). Since the Dirichlet distribution is the conjugate prior of the Multinomial distribution, it is ideal for modeling the prior distribution of the parameter vector corresponding to the probability values of the d faces of the topic-specific die.

The probability density function for a d -dimensional vector $\vec{x} = [x_1, x_2, \dots, x_d]$ drawn from the Dirichlet distribution is as follows:

$$f_{\vec{X}}(\vec{x}) = \begin{cases} \frac{1}{B(\alpha_1, \alpha_2, \dots, \alpha_d)} \prod_{i=1}^d x_i^{\alpha_i - 1} & [\text{if } x_i \geq 0 \text{ and } \sum_{i=1}^d x_i = 1] \\ 0 & \text{otherwise} \end{cases}$$

The d -dimensional beta function $B(\alpha_1, \alpha_2, \dots, \alpha_d)$ is the (inverse) proportionality factor of the above density function. It is defined in such a way that the density function of the Dirichlet distribution integrates to 1 over the entire domain of d -dimensional nonnegative vectors \vec{x} satisfying $\|\vec{x}\|_1 = 1$:

$$B(\alpha_1, \alpha_2, \dots, \alpha_d) = \int_{\vec{x}: \vec{x} \geq 0, \|\vec{x}\|_1=1} \left[\prod_{i=1}^d x_i^{\alpha_i-1} \right] dx_1 dx_2 \dots dx_d$$

Note that the above function is the direct d -dimensional generalization of the 2-dimensional beta function defined in the previous section. This is not particularly surprising, given that the Dirichlet distribution is the multiway generalization of the beta distribution. Interestingly, the beta function can be defined in terms of the gamma function as follows:

$$B(\alpha_1, \alpha_2, \dots, \alpha_d) = \frac{\prod_{i=1}^d \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^d \alpha_i\right)}$$

The aforementioned relationship also holds for the 2-dimensional beta function used in the case of the beta distribution. The scale parameters $\alpha_1 \dots \alpha_d$ are all positive. The Dirichlet distribution is denoted as follows:

$$\vec{X} \sim \text{Dir}(\alpha_1, \alpha_2, \dots, \alpha_d)$$

The mean of the i th component X_i of \vec{X} is as follows:

$$\mu_{X_i} = \frac{\alpha_i}{\sum_{j=1}^d \alpha_j}$$

One can see that the sum of different components of the mean vector will sum to 1 (just like the mean vector). The mode of the distribution exists when each $\alpha_i > 1$; in these cases the mode is $[z_1, z_2, \dots, z_d]$, where $z_i = (\alpha_i - 1)/C$ and $C = (\sum_{i=1}^d \alpha_i) - d$. However, as we will see later, the Dirichlet distribution is typically used in settings where each α_i is less than 1. In such cases, the density increases asymptotically near $\alpha_i = 0$. This is not particularly surprising since the beta distribution is a special case of the Dirichlet distribution.

The variance of the i th component of the Dirichlet distribution can be expressed in terms of the means as follows:

$$\sigma_{X_i}^2 = \frac{\mu_{X_i}(1 - \mu_{X_i})}{1 + \sum_{j=1}^d \alpha_j}$$

The domain of the Dirichlet distribution is the unit cube defined by $[0, 1]^d$.

An important special case of the Dirichlet distribution is the *symmetric* Dirichlet distribution in which each α_i has the same value. The Dirichlet distribution is often used as a conjugate prior of the multinomial distribution, and choosing the symmetric Dirichlet distribution ensures a similar prior bias of each multinomial parameter. A common application of this setting is text data in which the categories of the multinomial distribution might represent words or topics. In such cases, it is common to choose a symmetric Dirichlet distribution as a conjugate prior in which each α_i has the same value and is less than 1.

Since the Dirichlet distribution is a multivariate generalization of the beta distribution, it shares a number of properties of the beta distribution. These properties are particularly useful in the most common application of the Dirichlet distribution in the text domain:

1. It is helpful to examine the behavior of the Dirichlet distribution while thinking of the outcomes in x_1, x_2, \dots, x_d as probabilities of a categorical distribution. Choosing values of $\alpha_1, \alpha_2, \dots, \alpha_k$ to be less than 1 leads to most values of x_1, x_2, \dots, x_d that are close to 0 and a few large values contributing to most of the probability sum of 1. Choosing values of $\alpha_1, \alpha_2, \dots, \alpha_d$ that are greater than 1 leads to a bias towards evenly distributed probabilities in x_1, x_2, \dots, x_d .

The Dirichlet distribution is often used to model prior probabilities of topics in documents or the prior probability of words in topics. It is well known that these types of frequencies are inherently sparse. Therefore, it makes sense to choose values of $\alpha_1, \alpha_2, \dots, \alpha_d$ less than 1. This choice ensures that a prior bias is provided to the multinomial parameters in a manner that encourages sparsity. Typically, a symmetric Dirichlet distribution is used in such applications, where each α_i is set to the same value less than 1. As a result, the prior bias provided by values of the Dirichlet parameters less than 1, causes the MAP estimation of the posterior parameters to be such that only a small number of the multinomial probabilities are nonzero. This type of bias regularizes the optimization so that the multinomial parameters are guided towards values that make sense⁴ in sparse settings. A common application of this setting is *Latent Dirichlet Allocation* [12], which builds a topical distribution from a corpus of documents.

2. The relative levels of sparsity in $[x_1, x_2, \dots, x_d]$ depend on the corresponding values of $\alpha_1, \dots, \alpha_d$. This is evident from the fact that the mean of x_i is proportional to α_i . However, when the parameters $\alpha_1 \dots \alpha_k$ are chosen to be less than 1, this proportionality will be reflected in which specific dimensions will take on significant non-zero values more frequently in the vector \vec{x} . If a symmetric Dirichlet is used, there is no specific preference for any particular vector. One advantage of using the symmetric Dirichlet distribution is that one has to specify a fewer number of Dirichlet hyper-parameters up front during the MAP estimation process.

The Dirichlet distribution also exhibits nonzero covariances among attributes, since it represents a joint distribution. We refer the reader to the bibliographic notes for useful resources on the Dirichlet distribution.

Example 6.20 (Multinomial MAP Estimation with Dirichlet Prior)

Consider a multinomial distribution in d dimensions (with parameters p_1, \dots, p_d) and you are given n d -dimensional vectors $\vec{x}_1 \dots \vec{x}_n$ that are drawn from this distribution. Section 6.3.5 shows how to perform MLE in this setting. What is the MAP estimation of $p_1 \dots p_d$ using the Dirichlet prior with hyper-parameters $\alpha_1 \dots \alpha_d$? Assume that each α_i is at least 1. Discuss why the problem would be much more difficult for values of α_i less than 1.

Solution: The Dirichlet prior on the parameters p_1, p_2, \dots, p_d is as follows:

$$f_{\vec{\Theta}}(p_1, \dots, p_d) = C \prod_{j=1}^d p_j^{\alpha_j - 1}$$

Here, C is a constant independent of $p_1 \dots p_d$ based on the use of the gamma function. Let m_i be the sum of the components of the vector \vec{x}_i for each i . Furthermore, the

⁴This is a different scenario from the use of the Beta(α, β) prior in Laplacian smoothing where $\alpha, \beta > 1$. The Dirichlet optimization is much harder with hyper-parameters in $\vec{\alpha}$ less than 1, because it leads to non-convex optimization. Therefore, techniques like Gibbs sampling are often used [12, 28].

components of each \vec{x}_i are $[x_{i1}, \dots, x_{id}]$. Based on the derivation in section 6.3.5, the multinomial distribution provides the following negative log-likelihood when MLE rather than MAP is used (cf. Equation 6.9):

$$\mathcal{LL}(\vec{x}_1, \dots, \vec{x}_n, p_1, \dots, p_d) = \underbrace{-\ln \left[\prod_{i=1}^n \frac{m_i!}{\prod_{j=1}^d x_{ij}} \right]}_{\text{Constant}} - \sum_{i=1}^n \sum_{j=1}^d x_{ij} \ln(p_j)$$

According to Equation 6.20, the negative log-posterior may be obtained by adding the negative logarithm of the prior to the above expression (ignoring constants). In other words, we have the following expression for the negative log-posterior:

$$\begin{aligned} \mathcal{LP}(\vec{x}_1, \dots, \vec{x}_n, p_1, \dots, p_d) &= \mathcal{LL}(\vec{x}_1, \dots, \vec{x}_n, p_1, \dots, p_d) - \ln(f_{\vec{\theta}}(p_1, \dots, p_d)) + C_0 \\ &= - \sum_{i=1}^n \sum_{j=1}^d x_{ij} \ln(p_j) - \sum_{j=1}^d (\alpha_j - 1) \ln(p_j) + \text{Constant} \end{aligned}$$

In addition, we have the constraint $\sum_{j=1}^d p_j = 1$. Therefore, one must create the Lagrangian relaxation of the above expression to optimize it. This is achieved by adding the term $\lambda(\sum_j p_j - 1)$ to the above expression for free Lagrange parameter λ :

$$\mathcal{LP}(\vec{x}_1, \dots, \vec{x}_n, p_1, \dots, p_d) = - \sum_{i=1}^n \sum_{j=1}^d x_{ij} \ln(p_j) - \sum_{j=1}^d (\alpha_j - 1) \ln(p_j) + \lambda(\sum_j p_j - 1) + \text{Const.}$$

Differentiating the above with respect to p_j and setting it to 0, one obtains the following:

$$\frac{\alpha_j - 1 + \sum_{i=1}^n x_{ij}}{p_j} = \lambda$$

In other words, each p_j is proportional to the following:

$$\hat{p}_j \propto \alpha_j - 1 + \sum_{i=1}^n x_{ij}$$

Since the different values of p_j sum to 1, the optimal estimate is as follows:

$$\hat{p}_j = \frac{\alpha_j - 1 + \sum_{i=1}^n x_{ij}}{\sum_{j=1}^d \alpha_j - d + \sum_{i=1}^n \sum_{j=1}^d x_{ij}} = \frac{\alpha_j - 1 + \sum_{i=1}^n x_{ij}}{\sum_{j=1}^d \alpha_j - d + \sum_{i=1}^n m_i}$$

The similarity of the above expression to Laplacian smoothing is striking. As long as $\alpha_j > 1$, it can also be shown that the objective function is convex and the optimal solution does not lie at the end points $p_j \in \{0, 1\}$.

The approach does not work for $\alpha_j < 1$ because this choice of hyper-parameter(s) results in a non-convex component to the optimization, which is $-(\alpha_j - 1) \ln(p_j)$. This choice yields a sparse solution for $p_1 \dots p_d$, and the solution would typically lie on an edge or face of the d -dimensional cube $[0, 1]^d$ in which $p_1 \dots p_d$ lie. This type of optimization in a bounded cube would also require a large number of end-point checks. In such cases, other techniques like *Gibbs sampling* can be used. ■

6.9 Summary

This chapter introduces various methods for reconstructing probability distributions from data, such as maximum likelihood estimation and maximum a posteriori estimation. In the real world, mixtures of distributions are required in order to popularly model clustered data. The expectation-maximization method for estimating the parameters of mixture models is discussed. The kernel density estimation method is introduced and its application to reconstruction of distributions of arbitrary shape is discussed. When the amount of data is limited, the estimation of the parameters can be poor. This is because the variance of parameter estimation is high when there is limited data, causing random variations reflecting the specific quirks of the data set at hand. The MAP method for reducing reconstruction variance is discussed and a specific example is provided in the context of the estimation of the parameters of the categorical distribution. The bias-variance trade-off is introduced; a solid understanding of this concept can provide guidance in the design of statistical estimation methods in order to reduce the overall error.

6.10 Further Reading

The maximum likelihood method is discussed in detail in [10, 33, 51]. These books also discuss the maximum a posteriori method along with its connections to regularization. A tutorial on maximum likelihood estimation is available in [52]. A compendium of conjugate prior distributions is provided in [21]. Bayesian statistical methods are discussed in [14, 27, 47]. The best source for learning the EM algorithm is the book by McLachian and Krishnan [49]. Kernel density estimation is discussed in detail in the book by Silverman [57]. Fast methods for kernel density estimation are presented in [64]. The beta and the gamma distributions are described in detail in [13]. Several discussions on the use of the Dirichlet distribution for machine learning applications are provided in [12, 28].

6.11 Exercises

1. Consider a family of density functions $f_{X|\Theta=\theta}(x) = \theta + 2x(1 - \theta)$, where $\theta \in [0, 2]$ is the distribution parameter and $x \in (0, 1)$. Set up a condition for finding the maximum likelihood estimate $\hat{\theta} = \hat{\theta}^*$ in terms of observed data points $x_1 \dots x_n$. Show that the optimal numerical value of $\hat{\theta}^* \in [0, 2]$ for $n = 1$ and a single observed point x_1 is $\hat{\theta}^* = 0$ when $x_1 > 0.5$ and is $\hat{\theta}^* = 2$ when $x_1 < 0.5$.
2. A coronavirus test has false positive probability p and false negative probability 0.1. Consider three patients of which one is actually infected. The test predicts two infected patients (although you don't know individual results). Find the MLE of p . Why is the MLE of p greater than the false negative probability?
3. A coronavirus test has false positive probability p_1 and false negative probability p_2 . Consider three patients of which one was actually infected. The test predicts two infected patients (although you don't know individual results). Find the maximum likelihood estimates of both p_1 and p_2 , and show that the MLE occurs in a case where the tests are always wrong. [Hint: While optimizing over an interval, be careful about checking interval end-points.]

4. You sample the points 9.7, 3.4, 11.2, 1.5 from a uniform distribution. What are the maximum likelihood estimates of the two bounds of the uniform distribution?
5. Suppose you roll a die 60 times and the face of 2 shows up 9 times. What is the maximum likelihood estimate of the probability of the face value 2?
6. You toss a biased coin until it shows up heads and count the number of tosses to get to heads. You repeat this process 6 times and you require 4, 3, 1, 2, 6, and 5 tosses. What is the maximum likelihood estimate of the probability of heads?
7. Suppose you roll a biased die 11 times and you obtain the frequencies of the six sides as 2, 1, 3, 1, 2, and 2 times (in the order as face number). Find the maximum likelihood estimates of the probabilities of the die faces.
8. The data samples 6.2, 3.5, 1.6, 4.2, and 7.1 are drawn from the exponential distribution. What is the maximum likelihood estimate of its arrival rate λ ?
9. A Poisson process with arrival rate λ per minute is repeated six times on a window of length three minutes. The six data samples were 3, 5, 3, 2, 5, and 1. What is the maximum likelihood estimate of the arrival rate λ ?
10. Consider the set of points 2.3, 4.1, 4.1, 5.3, which are sampled from a normal distribution. Estimate the maximum likelihood fit of the parameters of the distribution.
11. Consider the following sample of three points (2, 2), (3, 2), and (5, 3). Derive the joint Gaussian distribution under the assumption of dimension independence.
12. Repeat Exercise 11 with a multivariate Gaussian distribution containing dependent attributes.
13. Suppose you roll a (possibly biased) die three times and it never turns up a six. Find the MLE estimate of the probability of a six in the die roll. Compute the Laplacian-smoothed MAP estimate assuming all outcomes are equally likely from the prior belief perspective and $\lambda = 1$. What happens to the MAP estimate when you use $\lambda = 100$?
14. Consider the setting in Exercise 2, in which the false positive probability p has a prior density $f_\Theta(p) = 2 - 2p$ for $p \in (0, 1)$. Find the MAP estimate of the false positive probability. Explain the fact that the MAP estimate is lower than the MLE estimate you obtained in Exercise 2.
15. Let X be a random variable with the following conditional distribution (i.e., conditional on its mean Θ being fixed to θ):

$$f_{X|\Theta=\theta}(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\theta)^2}{2}\right)$$

This unknown mean Θ is itself a random variable drawn from the standard normal distribution $f_\Theta(\theta) \propto \exp(-\theta^2/2)$. The points 2, 3, and 5 are observed as samples of X . Find the MAP and MLE estimates of Θ . Compare the MAP estimate to the MLE estimate and comment on why one of them is less than the other.

16. Modify Exercise 15 so that the prior distribution of Θ has mean 0 and variance 0.1. Find the MAP estimate of Θ and comment on why this estimate is less or more than the estimate obtained in Exercise 15.

17. Let X be a discrete random variable with the following conditional Poisson PMF (i.e., conditional on its parameter Θ being fixed to λ):

$$p_{X|\Theta=\lambda}(x) = \frac{\lambda^x \exp(-\lambda)}{x!}$$

This parameter Θ is itself a random variable drawn from the exponential distribution $f_\Theta(\lambda) = 3\exp(-3\lambda)$. The points 4, 3, 2, 1, and 6 are observed as samples of X . Find the MAP and MLE estimates of Θ . Compare the MAP estimate to the MLE estimate and comment on why one of them is less than the other.

18. Consider the kernel density estimate $\hat{f}_X(z)$ of Equation 6.24. Show that this estimate satisfies the important property of PDFs that $\int_{z=-\infty}^{\infty} \hat{f}_X(z) dz = 1$.
19. Section 6.3.7 discusses the MLE estimation of the Poisson rate parameter λ as $(\sum_{i=1}^n x_i)/n$ for n observed frequency samples x_1, \dots, x_n in one unit of time. Show that if MAP estimation is used with a prior $\text{Gamma}(\alpha, \beta)$ distribution, then the following estimate is obtained for the Poisson rate parameter λ :

$$\hat{\lambda} = \frac{\sum_{i=1}^n x_i + \alpha - 1}{n + \beta}$$

20. Write a program in Python or MATLAB to compute the kernel density estimate at a 2-dimensional grid of points from a 2-dimensional data set. Apply your program to create a surface plot of the data density (like Figure 6.5) on a 2-dimensional data set of your choice. You might find functions like `meshgrid` and `plot_surface` helpful from the `Matplotlib` library in Python. Similar functions are available in MATLAB.
21. Consider a 4-dimensional data set with n points in which each dimension is generated from an i.i.d. exponential distribution with parameter λ . You are given the projection of the data set along the unit vector $[0.5, 0.5, 0.5, 0.5]^T$ as 1-dimensional points x_1, \dots, x_n . How would you recover the MLE of λ ? [Hint: Find the distribution of the data set along the vector. It is not exponential.]
22. The Pareto distribution is used to model slowly decaying tails, and it has the following density function with unknown parameter $\theta > 1$ and known parameter $\min > 0$:

$$f_{X|\Theta=\theta}(x) = \theta \min^\theta x^{-\theta-1} \quad \forall x > \min$$

Suppose that data samples x_1, \dots, x_n belong to the Pareto distribution. Show that the MLE of θ for known value of the parameter \min is $\hat{\theta} = (\sum_{i=1}^n \ln(x_i)/n - \ln(\min))^{-1}$.

23. Repeat Exercise 22 with MAP estimation, where an exponential prior with hyper-parameter λ is used. Explain the effect of the setting of λ on the MAP estimate.
24. The binary data samples x_1, \dots, x_n are drawn from independent tosses of a biased coin with unknown success parameter p . The coin is known to be biased towards success and therefore the prior distribution of p is $f_\Theta(p) = (\lambda + 1)p^\lambda$ for $p \in (0, 1)$ and hyper-parameter $\lambda > 0$. Find an equation that yields the MAP estimate of p and solve it. Discuss the effect of the hyper-parameter λ . Why is this approach a special case of Laplacian smoothing?

- 25.** The binary data samples x_1, \dots, x_n are drawn from independent tosses of a biased coin with unknown success parameter p . The prior distribution of p is $f_\Theta(p) = \lambda \exp(-\lambda p) / (1 - \exp(-\lambda))$ for $p \in (0, 1)$ and hyper-parameter $\lambda > 0$. Find an equation that yields the MAP estimate of p . Is the equation solvable?
- 26.** You go to a casino and other gamblers tell you that a die is loaded so that its face probabilities are $[0.2, 0.2, 0.15, 0.15, 0.15, 0.15]$. You consider this information given by other gamblers to be the equivalent of having observed n_0 rolls of the die yourself but decide to eventually update your estimate of the die face probabilities using the next n die rolls. Discuss how this approach is equivalent to MAP estimation by setting Dirichlet prior hyperparameters in a particular way. What are the values $\alpha_1 \dots \alpha_6$ of the Dirichlet hyperparameters in terms of n_0 ?
- 27. [Fast Kernel Density Estimation]:** The main problem with kernel density estimation is its requirement to scan all points *for each estimation* (which is repetitive). In fast kernel density estimation, the points $x_1 \dots x_n$ are grouped into k tightly knit clusters $\mathcal{G}_1 \dots \mathcal{G}_k$ using an $O(nk)$ clustering algorithm in a single preprocessing phase. The number of points in \mathcal{G}_r is n_r . Then, the kernel density estimate at any $x = a$ is approximated by using the mean-squared distance of a to each cluster:

$$\hat{f}_X(a) = \sum_{r=1}^k \frac{n_r}{n} \frac{1}{\sqrt{2\pi}h} \exp\left(-\frac{\sum_{x_i \in \mathcal{G}_r} (x_i - a)^2}{n_r(2h^2)}\right)$$

Discuss why the density estimation at $x = a$ can be implemented in $O(k)$ time and space by pre-storing only the number of points n_r , first-order sum $\sum_{i \in \mathcal{G}_r} x_i$, and second-order sum $\sum_{i \in \mathcal{G}_r} x_i^2$ in each cluster \mathcal{G}_r . Generalize the approach to multi-dimensional data. [Hint: See Corollary 2.1 and Exercise 18 of Chapter 2.]



Chapter 7

Regression

“Statistics are no substitute for judgement.” — Henry Clay

7.1 Introduction

The regression problem works with pairs of observations $(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots (\vec{x}_n, y_n)$ in order to construct a model that maps each \vec{x}_i to y_i with a functional relationship. Here, each \vec{x}_i is a row vector containing d -dimensional numeric features and y_i is a numerical target. The variables in \vec{x}_i are referred to as the *predictors*, *independent variables*, *feature variables*, or *regressors*. The variable y_i is referred to as the *outcome*, *dependent variable*, *regressand*, or *response* variable. As the name suggests, the dependent variable *depends on* the independent variable in the form of a functional relationship:

$$y_i \approx f(\vec{x}_i)$$

The above equation uses the approximation symbol, because the value of y_i does not exactly match $f(\vec{x}_i)$. The difference $(y_i - f(\vec{x}_i))$ is referred to as *residual error*. The learning is done by using a function $f(\cdot)$ that is parameterized in some pre-defined form, and the parameters are learned in a data-driven manner so that y_i matches $f(\vec{x}_i)$ as closely as possible. The most basic setting of regression is one in which this function is linear, which is defined using a d -dimensional column vector of predictor coefficients $\vec{w} = [w_1, \dots, w_d]^T$, and a scalar variable b . The *linear regression* model defines the function $f(\vec{x}_i)$ as follows:

$$y_i \approx b + \vec{w} \cdot \vec{x}_i^T$$

The parameters in \vec{w} and b are estimated in a data-driven manner. The components of the parameter vector \vec{w} are referred to as *regression coefficients* and b is referred to as the *bias*. The bias variable provides the constant intercept of the regression function. The outcome is predicted to the bias b when the predictor variables are set to 0.

It is helpful to understand the interpretation of the coefficients and the bias with the use of an example. Consider the 1-dimensional linear regression model $y = b + wx$, where x refers to the clock time in seconds since noon, and y refers to the distance from point O of a train moving away from O at speeds that vary over time. Furthermore, the train was not at O at noon when the clock time was 0. In this application, it makes sense to use w as the (average) speed of the train and the bias b as the distance of the train from O at noon. Using linear regression with samples of (x, y) will create such a solution. The variable b corresponds to a consistent *bias* that is always added to the position of the train since noon. Both the regression coefficients and the bias need to be inferred from the samples (\vec{x}_i, y_i) that are available to the analyst. In this particular case, the samples correspond to pairs of time-stamps and train positions. The process of inferring the coefficients and bias from data samples is referred to as *model building* or *learning*. This process is conceptually similar to the process of reconstructing distributions from data (see Chapter 6). This is because the error is modeled by a Gaussian distribution, and learning (\vec{w}, b) reconstructs this distribution.

7.1.1 Chapter Organization

This chapter is organized as follows. The next section introduces the basics of regression. Section 7.3 provides an understanding the relationship between the squared-loss formulation and the probabilistic formulation of linear regression. Section 7.4 discusses the solutions to linear regression. Methods for modeling categorical variables are discussed in section 7.5. The regularization of linear regression to reduce overfitting is discussed in section 7.6. A probabilistic view of regularization is discussed in section 7.7. The evaluation of linear regression is discussed in section 7.8. Non-linear regression methods are introduced in section 7.9. A summary is given in section 7.10.

7.2 The Basics of Regression

An example of the solution to a linear regression problem for a single predictor variable is shown in Figure 7.1(a). This solution is referred to as the *regression line* or *line of best fit*. The goal is to discover a line that aligns well with the observed points (shown by dots), so that plugging in any particular value of the predictor variable yields a robust estimate of the outcome variable. In higher dimensions (i.e., when $d > 1$), the “line” of best fit is really a *hyperplane* of best fit, because we will have multiple parameters $w_1 \dots w_d$ and bias b defining the orientation of the hyperplane in $(d + 1)$ dimensions. An example of the hyperplane of best fit for the case of two predictor variables is illustrated in Figure 7.1(b). The case of $d = 1$ is referred to as *simple regression*, whereas the case of $d > 1$ is referred to as *multiple regression*.

In the simple regression case of Figure 7.1(a), the value of the single coefficient w_1 is 2 and the bias coefficient b is 3. Note that the slope of the line of best fit is 2 and the value of the intercept (bias) is 3. Ideally, we want to select b and w_1 in such a way so that the line of best fit hugs the sample points as closely as possible. This closeness can be quantified with a *loss function*, which is minimized. A common choice of the loss function is as follows:

$$\text{Minimize}_{\vec{w}, b} \sum_{i=1}^n (y_i - b - \vec{w} \cdot \vec{x}_i^T)^2$$

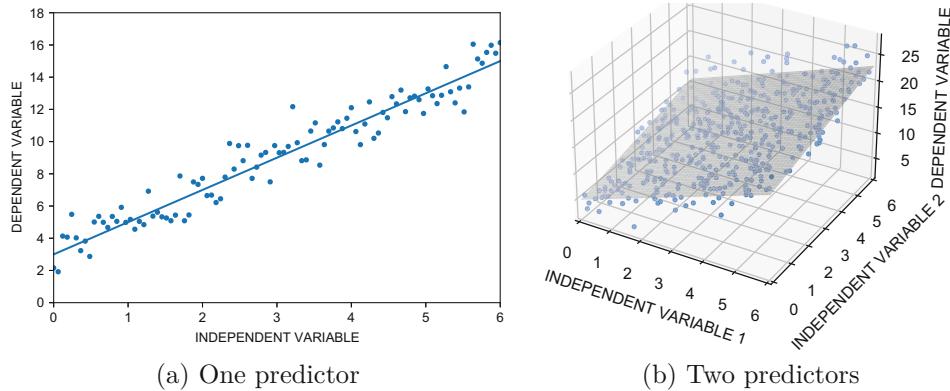


Figure 7.1: Simple examples of linear regression

Although the aforementioned optimization formulation may seem to be unrelated to probability theory, the loss function is itself derived using a probabilistic model. The key point is that the error $\epsilon_i = y_i - b - \vec{w} \cdot \vec{x}_i^T$ is treated as a sample from a Gaussian distribution. As the discussion in this chapter will show, the above loss function is simply a maximum-likelihood estimation problem defined with respect to this Gaussian distribution.

7.2.1 Interpreting the Coefficients

The coefficients of linear regression can be interpreted as the sensitivities of the outcomes to the values of the different predictor variables. In the case of a single predictor (cf. Figure 7.1(a)), the coefficient of the predictor is the slope. Therefore, it indicates how much the outcome changes *on average* by increasing the predictor by one unit. In the case of multiple predictors, the coefficient of a particular predictor is the amount by which the outcome changes when increasing the predictor by 1 and holding other predictors constant. We summarize this interpretation as follows:

Definition 7.1 (Interpretation of Predictor Coefficient) *A predictor coefficient represents how much the outcome variable increases on average, when that predictor is increased by 1, while keeping other predictors constant.*

One can also have a similar interpretation of the bias coefficient:

Definition 7.2 (Interpretation of Bias Coefficient) *The bias coefficient is the average value of the outcome variable when all predictors are set to 0.*

The interpretation of the bias is often complicated by the fact that setting predictors to 0 may not be meaningful. For example, when regressing salary against age, it may not make sense to set the age to 0, as all employees must necessarily be of legal working age. Nevertheless, the bias does provide an understanding of the constant salary offsets that are independent of the age variable.

7.2.2 Feature Engineering Trick for Dropping Bias

In this section, we will introduce a feature engineering trick in order to show that the bias can be dropped by augmenting the features with a single constant feature value. Consider

the pairs (\vec{x}_i, y_i) , where each $\vec{x}_i = [x_{i1}, \dots, x_{id}]$ is a d -dimensional feature variable vector. Throughout this chapter, it will be assumed that the feature vector \vec{x}_i is a row vector. One can add a single 1 as the $(d+1)$ th feature $x_{i(d+1)} = 1$ to create the expanded feature set $\vec{x}'_i = [x_{i1}, \dots, x_{i(d+1)}]$, and similarly we extend the d -dimensional coefficient vector \vec{w} with an additional coefficient to create the expanded coefficients $\vec{w}' = [w_1, \dots, w_{d+1}]^T$. With this expanded feature set, the linear regression problem may be defined without an explicit bias as follows:

$$y_i = \vec{w}' \cdot [\vec{x}'_i]^T + \epsilon_i \quad (7.1)$$

One can see that using the new set of feature variables and coefficients (without an explicit bias) is equivalent to setting the bias b to w_{d+1} with the original set of features and coefficients:

$$b = w_{d+1}$$

This is because one can add $\vec{w} \cdot \vec{x}'_i^T$ to both sides of the above equation, which yields the following:

$$\begin{aligned} \sum_{j=1}^d w_j x_{ij} + b &= \sum_{j=1}^d w_j x_{ij} + w_{d+1}(1) \\ \vec{w} \cdot \vec{x}'_i^T + b &= \vec{w}' \cdot [\vec{x}'_i]^T \end{aligned}$$

The left-hand side corresponds to the linear regression model with an explicit bias, whereas the right-hand side corresponds to the linear regression model without an explicit bias but an expanded feature set in which a single feature value of 1 is appended to each observation. The coefficient of this feature value yields the bias. Throughout this chapter, we will present linear regression without an explicit bias variable because it simplifies the presentation and reduces notational clutter by getting rid of b . Furthermore, without loss of generality, we can also assume that the features (including the additional feature with value 1) are d -dimensional rather $(d+1)$ -dimensional simply by selecting the definition of the dimensionality d appropriately. Therefore, the subsequent discussion will make the following assumption for the probabilistic errors:

$$\mathcal{E} = Y - \vec{w} \cdot \vec{X}^T \sim \mathcal{N}(0, \sigma^2)$$

Here (\vec{X}, Y) are random variables defining the distribution of which (\vec{x}_i, y_i) are independent samples. Note that b is missing in this relationship because it has been replaced with a single deterministic variable of value 1 in \vec{X} . The joint distribution of \vec{X} now contains a single deterministic variable whose coefficient models the bias. One can also restate this relationship at the sample level:

$$y_i = \vec{w} \cdot \vec{x}_i^T + \epsilon_i$$

The error values ϵ_i are independent samples from a Gaussian distribution.

Example 7.1 (Importance of Bias) Consider a regression problem with 1-dimensional regressor x_i and regressand y_i , so that the pairs (x_i, y_i) are $\{(1, 2), (2, 3), (3, 4), (4, 5), (5, 6)\}$. Do you see a pattern of the relationship between x_i and y_i ? Use inspection to guess a reasonable linear relationship between x_i and y_i . Discuss why bias is important for getting an accurate prediction of y_i for arbitrary x_i .

Solution: The value of y_i is always 1 more than x_i . Therefore, a reasonable linear relationship is $y_i = x_i + 1$. A bias is critical in this case because the linear relationship does not pass through the origin. In other words, one will always have errors in prediction for a relationship of the form $y_i = w_1 x_i + 0$, irrespective of what value of w_1 is chosen. ■

7.2.3 Regression: A Central Problem in Statistics and Linear Algebra

Regression is perhaps one of the most important problems in machine learning, as it serves as a “bridge problem” between linear algebra, statistics, and machine learning. From the linear algebra perspective, the regression problem is a generalization of the problem of solving linear equations. When systems of equations are inconsistent, linear regression finds a natural way of providing the solution that is the best fit to this inconsistent system of equations. The notion of “best fit” refers to the fact that the aggregate error in satisfying the system of equations is as small as possible.

Consider the following system of linear equations as a problem that arises often in numerical linear algebra:

$$\begin{aligned} 2w_1 + 3w_2 &= 4 \\ 3w_1 + 4w_2 &= 5 \\ 5w_1 + 7w_2 &= 6 \end{aligned}$$

This system of equations is inconsistent, because adding the first two equations and subtracting the third equation yields 0 on the left-hand side and 3 on the right-hand side. It is clear that there is no way to select any particular choice of w_1 and w_2 for which all three equations will hold. The best one can do is to penalize non-satisfaction of equality with a squared function J and find the values of $[w_1, w_2]$ that optimize the aggregate error:

$$J = (2w_1 + 3w_2 - 4)^2 + (3w_1 + 4w_2 - 5)^2 + (5w_1 + 7w_2 - 6)^2$$

Finding the coefficients of best fit by minimizing J has an identical form as that of the squared loss formulation in linear regression.

From the probability and statistics perspective, linear regression is a classical maximum-likelihood estimation problem in which the error $\epsilon_i = y_i - \vec{w} \cdot \vec{x}_i$ is modeled from a Gaussian distribution. The loss function of regression turns out to be equivalent to the negative log-likelihood function of reconstructing this distribution.

From the machine learning perspective, the regression problem is primarily viewed as a predictive application in which we want to predict missing outcome values for samples of d -dimensional predictor variables. Such instances are referred to as *out-of-sample instances* because they are not present in the original data instances (\vec{x}_i, y_i) , which were used to compute the vector \vec{w} . For example, consider a d -dimensional vector \vec{z} of predictors (not included in the original data of predictor-outcome pairs). Therefore, only the predictors \vec{z} are known in this case, and the missing outcome value \hat{y} can be predicted as follows:

$$\hat{y} = \vec{w} \cdot \vec{z}^T$$

Note the circumflex symbol on the outcome variable, indicating that it is a predicted value. The original data instances for learning \vec{w} are referred to as *training data*, whereas the out-of-sample instances on which predictions are made are referred to as *test data*. The term

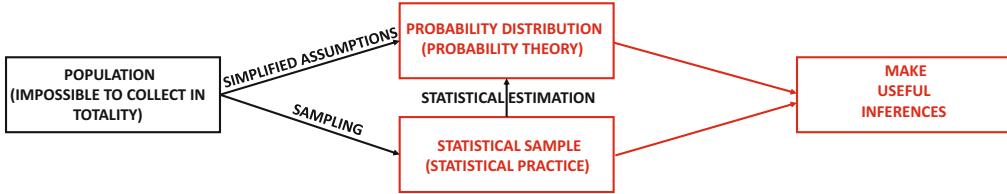


Figure 7.2: Revisiting Figure 1.6: How reconstructed distributions can be used for making useful inferences in machine learning.

“training data” refers to the fact that it is used for “teaching” the model useful data-driven information such as the coefficient vector \vec{w} . From an application-centric point of view, many machine learning models like regression and classification are primarily useful for making predictions on out-of-sample test data, because the outcomes for samples in the training data are already known. The regression problem provides the first concrete example of how reconstructed distributions are leveraged to make useful inferences (cf. Figure 7.2) — in this case, a 1-dimensional normal distribution is imposed on the error of prediction.

7.3 Two Perspectives on Linear Regression

In this section, we will introduce two perspectives on linear regression, which are equivalent. These two perspectives correspond to the linear-algebra-based perspective and the probabilistic perspective. The linear algebra perspective is easier to understand intuitively, because it works with a loss function penalizing the errors in prediction. Therefore, we will first present the linear algebra perspective. Later, we will present the maximum-likelihood perspective on linear regression and show that the (seemingly heuristic) loss-function of linear regression is actually derived from principles of maximum-likelihood estimation.

7.3.1 The Linear Algebra Perspective

The loss function perspective on linear regression defines the line of best fit in terms of optimizing the mean-squared vertical distance of the points to the hyperplane of best fit. In other words, we want to minimize the following loss function:

$$\text{Minimize}_{\vec{w}} J = \sum_{i=1}^n \epsilon_i^2 = \sum_{i=1}^n (y_i - \vec{w} \cdot \vec{x}_i^T)^2$$

This objective function can be viewed as a generalization of the problem of solving a set of linear equations. Imagine that we are given points (\vec{x}_i, y_i) that create a linear system of equations of the form $\vec{w} \cdot \vec{x}_i^T = y_i$. The components of \vec{w} are the unknowns of this linear system. Since there are n points, we have a total of n equations. If the system of equations were to be consistent, a set of coefficients \vec{w} can be found for which all linear conditions are satisfied with strict equality (i.e., $\epsilon_i = 0$). In such a case, the optimal objective function value of J is 0. Therefore, a consistent system of equations can be solved using the optimization formulation of linear regression, and it corresponds to the case where the hyperplane of best fit (cf. Figure 7.1) passes through all the points (\vec{x}_i, y_i) with an objective function value of 0.

However, the generic version of linear regression is naturally designed for cases where the number of samples n is greater than the dimensionality d and the system of equations is *over-determined*. In other words, the resulting system of equations is inconsistent and no valid solution can be found that satisfies all the constraints. From the perspective of Figure 7.1, the over-determined case corresponds to the situation where we can no longer find a hyperplane passing through all the points and would therefore have to make do with one that passes as closely to the points as possible on average. This desideratum is enforced with the use of a loss function that minimizes the sum of the squares of the residuals.

Although we are finding a hyperplane that hugs the given data samples as “closely” as possible, the distance between the samples and the hyperplane is measured along the direction of the outcome variable (because ϵ_i is defined along the direction of y_i and uses the same units). The key point is that predictive applications are focused on the all-important outcome variable, and the coefficients in \vec{w} will be computed by the optimization process in such a way that the outcome variable is predicted as accurately as possible. For example, in the case of a single outcome variable, a line of best fit is not optimized to perform the reverse task of predicting the predictor variable (for $d = 1$) from the outcome variable, although it often does a good job when the variances of the predictor and outcome are similar.

Example 7.2 Is it true or false that if the value of n (number of training samples) is less than d (the number of predictor variables including the engineered feature with value 1), a solution to the linear regression problem will always exist with zero error? Either prove the result or give a counterexample.

Solution: The main intuition for answering the above question is to ask yourself whether a system of two equations in three unknowns always has at least one valid solution. The problem of linear regression is a generalization of the problem of solving a system of linear equations in the coefficient vector \vec{w} , with nonzero error indicative of inconsistency in the underlying equation system. It is indeed possible to create an invalid linear system with more unknowns than variables.

Consider the following $n = 3$ feature vectors in $d = 4$ dimensions (including the engineered feature value of 1 at the end):

$$\vec{x}_1 = [1, 2, 3, 1], \vec{x}_2 = [2, 4, 6, 1], \vec{x}_3 = [3, 6, 9, 1]$$

The corresponding values of the dependent variables are as follows:

$$y_1 = 1, y_2 = 2, y_3 = 100$$

For the above data, it is evident that $(\vec{x}_1 + \vec{x}_3) = 2\vec{x}_2$. Taking the dot product of both sides with the coefficient vector \vec{w} would imply that $y_1 + y_3 = 2y_2$ (if the system has zero error). However, this is not the case for the data at hand in which $(y_1 + y_3) = 101$ and $2y_2 = 4$. Therefore, this counter-example shows that even when $n < d$, a zero-error solution may not exist. However, this type of counter-example is rare in practice. In most data sets from the real world, an infinite number of zero-error solutions generally do exist when $n < d$. ■

7.3.2 The Probabilistic Perspective

The least-squares loss function of linear regression might seem like a good heuristic objective function of first glance, and many practitioners of data science do tend to view it in this way. However, viewing the loss function of linear regression in a merely heuristic manner does not do it justice, as it is derived from formal probabilistic principles. The probabilistic perspective of linear regression formulates it as a maximum-likelihood estimation problem in which the parameters in \vec{w} are instantiations of random variables that need to be estimated in a data-driven manner. Each residual $\epsilon_i = y_i - \vec{w} \cdot \vec{x}_i$ is drawn from a normal distribution with zero mean and variance σ^2 ; this random variable is denoted by \mathcal{E} . If \vec{X} and Y be the predictor and outcome random variables respectively, we have:

$$\mathcal{E} = Y - \vec{w} \cdot \vec{X}^T \sim \mathcal{N}(0, \sigma^2)$$

Since the errors are assumed to follow a Gaussian distribution, one can use the Gaussian probability density function to explicitly quantify the likelihood that a predictor-outcome pair in the observed data was generated from this distribution:

$$\begin{aligned} f_{\mathcal{E}|\vec{W}=\vec{w}}(\epsilon_i) &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\epsilon_i^2}{2\sigma^2}\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \vec{w} \cdot \vec{x}_i^T)^2}{2\sigma^2}\right) \end{aligned}$$

Note that the parameter vector \vec{w} is an instantiation of the random variable \vec{W} . The likelihood fit $\mathcal{L}(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n, \vec{w}, \sigma^2)$ of linear regression over the entire training data set is the product of these density functions over all the points, which yields the following:

$$\begin{aligned} \mathcal{L}(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n, \vec{w}, \sigma^2) &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \vec{w} \cdot \vec{x}_i^T)^2}{2\sigma^2}\right) \\ &= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\sum_{i=1}^n \frac{(y_i - \vec{w} \cdot \vec{x}_i^T)^2}{2\sigma^2}\right) \end{aligned}$$

On taking the negative logarithm of both sides, we obtain the negative log-likelihood fit as follows:

$$\begin{aligned} \mathcal{LL}(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n, \vec{w}, \sigma^2) &= \frac{n}{2} \cdot \ln(2\pi) + n \cdot \ln(\sigma) + \sum_{i=1}^n \frac{(y_i - \vec{w} \cdot \vec{x}_i^T)^2}{2\sigma^2} \\ &= H(\sigma) + \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \vec{w} \cdot \vec{x}_i^T)^2 \end{aligned}$$

It is evident that no matter what the optimal value of σ might be, the objective function can be minimized only when the quantity $\sum_{i=1}^n (y_i - \vec{w} \cdot \vec{x}_i^T)^2$ in the second term is minimized. This objective function is exactly the same as the squared-loss function discussed in the previous section. Therefore, *the maximum-likelihood estimation of the coefficients of linear regression with Gaussian errors is equivalent to finding linear coefficients that minimize the sum of the squares of the residuals*.

Lemma 7.1 *The problem of maximum-likelihood estimation of the coefficients of linear regression with zero-centered Gaussian errors is equivalent to finding the linear coefficients that minimize the sum of squares of the residuals.*

It remains to determine the maximum-likelihood estimate of the parameter σ . On differentiating the negative log-likelihood with respect to σ and setting it to zero, we obtain the following:

$$\frac{n}{\sigma} - \sum_{i=1}^n \frac{(y_i - \vec{w} \cdot \vec{x}_i^T)^2}{\sigma^3} = 0$$

This yields the following maximum-likelihood estimate for σ :

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^n (y_i - \vec{w} \cdot \vec{x}_i^T)^2}{n}$$

This is a biased estimate of the average squared residual, and the unbiased estimate uses $(n - d)$ in the denominator instead of n . This type of adjustment is similar to the Bessel correction in the measurement of variance.

The loss function of regression problem has a probabilistic interpretation. This theme is repeated in subsequent chapters. Most loss functions in machine learning formulations have probabilistic interpretations. This is also true of many classification models that are essentially modifications of the regression framework discussed in this chapter — an example is the *logistic regression* model discussed in the next chapter.

7.3.2.1 Example: Regression with L_1 -Loss

In order to understand how different loss functions in the regression/classification family of models map to maximum-likelihood estimation on different probability distributions from the exponential family, we will consider the problem of linear regression with L_1 -loss. The L_1 -regression problem also uses the same predictive relationship between the predictor and outcome as squared-loss regression:

$$y_i = \vec{w} \cdot \vec{x}_i^T + \epsilon_i$$

However, instead of minimizing the sum-of-squares of the residuals, we now minimize the sum of the *absolute values* of ϵ_i :

$$\text{Minimize}_{\vec{w}} J = \sum_{i=1}^d \|\vec{w} \cdot \vec{x}_i^T - y_i\|_1$$

Note that L_1 -regression yields very similar results to squared-loss regression, except that the optimal regression hyperplane is less sensitive to outliers. The reason is that squared error causes outliers to have unusually large weights. Therefore, the optimal solution for \vec{w} tries to adjust itself to reduce the contributions of outliers at the expense of other (more representative) points. Examples of the optimal regression lines for L_1 -loss and L_2 -loss in the presence of a single outlier point at $(6, 15)$ are shown in Figure 7.3. It is evident that the best-fit line for squared-loss is heavily influenced by the single outlier point, and it does not reflect the trend implied by most of the other points in the data. On the other hand, L_1 -loss regression does a much better job of reflecting the aggregate trends in the data. Since outliers are often caused by various types of unusual events (including errors in data collection), it is inadvisable to allow the regression line to be unduly influenced by such data points.

An equivalent way of formulating L_1 -loss regression is to assume that the errors are drawn from the *Laplace distribution* (instead of the Gaussian distribution used for squared

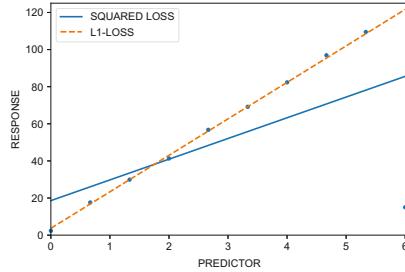


Figure 7.3: A comparison of squared-loss regression with L_1 -loss regression in the presence of a single outlier

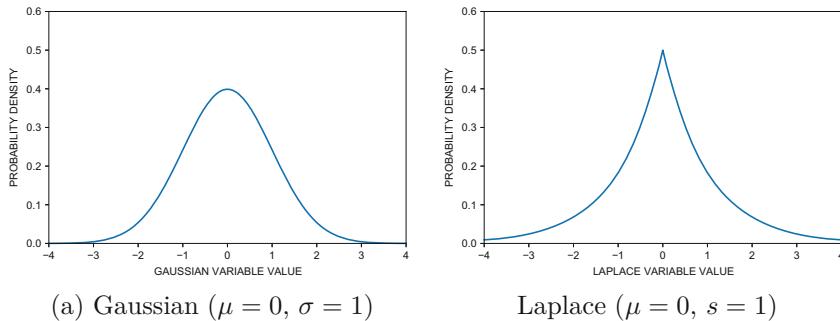


Figure 7.4: A comparison of the Gaussian and Laplace distributions. Both distributions are centered at 0, and have their respective dispersion parameters set to 1.

loss). The Laplace distribution is sometimes referred to as the *double exponential distribution*. The density function of the Laplace distribution uses a location/mean parameter μ and a scale factor s . The scale parameter is analogous to the Gaussian standard deviation, and it is equal to the mean absolute deviation of the Laplace distribution. The probability density function of the Laplace distribution is defined as follows:

$$f_Z(z) = \frac{1}{2s} \exp\left(-\frac{\|z - \mu\|}{s}\right)$$

The membership of random variable Z to the Laplace distribution is denoted by $Z \sim \text{Laplace}(\mu, s)$. An example of the 1-dimensional Laplace distribution along with its comparable Gaussian counterpart is illustrated in Figure 7.4. The main assumption in L_1 -regression is that the error variable follows the Laplace distribution with mean of 0:

$$\mathcal{E} = Y - \vec{w} \cdot \vec{X}^T \sim \text{Laplace}(0, s)$$

Therefore, the residual $\epsilon_i = y_i - \vec{w} \cdot \vec{x}_i^T$ is drawn from the following probability density function:

$$f_{\mathcal{E}|\vec{W}=\vec{w}}(\epsilon_i) = \frac{1}{2s} \exp\left(-\frac{\|\epsilon_i\|_1}{s}\right)$$

A key result is that the maximum-likelihood estimation of the parameter vector \vec{w} with a Laplace error distribution is the same problem that of finding the vector \vec{w} that minimizes the aggregate L_1 -loss $J = \sum_{i=1}^n \|\vec{w} \cdot \vec{x}_i^T - y_i\|_1$ over all training instances.

Lemma 7.2 (Maximum-Likelihood Estimate of Laplace Errors) *When the regression errors $\epsilon_i = y_i - \vec{w} \cdot \vec{x}_i^T$ are assumed to be drawn from the Laplace distribution with location parameter 0 and scale parameter s , the maximum-likelihood estimate of \vec{w} is obtained by minimizing the aggregate L_1 -loss $J = \sum_{i=1}^n \|\vec{w} \cdot \vec{x}_i^T - y_i\|_1$ over all training instances. The maximum-likelihood estimate of the scale parameter is the mean absolute deviation of the error over all training instances.*

Proof: The proof of this result is similar to that of squared-loss regression. The likelihood fit is the product of Laplace density functions over all point-specific errors, which yields the following:

$$\begin{aligned}\mathcal{L}(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n, \vec{w}, s) &= \prod_{i=1}^n \frac{1}{2s} \exp\left(-\frac{\|y_i - \vec{w} \cdot \vec{x}_i^T\|_1}{s}\right) \\ &= \left(\frac{1}{2s}\right)^n \exp\left(-\sum_{i=1}^n \frac{\|y_i - \vec{w} \cdot \vec{x}_i^T\|_1}{s}\right)\end{aligned}$$

On taking the negative logarithm of both sides, we obtain the negative log-likelihood fit as follows:

$$\begin{aligned}\mathcal{LL}(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n, \vec{w}, s) &= n \cdot \ln(2s) + \sum_{i=1}^n \frac{\|y_i - \vec{w} \cdot \vec{x}_i^T\|_1}{s} \\ &= n \cdot \ln(2s) + \frac{1}{s} \sum_{i=1}^n \|y_i - \vec{w} \cdot \vec{x}_i^T\|_1 \\ &= n \cdot \ln(2s) + \frac{1}{s} L_1\text{-Loss}\end{aligned}$$

The scale parameter s is always assumed to be positive, and therefore the above expression is minimized when the L_1 -loss is minimized.

In order to determine the optimum value of the scale parameter, the negative log-likelihood function may be differentiated with respect to s and set to 0 to show the following:

$$\hat{s} = \frac{1}{n} \sum_{i=1}^n \|y_i - \vec{w} \cdot \vec{x}_i^T\|_1$$

The above estimate of s is a biased underestimate, as it will often yield a value of 0 when $n \leq d$. As in the case of squared-loss regression, the scale parameter s is not needed for prediction on new observations (and is also not included in the loss-centric formulation of the problem).

Problem 7.1 (L_p -Regression) *Consider a distribution centered at 0, whose probability density function is of the following form:*

$$f_{\epsilon}(\epsilon) \propto \frac{1}{s} \exp\left(-\frac{\|\epsilon\|^p}{p \cdot s^p}\right)$$

Here, s is a dispersion parameter. Note that the (zero-centered) Laplace and Gaussian distributions are special cases of this framework with $p = 1$ and $p = 2$, respectively. Show that when this distribution is used to model the errors in linear regression, the maximum-likelihood estimation of the errors is the same as finding the linear coefficients that minimize an L_p -loss function.

7.4 Solutions to Linear Regression

Linear regression is one of the few optimization problems in machine learning that has a solution in closed form. However, it can also be solved using gradient-descent methods. In this section, we will explore both these solutions.

7.4.1 Closed-Form Solution to Squared-Loss Regression

The closed-form solution to linear regression is best elucidated by reformulating the objective function in terms of matrices constructed from the training examples (\vec{x}_i, y_i) . Let D be an $n \times d$ data matrix whose i th row contains the row vector \vec{x}_i . Similarly, let \vec{y} be an n -dimensional column vector whose i th entry is y_i . Let \vec{w} be the d -dimensional column vector of regression coefficients and let $\vec{\epsilon}$ be an n -dimensional column vector of residuals in which the i th entry is $\epsilon_i = y_i - \vec{w} \cdot \vec{x}_i^T$. Then, it is easy to see that the residual vector is related to the data matrix D and outcome vector \vec{y} as follows:

$$\vec{\epsilon} = \vec{y} - D\vec{w}$$

Since the loss function of linear regression minimizes the sum of squares of the residuals, it follows that the objective function may be expressed as follows:

$$\text{Minimize}_{\vec{w}} \|\vec{y} - D\vec{w}\|^2 = \|D\vec{w} - \vec{y}\|^2$$

This objective function is convex in \vec{w} and therefore setting its gradient to the zero vector suffices to find the value of \vec{w} that yields a global minimum. We express this gradient in matrix calculus notation¹ in the denominator layout:

$$\frac{\partial J}{\partial \vec{w}} = 2 D^T (D\vec{w} - \vec{y}) = \vec{0}$$

Rearranging the optimality condition, one obtains the following:

$$(D^T D)\vec{w} = D^T \vec{y}$$

For now, let us assume that the matrix $D^T D$ is invertible. In such a case, the solution to the column vector \vec{w} is as follows:

$$\vec{w} = (D^T D)^{-1} D^T \vec{y}$$

Given this solution vector \vec{w} , the out-of-sample test instance \vec{z} can be predicted into its outcome as $\vec{w} \cdot \vec{z}^T$. If an $n_t \times d$ out-of-sample data matrix D_t is available, the n_t -dimensional prediction vector \vec{y}_t can be obtained using the following relationship:

$$\vec{y}_t = D_t \vec{w}$$

The above results assume that $D^T D$ is invertible, which occurs if and only if D has linearly independent columns [6]. In the event that D does not have linearly independent columns, a *uniquely* optimal solution to this problem does not exist. However, one can still find one of the optimal solution vectors for \vec{w} using a variety of methods² including gradient descent

¹See [6] for a discussion of matrix calculus and different types of layouts of vector derivatives.

²An infinite number of optimal solutions may exist. A closed-form solution can be found using the *Moore-Penrose pseudoinverse*. This solution is one of the alternative optima and has some desirable properties so that predictions on *out-of-sample* test data are often more accurate than other solutions.

(cf. section 7.4.3). In such cases, the solution vector to linear regression often does not give good predictions on unseen test instances. The lack of invertibility is often caused by too few observations, which tends to make the model highly specific to the training sample at hand. This is an indicator of high variance, which causes high levels of inaccuracy. In such cases, one is forced to add *regularization* to the model in order to reduce the variability of predictions. This will be the topic of discussion in a later section. A particularly notable case is when the matrix D is square and invertible. In such a case, the solution to regression simplifies greatly, as shown in the following example:

Example 7.3 Suppose that the matrix D is square and invertible. Show that the solution to linear regression simplifies to $\vec{w} = D^{-1}\vec{y}$.

Solution: Using invertibility, one can simplify the solution as follows:

$$\vec{w} = (D^T D)^{-1} D^T \vec{y} = D^{-1}[(D^T)^{-1} D^T] \vec{y} = D^{-1} \vec{y}$$

■

As discussed earlier, the residuals ϵ_i are distributed according to a Gaussian distribution with zero mean at the population level. What about the n *specific* samples of ϵ_i that are returned on the samples used to construct the model? Do they also sum to zero *at the sample level*? It turns out that they *do* indeed sum to 0.

Lemma 7.3 (Residuals Sum to Zero at Sample Level) *The optimal solution to linear regression constructed on a sample of n points is always such that the n residuals obtained by evaluating the predictions on the original sample sum to 0.*

Proof: We will proceed via proof by contradiction. Consider a situation where such an optimal solution \vec{w}_0 exists so that the residuals $\epsilon_1 \dots \epsilon_n$ sum to $a \neq 0$ for the optimal solution \vec{w}_0 . Note that one of these coefficients, corresponding to the engineered feature of 1, is a bias coefficient that is included within \vec{w}_0 in lieu of an explicit bias variable b (cf. section 7.2.2). Now, suppose we modify the bias coefficient of \vec{w}_0 to create \vec{w}_1 , so that the bias coefficient of \vec{w}_1 is less than that of \vec{w}_0 by a/n . Then, the new residuals $\epsilon'_1 \dots \epsilon'_n$ do sum to 0 and we have $\epsilon_i = \epsilon'_i + a/n$. Now consider the sum of squares of the residuals of the solution \vec{w}_0 :

$$\begin{aligned} \sum_{i=1}^n \epsilon_i^2 &= \sum_{i=1}^n (\epsilon'_i + a/n)^2 = \sum_{i=1}^n [\epsilon'_i]^2 + (2a/n) \underbrace{\sum_{i=1}^n \epsilon'_i}_{0} + a^2/n \\ &= \sum_{i=1}^n [\epsilon'_i]^2 + a^2/n \end{aligned}$$

In other words, the sum of squares of the residuals of \vec{w}_1 is lower than that of \vec{w}_0 . Therefore, \vec{w}_0 cannot be optimal, and we have a contradiction. In other words, our original assumption of the residuals not summing to 0 must have been incorrect. ■

The fact that the residuals sum to zero implies that the optimal hyperplane of linear regression passes through the mean of the training sample. One can show this relationship by separately averaging the left-hand and right-hand sides of $y_i = \vec{w} \cdot \vec{x}_i^T + \epsilon_i$ for all i (while observing that average of the different ϵ_i is 0). Therefore, the sample means $\hat{\mu}_Y$ and $\hat{\mu}_X$ of

the dependent and independent variables satisfy the following relationship:

$$\hat{\mu}_Y = \vec{w} \cdot \hat{\mu}_X^T$$

Corollary 7.1 *The optimal solution to linear regression exactly passes through the means of the predictor and outcome variables at the sample level.*

It is noteworthy that this result is true only for the most basic version of linear regression without regularization. It does not hold for regularized models.

Example 7.4 Consider a single-predictor regression problem in which the addition of an engineered feature value of 1 results in an $n \times 2$ data set D in which the second column contains values of 1. The n -dimensional outcome vector is denoted by \vec{y} . Suppose that each 2-dimensional point in the data set is rotated clockwise by 30° to create a new $n \times 2$ data set D_r as follows:

$$D_r = D \begin{bmatrix} \cos(30) & -\sin(30) \\ \sin(30) & \cos(30) \end{bmatrix}$$

Let \vec{w} and \vec{w}_r be the optimal solutions to linear regression for the feature-outcome pairs (D, \vec{y}) and (D_r, \vec{y}) , respectively. Show using the closed-form solution to linear regression that the column vector \vec{w}_r can be obtained from \vec{w} using a 30° clockwise rotation as well. Do the predictions on test-data matrix D_t change by the rotation of the training data matrix D (assuming that the test data matrix D_t is rotated by 30° as well)?

Solution: Let the rotation matrix be denoted by R . Then, the solution to linear regression is given by the following:

$$\vec{w}_r = (D_r^T D_r)^{-1} D_r^T \vec{y}$$

On substituting $D_r = DR$, we obtain the following:

$$\vec{w}_r = ([DR]^T DR)^{-1} (DR)^T \vec{y} = (R^T D^T DR)^{-1} R^T D^T \vec{y}$$

The inverse of the product of matrices is the product of the inverse of the matrices after reversing the order of multiplication. Furthermore, rotation matrices satisfy $R^{-1} = R^T$. Using these two facts, we obtain the following:

$$\vec{w}_r = (\underbrace{R^{-1}}_{R^T} (D^T D)^{-1} \underbrace{(R^T)^{-1} R^T}_{I} D^T \vec{y}) = R^T \underbrace{(D^T D)^{-1} D^T}_{\vec{w}} \vec{y}$$

Therefore, the optimal solution \vec{w}_r is related to the original solution as $\vec{w}_r = R^T \vec{w}$. Since \vec{w} is a column vector, pre-multiplication by R^T corresponds to a clockwise rotation by 30° , just as post-multiplication with R corresponds to a clockwise rotation of 30° for the row vectors of D . Therefore, the original solution \vec{w} is also rotated clockwise by 30° to create the modified solution \vec{w}_r . This solution makes intuitive sense because rotating both the data vectors and \vec{w} by the same angle does not change the dot product between them. After all, the optimization formulation of regression can be expressed in terms of dot products as the problem of minimizing $\sum_i (\vec{w} \cdot \vec{x}_i^T - y_i)^2$.

The predictions of the rows in D_t are given by $D_t \vec{w}$. These predictions do not change by transforming the training and test representations. This is because the test data is rotated to $D_t R$ and the prediction vector $(D_t R) \vec{w}_r$ can be expressed as follows:

$$D_t R \vec{w}_r = D_t \underbrace{(R R^T)}_I \vec{w} = D_t \vec{w}$$

This problem is important from a linear algebra perspective. It provides us the insight that the basis of representation of the feature vectors $\vec{x}_1 \dots \vec{x}_n$ is irrelevant to the predictions found by the model (as long as the same basis is used for the test data). Therefore, rotating the data vectors by 30° has no effect on the predictions. A basis change of the feature vectors is matched by a corresponding basis change of the coefficient vector so that the predictions on the test data remain unchanged. In fact, any type of affine transformation on the feature vectors \vec{x}_i such as scaling or translation does not affect the prediction results as long as the same transformations are performed on the test data. This result is only true for purely linear models like regression. It is generally not true for other models in machine learning that have some level of nonlinearity. ■

Example 7.5 Consider the single predictor problem in which the independent-dependent pairs are as follows:

$$\{(4, 5), (3, 4), (2, 3), (1, 2), (-1, 0), (-2, -1), (-3, -2), (-4, -3), (1, 0), (-1, 2)\}$$

Find the optimal solution to linear regression in the form $y = mx + b$, where x is the single predictor and b is the bias. Technically, the problem has two predictors (including the engineered predictor that captures the bias) although only one of them is non-trivial and is a part of the data.

Solution: The above data is presented in pairs (x_i, y_i) including the dependent variable as well. The data does not include the feature-engineered variable for bias. We compute the 10×2 independent variable data matrix D whose rows contain the independent variable row vectors (including the engineered variable value of 1 for the bias). The rows of the 10×2 matrix D are the 2-dimensional tuples $\vec{x}_i = [x_i, 1]$ of the set below in that specific order:

$$\{[4, 1], [3, 1], [2, 1], [1, 1], [-1, 1], [-2, 1], [-3, 1], [-4, 1], [1, 1], [-1, 1]\}$$

One can show that the matrix $D^T D$ and its inverse are as follows:

$$D^T D = \begin{bmatrix} 62 & 0 \\ 0 & 10 \end{bmatrix} \quad (D^T D)^{-1} = \begin{bmatrix} 1/62 & 0 \\ 0 & 1/10 \end{bmatrix}$$

The dependent variable vector $\vec{y} = [5, 4, 3, 2, 0, -1, -2, -3, 0, 2]^T$ is used to compute $D^T \vec{y}$:

$$D^T \vec{y} = \begin{bmatrix} 58 \\ 10 \end{bmatrix}$$

One can then compute the weight vector as follows:

$$\vec{w} = (D^T D)^{-1} D^T \vec{y} = \begin{bmatrix} 58/62 \\ 1 \end{bmatrix}$$

One can then use this weight vector to define the optimal equation of linear regression as follows:

$$y_i = \frac{58}{62}x_i + 1 + \epsilon_i$$

Note that the regression line passes through the mean of the (non-trivial) independent variable and dependent variable, which is the pair $(0, 1)$. We leave it as an exercise for the reader to verify that the residuals ϵ_i sum to 0. ■

We encourage the reader to work out the following problem in which the data for the dependent and independent variables are switched.

Problem 7.2 Consider a variation of the single non-trivial predictor problem in Example 7.5 in which the data for the dependent and independent variables have been switched as follows:

$$\{(5, 4), (4, 3), (3, 2), (2, 1), (0, -1), (-1, -2), (-2, -3), (-3, -4), (0, 1), (2, -1)\}$$

Show that the equation of the regression line is as follows:

$$y_i = \frac{58}{62}(x_i - 1) + \epsilon_i$$

Note that the regression line for Example 7.5 cannot be obtained by switching the variables x_i and y_i in the above equation, as both the slope and the intercept would be incorrect.

Problem 7.3 Consider an $n \times d$ data matrix D in which the rows (i.e., different values of \vec{x}_i) sum to the zero vector. The dependent variables are contained in the vector \vec{y} . The i th feature vector \vec{x}_i is scaled up (multiplied) with the i th dependent variable y_i , and the d -dimensional mean of these vectors is stored in $\vec{\mu}_{xy}$. Show that the closed-form solution \vec{w} to linear regression is related to the $d \times d$ covariance matrix C of the d -dimensional features and mean vector $\vec{\mu}_{xy}$ using the following relationship:

$$\vec{w} = C^{-1}\vec{\mu}_{xy}^T$$

The covariance matrix is computed without the Bessel correction.

A hint for solving the above problem is to examine the interpretation of $D^T D$ and $D^T \vec{y}$ in the context of the above problem. The closed-form solution to linear regression contains these subexpressions.

7.4.2 The Case of One Non-Trivial Predictor Variable

It is helpful to examine the special case of linear regression with a single *non-trivial* predictor variable (i.e., number of variables not including the engineered feature of 1 for the bias). In such a case, the solution vector $\vec{w} = [w_1, w_2]^T$ is 2-dimensional, where w_1 is the coefficient of the single predictor variable, and w_2 is the bias that is always multiplied with the synthetic (engineered) variable containing a value of 1. In such a case, it can be shown that the coefficient w_1 is the ratio of the sample covariance between the non-trivial predictor X and outcome to the variance of the predictor variable:

$$\text{Slope of Regression Line} = w_1 = \frac{\hat{\sigma}_{XY}}{\hat{\sigma}_X^2} \quad (7.2)$$

The coefficient w_2 can then be found by using the fact that the solution to linear regression always passes through the sample mean of the predictor and outcome variables:

$$\text{Intercept of Regression Line} = w_2 = \hat{\mu}_Y - w_1 \hat{\mu}_X \quad (7.3)$$

It is noteworthy that the slope w_1 of the regression line contains the variance of only the predictor in the denominator. Therefore, the solution is not symmetric in terms of the predictor and outcome variables. This is because linear regression is posed in terms of asymmetrically minimizing the error on outcomes, rather than an objective function that minimizes the perpendicular distances of data points to the regression line.

Example 7.6 Consider the single non-trivial predictor problem of Example 7.5 in which the independent-dependent pairs are as follows:

$$\{(4, 5), (3, 4), (2, 3), (1, 2), (-1, 0), (-2, -1), (-3, -2), (-4, -3), (1, 0), (-1, 2)\}$$

Use the variance-covariance slope formula of this section to derive the optimal solution to linear regression.

Solution: It can be shown that the required covariance and variance are as follows:

$$\begin{aligned}\hat{\sigma}_{XY} &= \frac{58}{10} \\ \hat{\sigma}_X^2 &= \frac{62}{10}\end{aligned}$$

Here, X is the single non-trivial predictor. Therefore, the slope of the regression line is $58/62$. Using the fact that the regression line must pass through the mean $(\bar{x}, \bar{y}) = (0, 1)$ of the data, the optimal regression line can be shown to be the following:

$$y_i = \frac{58}{62}x_i + 1 + \epsilon_i$$

■

Example 7.7 Consider a linear regression problem with a single non-trivial predictor in which the means of the independent and dependent variables are 1 and 3, respectively. The covariance of the independent and dependent variable is 5 and the variance of the independent variable is 2.5. Find the equation of the best-fit regression line obtained by the least-squares approach.

Solution: The slope of the regression line is given by $w_1 = 5/2.5 = 2$. The intercept of the regression line is given by $w_2 = \hat{\mu}_Y - 2\hat{\mu}_X = 3 - 2(1) = 1$. Therefore, the equation of the regression line is as follows:

$$y_i = 2x_i + 1$$

■

Example 7.8 Consider a regression problem with a single non-trivial predictor variable (and bias). The variance of the predictor variable is 5 and the slope is -2 for the regression line. What is the covariance between the outcome and predictor variables? What is the regression line, if the means of the regressor and regressand are 3 and 4, respectively. Is there enough information to find the variance of the outcome (regressand) variable?

Solution: The covariance is given by the product of the slope w_1 and the predictor variance using the relationship $\hat{\sigma}_{XY} = w_1 \hat{\sigma}_X^2$. Therefore the covariance is equal to $(-2)(5)$, which is -10 .

One can use the point-slope form in order to infer that the regression line is given by $y_i - 4 = -2(x_i - 3)$, which is the same as $y_i = -2x_i + 10$. There is not enough information to find the variance of the regressand. ■

Example 7.9 Consider the linear regression problem with a single non-trivial predictor variable. Show that all residuals are zero if and only if the sample correlation between the predictor and outcome variable is either $+1$ or -1 .

Solution: Suppose that all residuals are 0 in the relationship $y_i = \frac{\hat{\sigma}_{XY}}{\hat{\sigma}_X^2} x_i + b + \epsilon_i$. Therefore, we have the following:

$$y_i = \frac{\hat{\sigma}_{XY}}{\hat{\sigma}_X^2} x_i + b \quad \forall i$$

Since the regression line always passes through the mean of the data, one obtains the following:

$$\hat{\mu}_Y = \frac{\hat{\sigma}_{XY}}{\hat{\sigma}_X^2} \hat{\mu}_X + b$$

By taking the difference of the above two equations, the following is obtained:

$$(y_i - \hat{\mu}_Y) = \frac{\hat{\sigma}_{XY}}{\hat{\sigma}_X^2} (x_i - \hat{\mu}_X) \quad \forall i \tag{7.4}$$

Multiplying each side with $y_i - \hat{\mu}_Y$, adding over all i , and dividing by n , the following is obtained:

$$\underbrace{\sum_{i=1}^n (y_i - \hat{\mu}_Y)^2}_{{\hat{\sigma}_Y^2}} = \frac{\hat{\sigma}_{XY}}{\hat{\sigma}_X^2} \underbrace{\sum_{i=1}^n (x_i - \hat{\mu}_X)(y_i - \hat{\mu}_Y)}_{{\hat{\sigma}_{XY}}}$$

Simplifying the above relationship, it is possible to show that $\hat{\rho}_{XY}^2 = \hat{\sigma}_{XY}^2 / (\hat{\sigma}_X^2 \hat{\sigma}_Y^2) = 1$. In other words $\hat{\rho}_{XY}$ is either $+1$ or -1 .

To show the converse, suppose that the sample correlation is 1 or -1 . Then, one can use a similar analysis to the above to derive the following generalized form of Equation 7.4 that includes ϵ_i :

$$(y_i - \hat{\mu}_Y) = \frac{\hat{\sigma}_{XY}}{\hat{\sigma}_X^2} (x_i - \hat{\mu}_X) + \epsilon_i \quad \forall i \tag{7.5}$$

Isolating ϵ_i on one side, squaring, adding over all i , and dividing by n , the following is obtained:

$$\frac{\sum_{i=1}^n \epsilon_i^2}{n} = \sum_{i=1}^n \frac{(y_i - \hat{\mu}_Y)^2}{n} - 2 \frac{\hat{\sigma}_{XY}}{\hat{\sigma}_X^2} \frac{\sum_{i=1}^n (x_i - \hat{\mu}_X)(y_i - \hat{\mu}_Y)}{n} + \frac{\hat{\sigma}_{XY}^2}{\hat{\sigma}_X^4} \sum_{i=1}^n \frac{(x_i - \hat{\mu}_X)^2}{n}$$

Simplifying the above equation, we obtain the following:

$$\frac{\sum_{i=1}^n \epsilon_i^2}{n} = \hat{\sigma}_Y^2 - \frac{\hat{\sigma}_{XY}^2}{\hat{\sigma}_X^2}$$

Then, one can replace $\hat{\sigma}_{XY}^2$ with $\hat{\sigma}_X^2 \hat{\sigma}_Y^2$ because the squared correlation is 1. The right-hand side simplifies to 0. In other words, the sum of the squares of the residuals is 0. Therefore, each residual must be 0 as well. ■

Problem 7.4 Consider the linear regression problem with a single non-trivial predictor variable (and bias). Show that the slope of the regression line is positive if and only if the sample covariance between the outcome and predictor variable is positive.

Problem 7.5 Consider the linear regression problem with a single non-trivial predictor variable (and bias). Show that the slope of the optimal regression line can be no larger in absolute magnitude than the ratio of the sample standard deviations of the outcome variable and the non-trivial predictor variable. Under what conditions are the two exactly equal?

Both of the above problems can be solved by examining the relationship between the slope, $\hat{\sigma}_{XY}$, and $\hat{\sigma}_X$. Furthermore, note that $\hat{\sigma}_{XY}$ is at most $\hat{\sigma}_X \hat{\sigma}_Y$.

7.4.3 Solution with Gradient Descent for Squared Loss

The solution to linear regression thus far has used a closed-form approach, where \vec{w} is expressed in terms of the $n \times d$ data matrix D and n -dimensional regressand vector \vec{y} . Least-squares regression is one of the few machine learning problems that has a closed-form solution. In most cases, it is necessary to solve the optimization problem computationally by using a gradient-descent approach. For example, even L_1 -loss regression does not have a closed-form solution. In this section, we will explore the more general approach of gradient descent for least-squares regression. We will first describe the solution with the use of matrix-based updates and then show how one can make the solution more efficient with stochastic gradient descent in which updates are made for smaller sets of points. We first restate the objective function of linear regression (from the previous section) along with its gradient with respect to the parameter vector \vec{w} :

$$\text{Minimize}_{\vec{w}} J = \|D\vec{w} - \vec{y}\|^2$$

We express the derivative of this objective function in matrix calculus notation in the denominator layout:

$$\frac{\partial J}{\partial \vec{w}} = 2 D^T (D\vec{w} - \vec{y})$$

The approach starts by initializing the vector \vec{w} , so that the entries have small random values. In gradient descent, the vector \vec{w} is modified in the negative direction of the gradient

with the use of a learning rate $\alpha > 0$. This is a standard approach used in unconstrained optimization. Therefore, we have the following gradient-descent update for linear regression:

$$\vec{w} \leftarrow \vec{w} - \alpha \frac{\partial J}{\partial \vec{w}}$$

On substituting the value of the gradient in the update equation and absorbing the factor of 2 within the learning rate, the following update is obtained:

$$\vec{w} \leftarrow \vec{w} - \alpha D^T \underbrace{(D\vec{w} - \vec{y})}_{\text{Error vector}}$$

Note that $D\vec{w}$ is an n -dimensional vector containing the predictions on the training instances, and therefore $(D\vec{w} - \vec{y})$ is an n -dimensional error vector containing the differences between the predicted and observed values. Therefore, gradient descent uses the vector of errors made by the algorithm in order to make updates. This is a reasonable strategy, as the algorithm needs to modify the coefficient vector in order to rectify the errors made by the algorithm. The updates are continually made until the algorithm converges to an optimal solution.

More insight can be obtained by decomposing the gradient in terms of the individual rows \vec{x}_i of D :

$$\frac{\partial J}{\partial \vec{w}} = 2 D^T (D\vec{w} - \vec{y}) = 2 \sum_{i=1}^n \vec{x}_i^T (\vec{w} \cdot \vec{x}_i^T - y_i)$$

Therefore, the gradient-descent updates may be written in the following way after absorbing the factor of 2 within the learning rate:

$$\vec{w} \leftarrow \vec{w} - \alpha \sum_{i=1}^n \vec{x}_i^T \underbrace{(\vec{w} \cdot \vec{x}_i^T - y_i)}_{\text{Error } i}$$

This particular form of the update is more insightful as it shows how the updates can be decomposed into a separable sum of the contributions of different training instances. This form of the update also shows that it can take a while to make a single update when there is a large number of training instances.

Example 7.10 Consider the single non-trivial predictor problem of Example 7.5 in which the independent-dependent pairs are as follows:

$$\{(4, 5), (3, 4), (2, 3), (1, 2), (-1, 0), (-2, -1), (-3, -2), (-4, -3), (1, 0), (-1, 2)\}$$

We know from the solution to Exercise 7.5 that the optimal solution is $w_1 = 58/62$, $w_2 = 1$. Here, w_2 is the bias coefficient of the engineered feature value of 1. Consider the situation, where the initial starting point is $[w_1, w_2] = [3, -1]$. Show that the gradient descent step moves $[w_1, w_2]$ closer to the optimal solution for appropriate choice of step size.

Solution: The pairs above are denoted by (x_i, y_i) . The data points $\vec{x}_i = [x_i, 1]$ along with the engineered variable value of 1 (corresponding to the bias coefficient) are as follows:

$$\{[4, 1], [3, 1], [2, 1], [1, 1], [-1, 1], [-2, 1], [-3, 1], [-4, 1], [1, 1], [-1, 1]\}$$

Gradient descent requires the computation of error on each data point by current solution $\vec{w} = [3, -1]^T$. On plugging in the above data pairs into the error equation $e_i = \vec{w} \cdot \vec{x}_i^T - y_i = (3x_i - 1 - y_i)$, one obtains the following list of errors:

$$6, 4, 2, 0, -4, -6, -8, -10, 2, -6$$

On computing $\sum_i e_i \vec{x}_i$ one obtains the following:

$$\sum_i e_i \vec{x}_i = [128, -20] \propto [32, -5]$$

Therefore, the gradient descent update will be as follows:

$$\vec{w} = [3, -1]^T - \alpha [32, -5]^T$$

Here, the step-size α is critical. For small step sizes both components of \vec{w} move in the correct direction since w_1 is decreased and w_2 is increased from the initial solution (in which the former is too large and the latter is too small). For example, choosing $\alpha = 0.05$ results in a new weight vector of $[1.4, -0.75]$, which is a step in the correct direction for both components of the vector \vec{w} . Choosing very large step sizes will cause the solution to overshoot. One can use a technique called *line search* [6] to set α optimally. ■

7.4.3.1 Stochastic Gradient Descent

The above form of the update leads to some natural questions as to whether one needs to use *all* training instances in order to make a robust update. A key point is that if one samples a subset S of a few hundred training instances randomly and computes the direction of the gradient using only these instances, a sampled direction is obtained which approximates the true direction quite well. If one views a gradient as a random variable that is computed in point-wise fashion, then computing the mean of this random variable over a hundred points drops the variance of the mean gradient direction by a factor of 100 as well. Furthermore, the initial values of \vec{w} are very far from their optimal values, which enables the calculation of a reasonable direction of update with only a modest number of points. This type of update is referred to as *mini-batch stochastic gradient descent*, and is expressed as follows:

$$\vec{w} \leftarrow \vec{w} - \alpha \sum_{i \in S} \vec{x}_i^T \underbrace{(\vec{w} \cdot \vec{x}_i^T - y_i)}_{\text{Error } i}$$

Note that the only difference from the gradient-descent update is that we are now sampling over a random subset S of training instances instead of using all the training instances. Using subsets of size of a few hundred instances can often lead to very robust updates. Surprisingly, even though individual updates become more noisy, the overall algorithm still converges to a high-quality solution. This continues to be the case when the size of the set S is chosen to be 1. This case is referred to as pure stochastic gradient descent, and it is especially recommended when the number of training instances is not significantly larger than the number of predictor variables. The reason for the robustness of stochastic gradient descent in such cases is that such updates lead to a final solution that tends to vary less with the specific choice of training sample. In other words, the resulting solution has lower

variance in terms of prediction on out-of-sample data and often provides more accurate prediction on unseen test instances (without overfitting). This is a surprising result at first glance, considering the high level of randomness in an individual stochastic gradient descent update. The key point to understand is that stochastic gradient descent will often provide poorer accuracy on in-sample instances but better accuracy on out-of-sample instances. Since the utility of machine learning applications is primarily governed by its accuracy on out-of-sample data, it follows that stochastic gradient descent is an effective approach in practical settings.

7.4.4 Gradient Descent For L_1 -Loss Regression

As discussed earlier in this chapter (cf. page 311), L_1 -loss regression can be formulated as follows:

$$\text{Minimize}_{\vec{w}} J = \sum_{i=1}^d \|\vec{w} \cdot \vec{x}_i^T - y_i\|_1$$

Then, the derivative of the objective function with respect to \vec{w} (in matrix calculus notation) is³ as follows:

$$\frac{\partial J}{\partial \vec{w}} = \sum_{i=1}^n \vec{x}_i^T \underbrace{\text{sign}(\vec{w} \cdot \vec{x}_i^T - y_i)}_{\text{Error sign}}$$

Therefore, the gradient-descent updates may be written in the following way:

$$\vec{w} \leftarrow \vec{w} - \alpha \sum_{i=1}^n \vec{x}_i^T \underbrace{\text{sign}(\vec{w} \cdot \vec{x}_i^T - y_i)}_{\text{Error sign}}$$

Here, $\alpha > 0$ is the learning rate. The gradient-descent steps may be iterated to convergence. As in the case of least-squares regression, one can also use stochastic gradient-descent steps by sampling training instances for updates.

7.5 Handling Categorical Predictors

The solution methods discussed thus far assume that the predictors are numerical. How does one perform regression when the predictors are categorical? Categorical predictors present a significant challenge because there is no ordering among the values of the features, and the regression model is inherently designed to deal with numerical predictors. One observation is that the special case of binary categories can be viewed in two different ways. A binary predictor variable is both categorical and numerical. This is because one can impose an arbitrary ordering between binary categories (e.g., biological gender) by setting one of the categorical values to 0 and the other to 1.

The multiway case is slightly more difficult. For example, consider the case where there are five races corresponding to Caucasian, African American, Hispanic, Native American, and Asian. There is no natural ordering among the races like numeric variables. Certain types of attributes, such as the ZIP code, have a large number of categorical values. For

³Although the derivative does not exist at points where $\vec{w} \cdot \vec{x}_i^T - y_i = 0$ because of the difference in sign between the left derivative and right derivative, this fact turns out to be numerically irrelevant as $\vec{w} \cdot \vec{x}_i^T - y_i = 0$ will rarely be exactly zero during updates. In cases where it is exactly zero, the derivative can also be set to zero using the *subgradient* principle [6].

notational purposes, we assume that the number of categorical values is m for the categorical predictor that is being considered. There are two common solutions that are used to address this setting. The preferred choice of setting depends on the particular community of practitioners or researchers to which an analyst belongs:

1. *One-hot encoding*: In one-hot encoding, each of the m values of the category is treated as a binary attribute. Only one of the binary attributes is “hot” by taking on the value of 1, which corresponds to the value of the categorical variable. It is noteworthy that there is an inherent redundancy among the values of the binary attributes, as knowing any $(m - 1)$ of the binary attributes automatically reveals the remaining binary attribute. This approach is very popular in the machine learning community of practitioners and researchers.
2. *The reference and dummy variable approach*: In this approach one of the categorical attribute values is treated as special, and is referred to as the *reference value* of the categorical attribute. The remaining $(m - 1)$ categorical values have a binary predictor referred to as a *dummy predictor*. When all dummy predictors take on the value of 0, it refers to the fact that categorical variable takes on the reference value. Otherwise, exactly one of the dummy predictors takes on the value of 1, which is interpreted in a manner similar to one-hot encoding. This type of approach is popular in the statistics community.

The dummy-variable approach results in less redundancy among attributes as compared to the one-hot encoding approach. The interpretation of the different coefficients is also different. In the one-hot encoding approach, the difference between the regression coefficients of two binary variables derived from the same category is equal to the average change in outcome caused by switching between those categories (while keeping other predictors constant). Therefore, regression coefficients can only be interpreted in pairwise fashion. In the dummy variable approach, the coefficient of that categorical value is equal to the change in outcome caused by switching from the reference categorical value to the (non-reference) value denoted by that dummy predictor (while keeping all other variables constant). Note that setting all dummy predictors to 0 always corresponds to using the reference value for all predictors.

Example 7.11 Suppose that you use the dummy variable representation for the categorical predictors of race and gender. The possible values for race are Caucasian, African American, Asian, Native American, and Hispanic. The possible values for gender are Male and Female. The “Caucasian” value is assumed to be the reference for race, and “Male” is assumed to be reference for gender. You are regressing demographic data against annual income in the United States. You find that the coefficient of the “Hispanic” predictor is $-12,232$ and the coefficient of the “Female” predictor is -9123 . Give an intuitive interpretation of these coefficient values.

Solution: The coefficient of the Hispanic predictor means that Hispanic people make $12,232$ less than Caucasian people on average when all other attributes are held constant. The coefficient of the “Female” predictor means that females make 9123 less than males on average when all other attributes are held constant. ■

7.6 Overfitting and Regularization

The primary application of the regression model occurs in cases where the prediction is used over new observations that are not present in the training data. A problem arises in making robust predictions for out-of-sample test instances, when few observations are available to train the model. In such cases, the predictions on out-of-sample data are often highly inaccurate. This weakness is especially notable when the number of variables is greater than the number of observations. In such cases, the optimal solution to the regression problem is not unique. Recall that the closed-form solution to the regression problem is as follows:

$$\vec{w} = (D^T D)^{-1} D^T \vec{y}$$

Here, the rows of the data matrix D contain the observations with predictors and the vector \vec{y} contain the responses. The main assumption in being able to compute the above solution is that the matrix $D^T D$ is invertible. Otherwise, the solution for \vec{w} might be non-unique. For example, when one uses gradient descent to find the optimal solution to \vec{w} , drastically different values of the vector \vec{w} might be obtained depending on the initialization point of gradient descent. Most of these solutions have overfit to the nuances of the training data and they will not be useful for making predictions on new observations. The different solutions for \vec{w} might give different predictions for new test instances, which makes predictions on new test instances unreliable. The accuracy of prediction on the training data will be very high (i.e., $\epsilon_i \approx 0$ for most i), but the accuracy of prediction on out-of-sample test instances will be poor. Furthermore, changing the training data will lead to very different (inaccurate) predictions on the same out-of-sample test instances. In other words, the errors are caused by high *variance* of the model (cf. section 6.7 of Chapter 6).

In order to understand this point, consider the situation where we have data about adult males, containing their height, hair length, beard length, and right-thumb nail length. We are trying to predict the arm-span of the individual from these features. It is well known that the arm span can be predicted very accurately from the height, but the other variables are unrelated to the arm span. Therefore, such variables should (ideally) have no impact on the prediction. However, if we have only a small amount of data, it is easy to build spurious models that have nothing to do with reality. For example, consider the case where one only has two data instances for training, which are as follows:

Height x_1 (m)	Hair Length x_2 (in)	Beard Length x_3 (in)	Nail length x_4 (mm)	Arm Span y (m)
1.57	0.3	0.5	1.0	1.6
1.83	0.6	0.3	1.4	1.8

The prediction equation here is as follows:

$$y \approx w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + b$$

In this case, it is evident that the correct regression relationship is $y = x_1$ (based on domain knowledge), and this relationship is largely consistent with the data, although there are small errors in the prediction (as one would normally expect). One can view the problem of finding the coefficients of linear regression in the above problem as that of solving the equations below as accurately as possible:

$$1.6 \approx 1.57w_1 + 0.3w_2 + 0.5w_3 + 1.0w_4 + b$$

$$1.8 \approx 1.83w_1 + 0.6w_2 + 0.3w_3 + 1.4w_4 + b$$

The two equations above are derived using the two data points in the table above. Since this is a set of two equations in five unknowns, one will usually be able to find an *infinite* number of possible solutions to this system of equations. For example, the prediction $y = 2x_2 + 2x_3$ has zero error on the training data. Similarly, the prediction $y = 2x_4 - 2x_1 + 0.2$ also has zero error on the training data. These are spurious relationships because the arm span has nothing to do with measurements based on grooming habits of individuals. Any linear combination of the two aforementioned predictive equations will also provide zero error. In most cases, having fewer training instances than the number of variables will result in an infinite number of possible solutions, even if the training data is generated completely randomly. Such coefficients will not generalize very well to new test instances where the predictors are known but the response variable is not known. An important point is that choosing a different set of two training instances will yield a completely different set of coefficients that yield perfect predictions on the training data but very poor predictions on new samples of predictors. This is the classical problem of overfitting, which is caused by too few data points compared to the complexity of the formulation (reflected by the number of variables). The wide variation in predictor coefficients across different data samples is reflective of high variance in coefficient estimation. Here, an important point is that the sample standard deviation of the coefficient estimates reduces⁴ with increasing number of observations. Clearly, methods are needed to add *bias* to the problem, so that one can reach a point of optimal model complexity (cf. Figure 6.8 of Chapter 6).

It is noteworthy that the problem of increased number of variables compared to the number of points can be partially solved by forcing a subset of the coefficients to be zero. By doing so, one is effectively reducing the number of variables used in the system of equations, although one is allowed to use the subset that provides the most accurate prediction on the training data. In other words, one might formulate the regression problem as follows:

$$\text{Minimize}_{\vec{w}} J = \frac{1}{2} \sum_{i=1}^n (\vec{w} \cdot \vec{x}_i - y_i)^2$$

subject to:

Atmost k dimensions of \vec{w} are nonzero.

The factor of $1/2$ has been added to the objective function in order to avoid the pesky factor of 2 in various derivatives during differentiation. If this approach is used with $k = 1$ on the previous example, one will be able to obtain the correct solution in which only the value of w_1 is set to 1 . This improvement in performance by adding constraints is not a coincidence that is specific to this example. When the number of variables is too large compared to the number of observations, one is able to combine them in complicated ways in order to force correct predictions (on the training data) in ways that are not truly representative of the underlying data distribution. Such models are not generalizable to new observations; by adding constraints to the maximum number of non-zero coefficients, such overly complicated and unrepresentative solutions are dropped. Note that this is a way of adding *bias*, because it reflects a natural “prior belief” that one is more likely to find representative trends in the coefficients if fewer of them have nonzero values. In other words, we have the *bias* that “simpler” solutions are likely to be more representative of new observations. Such a bias becomes particularly important when few observations are available and the loss function of linear regression tries to find overly complicated solutions to create perfectly fitting solutions.

⁴A detailed discussion is beyond the scope of this book.

However, adding constraints to the number of non-zero coefficients makes the problem overly difficult to solve from a computational point of view. A simpler approach is to add penalties on the aggregate squared magnitude of the coefficients. Such an approach is referred to as penalty-based *regularization*. The resulting optimization problem is differentiable and easy to solve:

$$\text{Minimize}_{\vec{w}} J = \frac{1}{2} \sum_{i=1}^n (\vec{w} \cdot \vec{x}_i^T - y_i)^2 + \frac{\lambda}{2} \|\vec{w}\|^2$$

Here, $\lambda > 0$ is the regularization parameter reflecting the level of “simplicity” (bias) we want in our model. A larger value of λ increases the value of the bias, and is reflective of lower model complexity — at the same time, increasing the bias reduces the effect of variance on the error on out-of-sample observations (cf. Figure 6.8 of Chapter 6).

One can also formulate this problem in matrix form using the $n \times d$ matrix D containing the vectors $\vec{x}_1 \dots \vec{x}_n$ in its rows and the n -dimensional column vectors $\vec{y} = [y_1 \dots y_n]^T$:

$$\text{Minimize}_{\vec{w}} J = \frac{1}{2} \sum_{i=1}^n \|D\vec{w} - \vec{y}\|^2 + \frac{\lambda}{2} \|\vec{w}\|^2$$

This form of regularization is referred to as L_2 -regularization, since it adds an L_2 -norm penalty on the coefficients. It is also referred to as *Tikhonov Regularization*.

Example 7.12 Consider a data set containing the three independent-dependent pairs (x_i, y_i) as follows:

$$\{(x_i, y_i) : i \in \{1, 2, 3\}\} = \{(1, 1), (2, 2), (3, 3)\}$$

Suggest a good regression line by inspection, which is of the form $y_i = w \cdot x_i + b$. Here, w is a regression coefficient and b is the bias. Now suppose that someone added three random integers to each x_i to create a new 4-dimensional representation \vec{x}'_i . The resulting 5-dimensional independent-dependent pairs are as follows:

$$\{(\vec{x}'_i, y_i) : i \in \{1, 2, 3\}\} = \{(1, 2, 0, 1, 1), (2, 1, 1, 2, 2), (3, 0, 0, 1, 3)\}$$

Show that it is possible to create perfect predictions on the training data using only the random integers as regressors. What is the drawback of using such a model? Compare the L_2 -norm of the new coefficient vector with one that uses only the original predictor. Discuss why regularization will help.

Solution: A good regression line using inspection is $y_i = x_i$, which fits the training data well. It uses a single nonzero coefficient and fits all three points exactly. Consider the matrix R defined by the three random integers:

$$R = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

The dependent variable vector is $\vec{y} = [1, 2, 3]^T$. Because the matrix R is square and invertible, the solution to regression can be simplified as follows:

$$\vec{w} = (R^T R)^{-1} R^T \vec{y} = R^{-1} (R^T)^{-1} R^T \vec{y} = R^{-1} \vec{y}$$

On computing R^{-1} using a web calculator, the coefficients of the three added variables (containing) are as follows:

$$\vec{w} = \frac{1}{2} \begin{bmatrix} 1 & 0 & -1 \\ -1 & 2 & -3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -2 \\ -6 \\ 6 \end{bmatrix} = \begin{bmatrix} -1 \\ -3 \\ 3 \end{bmatrix}$$

In other words, one can now predict the regressand exactly using the following regression on the 4-dimensional representation:

$$y = 0x_1 - x_2 - 3x_3 + 3x_4$$

The most relevant regressor x_1 is ignored and yet perfect prediction on the training data is obtained with random regressor values. This is a case of overfitting, and the resulting regressor variables are likely to provide poor accuracy on the test data. In other words, good performance on the training data does not generalize to the test data. The L_2 -norm of the new set of coefficients is $\sqrt{19}$, which is much greater than the original L_2 -norm of 1. Imposing a penalty on the coefficients will help find the original result, since it also has zero prediction error. ■

7.6.1 Closed-Form Solution for Regularized Formulation

On differentiating the above matrix-based objective function J with respect to \vec{w} and setting it to $\vec{0}$, the following condition is obtained using matrix calculus notation in denominator layout:

$$\frac{\partial J}{\partial \vec{w}} = D^T(D\vec{w} - \vec{y}) + \lambda \vec{w} = \vec{0}$$

Rearranging the optimality condition, one obtains the following:

$$(D^T D + \lambda I)\vec{w} = D^T \vec{y}$$

A key point is that $D^T D + \lambda I$ is invertible for $\lambda > 0$. This is because $(D^T D + \lambda I)$ is known to be positive definite for $\lambda > 0$, and such a matrix is always invertible. Therefore, the solution for \vec{w} is as follows:

$$\vec{w} = (D^T D + \lambda I)^{-1} D^T \vec{y} \quad (7.6)$$

The regularization parameter can be chosen by not using some of the observations in the matrix D and using them to evaluate the sum of the squares of the residuals instead. The value of λ is chosen to optimize performance on this out-of-sample data by trying different values of λ over a range and selecting the one that provides best performance.

Example 7.13 Give an example of the regularized version of the problem to show that the residuals $\epsilon_i = y_i - \vec{w} \cdot \vec{x}_i^T$ may no longer sum to 0 over the n points for the optimal solution (as is the case in the unregularized version of the problem).

Solution: If we choose very large values of the regularization parameter λ , the optimal vector \vec{w} will have very small magnitudes of entries. In such a case, each prediction on the training data will be almost 0, and therefore each residual ϵ_i will be almost equal to y_i . Selecting any training data set in which the outcome variables do not sum to 0 will result in an example in which the residuals do not sum to 0 either. ■

Problem 7.6 Since residuals do not sum to 0 in the regularized version of the problem, it is evident that the proof of Lemma 7.3 cannot be generalized to the regularized setting. Which step(s) in the proof of Lemma 7.3 will no longer be valid on adding regularization?

7.6.2 Solution Based on Gradient Descent

In this section, we will discuss an approach that solves the problem of regularized linear regression with gradient descent. Consider the matrix-based objective function from the previous section:

$$\text{Minimize}_{\vec{w}} = \frac{1}{2} \|D\vec{w} - \vec{y}\|^2 + \frac{\lambda}{2} \|\vec{w}\|^2$$

We express the derivative of this objective function using matrix calculus notation in the denominator layout:

$$\frac{\partial J}{\partial \vec{w}} = D^T(D\vec{w} - \vec{y}) + \lambda \vec{w}$$

The gradient-descent approach iteratively updates the weight vector in the negative direction of the gradient, starting from an initial set of (possibly random) components in the vector \vec{w} . This update in the negative direction of the gradient uses a learning rate $\alpha > 0$, which controls the rate at which updates are made. Therefore, we have the following:

$$\vec{w} \leftarrow \vec{w} - \alpha \frac{\partial J}{\partial \vec{w}}$$

On substituting the value of the gradient in the update equation, the following update is obtained:

$$\vec{w} \leftarrow \vec{w}(1 - \alpha\lambda) - \alpha D^T \underbrace{(D\vec{w} - \vec{y})}_{\text{Error vector}}$$

These updates are repeated to convergence. The main difference in the update equation with respect to the unregularized version of the problem is that the weight vectors are being shrunk by a factor of $(1 - \alpha\lambda)$ in each step along with the (original) data-driven updates. This shrinkage is designed to bias the solution towards weight vectors of smaller magnitude to discourage too many coefficients from having large absolute values.

One can also compute the gradient in terms of the individual rows \vec{x}_i of D :

$$\begin{aligned} \frac{\partial J}{\partial \vec{w}} &= D^T(D\vec{w} - \vec{y}) + \lambda \vec{w} \\ &= \sum_{i=1}^n \vec{x}_i^T (\vec{w} \cdot \vec{x}_i^T - y_i) + \lambda \vec{w} \end{aligned}$$

Therefore, the gradient-descent updates may be written in the following form:

$$\vec{w} \leftarrow \vec{w}(1 - \alpha\lambda) - \alpha \sum_{i=1}^n \vec{x}_i^T \underbrace{(\vec{w} \cdot \vec{x}_i^T - y_i)}_{\text{Error } i}$$

As in the case of the unregularized version of the problem, one can also perform stochastic gradient descent by performing updates on random samples of observations.

Problem 7.7 Are the following statements true or false? (a) The test-data predictions of unregularized linear regression do not change if the training/test data are both changed by multiplying one of the regressor variables with 2 without changing the regressand values in the training data, (b) Does your answer to (a) change if the regression is regularized?

7.6.3 LASSO Regularization

The acronym LASSO stands for *Least Absolute Shrinkage and Selection Operator*. The regularization approach discussed thus far uses the sum of the squares of the residuals to shrink the magnitudes of the parameters. In other words, the L_2 -penalty is used on the weight vector. However, it is also possible to use the L_1 -penalty on the weights, just as one can use the L_1 -penalty for the loss function. The resulting optimization formulation is as follows:

$$\text{Minimize}_{\vec{w}} J = \frac{1}{2} \sum_{i=1}^n (\vec{w} \cdot \vec{x}_i^T - y_i)^2 + \lambda \|\vec{w}\|_1$$

One can also write the above objective function as follows:

$$\text{Minimize}_{\vec{w}} J = \frac{1}{2} \sum_{i=1}^n (\vec{w} \cdot \vec{x}_i^T - y_i)^2 + \lambda \sum_{j=1}^d |w_j|$$

This optimization formulation does not have a closed-form solution. However, it is possible to solve the problem with gradient descent. The objective function does have a gradient at all points other than at points where any particular value of w_j is 0. This is because the left derivative of $|w_j|$ with respect to w_j is -1 and the right derivative of $|w_j|$ with respect to w_j is $+1$. At such points, one may fix⁵ the value of the derivative of the j th dimension of the regularization to 0. The resulting gradient-descent update is as follows:

$$\vec{w} \leftarrow \vec{w} - \alpha \lambda \cdot \text{sign}(\vec{w}) - \alpha \sum_{i=1}^n \vec{x}_i^T \underbrace{(\vec{w} \cdot \vec{x}_i^T - y_i)}_{\text{Error } i}$$

Here, the sign function, when applied to the d -dimensional vector \vec{w} , returns a d -dimensional vector with values of either $+1$ or -1 , depending on the signs of the corresponding components of \vec{w} .

LASSO regression finds coefficients that are qualitatively different from those obtained using L_2 -regularization. While L_2 -regularization mostly favors nonzero coefficients of small magnitude in the optimal solution, LASSO regularization tends to set many coefficients to values of 0. In other words, such variables are *completely dropped* from the final prediction equation and therefore do not factor into the model at all! Therefore, LASSO may be viewed as a form of feature selection. Therefore, LASSO regularization is often used for feature selection in high-dimensional data, when it is desired to select a small subset of relevant features. However, the quality of the optimal solution is generally (slightly) better with L_2 -regularization.

7.7 A Probabilistic View of Regularization

As discussed in the previous section, regularization can be viewed as an indirect way of making the prior assumption that the regression coefficients have small magnitude. The regularization assumption can be fit into the probabilistic framework for regression by using MAP estimation — the assumption has the probabilistic interpretation that *a prior probability distribution exists on the parameter vector \vec{w} that favors smaller magnitudes of these coefficients*. In other words, the probability density should be high for small values of

⁵This assumption is based on the principle of *subgradients*.

$\|\vec{w}\|$. This approach is based on the principle of maximum a posteriori estimation (MAP), as discussed in section 6.6.2 of Chapter 6. MAP estimation is useful when less data is available, and therefore biasing the search for \vec{w} reduces the variance of estimation. The unregularized version of regression can be considered a special case of MAP estimation in which the prior distribution on \vec{w} is assumed to be the uniform distribution⁶ on $[-\infty, \infty]^d$ (and therefore there is no bias towards selecting smaller values of $\|\vec{w}\|$).

As in the case of unregularized regression, it is assumed that the error \mathcal{E} is distributed according to the Gaussian $\mathcal{N}(0, \sigma^2)$, whereas the prior on the d -dimensional parameter vector \vec{W} is a spherical d -dimensional Gaussian distribution with zero mean and variance σ_0^2 in all directions:

$$f_{\vec{W}}(\vec{w}) = \left(\frac{1}{2\pi\sigma_0^2} \right)^{d/2} \exp\left(-\frac{\|\vec{w}\|^2}{2\sigma_0^2}\right)$$

Small values of σ_0 result in greater regularization. Here, σ_0 represents a prior belief about the *standard deviation of \vec{w}* that needs to be selected by the analyst a priori. It is common to express the parameter σ_0 in relative terms with respect to the *data-driven Gaussian standard-deviation of errors σ* . A *regularization parameter λ* is introduced, and the prior variance σ_0^2 is assumed to be equal to σ^2/λ . Therefore, the distribution of \vec{W} is as follows:

$$f_{\vec{W}}(\vec{w}) = \left(\frac{\lambda}{2\pi\sigma^2} \right)^{d/2} \exp\left(-\frac{\lambda\|\vec{w}\|^2}{2\sigma^2}\right)$$

The zero mean of the Gaussian distribution encourages the magnitudes of the parameters in \vec{w} to be small. Large values of λ restrict this distribution to have low variance (i.e., to be tightly distributed around 0), which will reduce the magnitudes of the parameters in \vec{w} . Let \mathcal{E}_i be the Gaussian random variable representing the i th error and ϵ_i be the realized value of this variable. The likelihood function is identical to that in the unregularized version of this model:

$$f_{\mathcal{E}_i|\vec{W}=\vec{w}}(\epsilon_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\epsilon_i^2}{2\sigma^2}\right)$$

MAP estimation maximizes the posterior density $f_{\vec{W}|\mathcal{E}_1=\epsilon_1 \dots \mathcal{E}_n=\epsilon_n}(\vec{w})$. One can derive an expression for the posterior probability density using the Bayes rule:

$$f_{\vec{W}|\mathcal{E}_1=\epsilon_1 \dots \mathcal{E}_n=\epsilon_n}(\vec{w}) = \frac{f_{\vec{W}}(\vec{w}) \prod_{i=1}^n f_{\mathcal{E}_i|\vec{W}=\vec{w}}(\epsilon_i)}{f_{\mathcal{E}_1 \dots \mathcal{E}_n}(\epsilon_1, \dots, \epsilon_n)}$$

The Bayes rule can be expressed in proportionality form to simplify the above expression:

$$f_{\vec{W}|\mathcal{E}_1=\epsilon_1 \dots \mathcal{E}_n=\epsilon_n}(\vec{w}) \propto \underbrace{f_{\vec{W}}(\vec{w})}_{\text{Prior}} \underbrace{\prod_{i=1}^n f_{\mathcal{E}_i|\vec{W}=\vec{w}}(\epsilon_i)}_{\text{Likelihood}} \quad (7.7)$$

By taking the negative logarithm of the aforementioned expression, one can obtain the posterior (negative) log-likelihood function:

$$\mathcal{LP}(\epsilon_1, \dots, \epsilon_n, \vec{w}) = C - \ln(f_{\vec{W}}(\vec{w})) - \sum_{i=1}^n \ln(f_{\mathcal{E}_i|\vec{W}=\vec{w}}(\epsilon_i))$$

⁶This type of prior is referred to as an improper prior or non-informative prior. Furthermore, since a uniform distribution needs to be defined over a finite interval, one can define this distribution over $[-a, a]^d$ for arbitrarily large a to achieve the same result.

Here, C is a constant that accounts for the proportionality factor in the Bayes expression of Equation 7.7. The use of the logarithm changes the multiplicative proportionality factor to an additive constant C . On substituting the Gaussian density functions of the prior and likelihood in the above equation, one obtains the following:

$$\begin{aligned}\mathcal{LP}(\epsilon_1, \dots, \epsilon_n, \vec{w}) &= C + \frac{n+d}{2} \cdot \ln(2\pi) + (n+d) \cdot \ln(\sigma) - \frac{d}{2} \ln(\lambda) + \frac{\lambda}{2\sigma^2} \|\vec{w}\|^2 + \frac{1}{2\sigma^2} \sum_{i=1}^n \epsilon_i^2 \\ &= C'(\sigma, \lambda) + \frac{1}{\sigma^2} \left(\frac{\lambda}{2} \|\vec{w}\|^2 + \frac{1}{2} \sum_{i=1}^n \epsilon_i^2 \right)\end{aligned}$$

Here, $C'(\sigma, \lambda)$ absorbs terms that are constant or dependent on σ or λ but independent of the weight vector being optimized. Irrespective of the value of σ , the expression $\frac{\lambda}{2} \|\vec{w}\|^2 + \frac{1}{2} \sum_{i=1}^n \epsilon_i^2$ will need to be minimized in order to minimize the negative log-posterior. Therefore, we obtain the following loss function for minimization:

$$J = \frac{1}{2} \sum_{i=1}^n \epsilon_i^2 + \frac{\lambda}{2} \|\vec{w}\|^2 = \frac{1}{2} \sum_{i=1}^n (y_i - \vec{w} \cdot \vec{x}_i^T)^2 + \frac{\lambda}{2} \|\vec{w}\|^2$$

This objective function is exactly the same loss function that we introduced earlier for L_2 -regularized least-squares regression. As we will see in subsequent chapters, L_2 -regularization is used frequently in many machine learning models like classification and matrix factorization. *Almost all applications of L_2 -regularization in machine learning can be shown to be MAP estimations with Gaussian priors on the parameters.* It is noteworthy that L_1 -regularization has a similar interpretation, except that a Laplace prior is assumed on the weight vector.

Example 7.14 Consider the use of the Laplace prior distribution with scale factor s :

$$f_{\vec{W}}(\vec{w}) = \left(\frac{1}{2s} \right)^d \exp \left(-\frac{\|\vec{w}\|_1}{s} \right)$$

Show that using the Laplace prior on the weight parameters in MAP estimation results in L_1 -regularization in which the penalty is proportional to $\|\vec{w}\|_1$.

Solution: As in the case of L_2 regularization, one can use the Bayes rule to derive the posterior distribution of the weight vector:

$$f_{\vec{W}|\mathcal{E}_1=\epsilon_1, \dots, \mathcal{E}_n=\epsilon_n}(\vec{w}) \propto f_{\vec{W}}(\vec{w}) \prod_{i=1}^n f_{\mathcal{E}_i|\vec{W}=\vec{w}}(\epsilon_i)$$

Correspondingly, the negative logarithm of the posterior can be shown to be the following identical form to L_2 -regularization shown in the preceding section:

$$\mathcal{LP}(\epsilon_1, \dots, \epsilon_n, \vec{w}) = C - \ln(f_{\vec{W}}(\vec{w})) - \sum_{i=1}^n \ln(f_{\mathcal{E}_i|\vec{W}=\vec{w}}(\epsilon_i))$$

The only term that is different from L_2 -regularization is the negative logarithm of the prior. On substituting the L_2 -loss with variance of σ^2 and the L_1 -prior, one obtains

the following:

$$\mathcal{LP}(\epsilon_1, \dots, \epsilon_n, \vec{w}) = C'(\sigma) + \left(\frac{1}{s} \|\vec{w}\|_1 + \frac{1}{2\sigma^2} \sum_{i=1}^n \epsilon_i^2 \right)$$

It is clear from the above expression that the squared penalty in L_2 -regularization has been replaced with a penalty proportional to $\|\vec{w}\|_1$ in L_1 -regularization. In fact, by using $s = \sigma^2/\lambda$, one can obtain a regularized objective function that is similar to the form used in L_2 -regularization (in which the hyper-parameter is λ). ■

An important observation is that the negative logarithm of the prior is added to the objective function in MAP estimation (in comparison with MLE estimation) while ignoring constant additive terms. This observation is consistent with the Equation 6.20 of the previous chapter.

Example 7.15 *In elastic-net regularization, both an L_1 -penalty and an L_2 -penalty are added to the loss function. Specifically, the penalty term is $\lambda_1 \|\vec{w}\|_1 + \lambda_2 \|\vec{w}\|^2/2$. What prior do you think is used on the weight parameters?*

Solution: As noted above, the regularization term is the negative logarithm of the prior while ignoring constant terms (cf. Equation 6.20). In other words, the prior is proportional to the exponentiation of the negation of the regularization term. This amounts to using a prior proportional to the following product:

$$f_{\vec{W}}(\vec{w}) \propto \exp(-\lambda_1 \|\vec{w}\|_1) \exp(-\lambda_2 \|\vec{w}\|^2/2)$$

This quantity can be obtained as a product of the Laplace prior and the Gaussian prior. The scale factor of the Laplace prior is proportional to $1/\lambda_1$, whereas the variance of the Gaussian prior is proportional to $1/\lambda_2$. ■

7.8 Evaluating Linear Regression

Linear regression uses a number of statistics to evaluate the performance of the underlying algorithms. Examples of such statistics include the Root-Mean-Squared-Error (RMSE) and the R^2 -statistic. The RMSE is an absolute measure of prediction error, whereas the R^2 -statistic is a relative measure.

7.8.1 Evaluating In-Sample Properties of Regression

In this section, we will present the key statistics that are used to evaluate linear regression. The discussion in this section is based on training data performance. We separate out the training data performance from out-of-sample test data performance because they have distinctive properties in terms of evaluation. The performance metrics on the training data show a number of important properties, especially when regularization is not imposed on the optimization model. Therefore, many of the properties presented in this section assume unregularized linear regression unless otherwise mentioned. Nevertheless, even though these

properties may not hold in the presence of regularization, many metrics such as RMSE are used for evaluation irrespective of whether the model is regularized. Similarly, the metrics presented in this section apply both to in-sample data and out-of-sample data irrespective of the distinctive nature of the underlying properties.

The most basic evaluation metric of regression is the *Root-Mean-Squared Error*, also known as the *RMSE*. The RMSE is defined as follows on the training errors $\epsilon_1 \dots \epsilon_n$, where $\epsilon_i = y_i - \vec{w} \cdot \vec{x}_i^T$:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n \epsilon_i^2}{n - k - 1}} \quad (7.8)$$

Here, k is the number of *non-trivial* predictors in the regression problem (i.e., number of predictors not counting the bias coefficient that is included in \vec{w} with a feature engineering trick). Since we have consistently used d as the dimensionality of \vec{w} (including the engineered predictor), we have $k = d - 1$. Therefore, one can also write the RMSE using d instead of k as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n \epsilon_i^2}{n - d}} \quad (7.9)$$

The *mean-squared-error* (MSE) is defined in a similar manner to the RMSE, except that the final step of taking the square-root is omitted. The relationship between RMSE and MSE is similar to that between standard-deviation and variance. In fact, at $k = 0$ (or $d = 1$), when only the bias coefficient is used for regression, the RMSE is equal to the standard deviation of the outcome and the MSE is equal to the variance.

The squared RMSE is an unbiased estimate of the variance of the error. Furthermore, since the error is assumed to have zero mean at the *population level* and it follows a Gaussian probability distribution, the RMSE can be used to characterize the probability density of the outcome at any particular value of the predictor variables. Since the error has zero mean, it means that the outcome has expected mean that is equal to its predicted value and a variance equal to the squared RMSE.

Another useful measure of regression performance is the R^2 -statistic. The R^2 -statistic is a value between 0 and 1 that explains the fraction of the total variability in the outcome that can be explained with linear regression. The R^2 -value is often used as a surrogate for how linear the data may be considered. The total variability in the outcome, also referred to as the *total sum-of-squares* (TSS), is given by the following:

$$TSS = \sum_{i=1}^n (y_i - \hat{\mu}_Y)^2$$

In the above equation, the quantity $\hat{\mu}_Y = \sum_{i=1}^n y_i / n$ denotes the mean prediction on the training data. Note that TSS is simply an unscaled version of the sample variance on the outcome. In other words, dividing TSS with $(n - 1)$ yields the variance.

The TSS measure is rather basic and does not incorporate the effects of regression. The portion of this variability that *cannot* be explained by the regression is the *residual sum-of-squares* (RSS):

$$RSS = \sum_{i=1}^n \epsilon_i^2$$

Note that RSS is imply an unscaled version of the MSE and the relationship between the $RMSE$ and RSS is as follows:

$$RMSE = \sqrt{\frac{RSS}{n - k - 1}} = \sqrt{\frac{RSS}{n - d}}$$

The fraction of variance in the outcome that cannot be explained by regression is RSS/TSS . This insight is used to create another measure of the quality of regression, which is referred to as the R^2 -statistic:

$$R^2 = 1 - \frac{RSS}{TSS}$$

The value of R^2 on the training data is always lies between 0 and 1. This is because $0 \leq RSS \leq TSS$, which means that the ratio of RSS and TSS lies between 0 and 1. One advantage of R^2 over RMSE is that it is a normalized quantity that is comparable across models. It tells us how well a model fits the linearity assumption. The R^2 -statistic is often useful in distinguishing between alternative hypotheses about which quantities are related with a linear model. We provide a practical example from chemistry to show how the R^2 -statistic is used in experimental science:

Example 7.16 (Chemical Kinetics) *Chemical kinetics measures the rate of chemical reactions with time, in which the **order** of a reaction defines the nature of dependence between the speed of the reaction and the concentration of the reactants. In these reactions, the concentration y_i of a reactant is related to time x_i as follows:*

$$f(y_i) = w_1 x_i + w_2$$

Here, $f(y_i)$ is either y_i , $\ln(y_i)$, or $1/y_i$, depending on whether the reaction is of the zeroth order, first order, or second order, respectively. The absolute value of w_1 provides the rate of the reaction. How can the R^2 -statistic be used to identify the order of the reaction?

Solution: The key point is that one can regress each of the choices of $f(y_i)$ with x_i in order to compute R^2 and evaluate how well the linearity assumption holds. The model with the largest value of the R^2 -statistic is selected as the relevant one. ■

Example 7.17 *Give an example to show that RSS can be larger than TSS when regularization is added to the problem. Assume that you are using the feature-engineered version of regression with a feature value of 1 in lieu of the bias variable. Therefore, both coefficients (including the coefficient of the engineered feature) are included in the regularization. What happens to the sign of R^2 in such a case?*

Solution: Consider the case when the true linear relationship is of the form $y = 0.1x + 2$ and the three training examples are $\{(x, y)\} = \{(0, 2), (1, 2.1), (2, 2.2)\}$. An extremely large L_2 -regularization is incorporated in the model, as a result of which both coefficients are 0 and the learned model is $y = 0x + 0$. As a result, the predicted values of the dependent variable is always 0. In this case, RSS is $2^2 + 2.1^2 + 2.2^2 = 13.25$. On the other hand, the value of TSS is $(2 - 2.1)^2 + (2.1 - 2.1)^2 + (2.2 - 2.1)^2 = 0.02$. Therefore, RSS is much larger than TSS .

In this case, the value of R^2 is $1 - 13.25/0.02 = -661.5$. The sign of R^2 is negative in this case. In spite of the rather contrived example illustrated by this problem, RSS is rarely larger than TSS in practice. ■

The R^2 -statistic can also be expressed in terms of the sample outcome variance $\hat{\sigma}_Y^2$ and squared RMSE (and the result is valid only for the unregularized version of the problem on the training data). Since the sample variance $\hat{\sigma}_Y^2 = TSS/(n - 1)$ is the scaled version of TSS and the squared RMSE is the scaled version of RSS, it is relatively easy to show the following:

$$R^2 = 1 - \frac{RMSE^2 \cdot (n - k - 1)}{\hat{\sigma}_Y^2 \cdot (n - 1)} \approx 1 - \frac{RMSE^2}{\hat{\sigma}_Y^2}$$

The above approximation works very well when the number of predictors k is much smaller than the training data size n . In such a case, the ratio of $(n - k - 1)$ to $(n - 1)$ is almost 1.

Problem 7.8 Consider an unregularized regression setting in which the sample variance of the outcome on the training data is 9 and RMSE is 2.5. What is the R^2 -statistic for this regression if there are 1000 training instances and 9 non-trivial predictors?

Note that the number of training samples in the above problem is much larger than the number of non-trivial predictors. Work out the R^2 -statistic both with and without using the assumption that $k \ll n$ to explore how well the approximation for R^2 in terms of $RMSE$ works.

7.8.1.1 Correlation Versus R^2 -Statistic

It is useful to examine the case when $d = 2$ (including the predictor corresponding to the bias variable), and therefore, we have a single non-trivial predictor. In the case of a single non-trivial predictor, the R^2 -statistic turns out to have a useful relationship with the sample correlation:

Lemma 7.4 (R^2 Is Squared Correlation) For the case of regression with a single non-trivial predictor, the value of the R^2 -statistic on the training data is equal to the squared correlation between the outcome and the non-trivial predictor.

We omit a formal proof of this result. The sign of the correlation is the same as the sign of the slope because positive correlation means that the outcome increases with the value of the non-trivial predictor. As evident from Equation 7.2, the sign of the slope is that of $\hat{\sigma}_{XY}$.

Another interesting observation is that when $R^2 = 1$, the value of RSS/TSS must be 0, which can only happen when we have $RSS = 0$. This means that all residuals calculated for points in the training data are 0. The converse can also be shown to be true. The value of R^2 is 1 if and only if the squared correlation is 1 (or correlation is ± 1). Therefore, the coefficient of correlation between the predictor and the outcome is ± 1 if and only if all residuals of linear regression are zero — in other words, all training points lie on the regression line. The relationship of the R^2 -statistic to the (symmetric) correlation coefficient implies the following property for simple regression:

Observation 7.1 Consider a regression problem with a single predictor variable defined on a data set $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ with a single non-trivial predictor x_i and outcome y_i . An additional feature value of 1 is added to the regression predictor variables to account

for the bias variable. Then, the R^2 -statistic stays the same even if y_i is the predictor and x_i is the outcome.

The above observation shows the symmetric nature of R^2 -statistic in the case of a single non-trivial predictor.

Problem 7.9 Consider a regression problem with a single non-trivial predictor in which the sample variance of the outcome on the training data is 9 and the RMSE is 1.5. The slope of the estimated regression line is -2.34. Estimate the coefficient of correlation between the predictor and outcome assuming that the number n of training samples is large enough that $(n - 1)/n \approx 1$.

A hint for solving the above problem is to consider the relationship of the R^2 -statistic to RMSE and the variance of the outcome variable. The value of the R^2 -statistic and the sign of the slope tell you all you need to know about the correlation.

7.8.2 Out-of-Sample Evaluation

The previous section discussed some key measures of regression, which are evaluated on the training data. Unfortunately, these in-sample measures, when evaluated only on the training data, are not very reflective of performance on sample data not included in training, especially when training data sets are small. This is because the regression coefficients often overfit to the training data, and the values of RMSE and RSS are usually grossly optimistic (i.e., small). Far more accurate evaluations are obtained by measuring the performance on samples that are not included in the training data.

Consider the case where the model is constructed on the n pairs denoted by (\vec{x}_i, y_i) [for $i = 1$ to n], but a separate set of t pairs (\vec{x}_i^o, y_i^o) [for $i = 1$ to t], referred to as *test examples* are used to evaluate the model. We add a superscript o to indicate that these are out-of-sample points. In practice, the training pairs and test pairs are often derived from the same data set by *holding out* a subset of examples before building the mode;. These examples are used for evaluation. The error on the test data with coefficient vector \vec{w} (learned during model construction) is defined as follows:

$$\epsilon_i^o = y_i^o - \vec{w} \cdot [\vec{x}_i^o]^T$$

The RMSE and the R^2 statistic can be used to evaluate out-of-sample performance. The RMSE is now defined slightly differently, when t test points are used:

$$RMSE = \sqrt{\frac{\sum_{i=1}^t [\epsilon_i^o]^2}{t}}$$

Note that the denominator is no longer adjusted for the number of predictors, as out-of-sample examples are used for evaluation. Under the Gaussian assumption of errors, both the in-sample estimation of RMSE (with denominator adjustment) and the out-of-sample estimation of RMSE (without denominator adjustment) ought to provide the same value of RMSE. However, the Gaussian assumption is usually quite far from reality. In such cases, the out-of-sample estimations are far more accurate than in-sample estimations of the accuracy of the model. However, many of the theoretical properties of measures like the R^2 -statistic are lost on the test data. The R^2 -statistic is computed in a similar manner on out-of-sample data as for in-sample data:

$$R_o^2 = 1 - \frac{RSS^o}{TSS^o}$$

The subscript o has been added to R_o^2 to show that it is evaluated on out-of-sample data. Although RSS^o is practically always much less than TSS^o on the out-of-sample test data, a theoretical guarantee does not exist (as in the case of training data). It is possible to create contrived example settings in which RSS^o is greater than TSS^o on the test data (resulting in negative R_o^2).

Example 7.18 Create an example of a training set and a test set in which the out-of-sample RSS^o is greater than the out-of-sample TSS^o .

Solution: The main idea is to create an example in which training and test data are drawn from different distributions. In other words, the slope of the regression line on the training data is inconsistent with the slope of the regression line if the test data were to be used as training data. The regression line for the training data is $y = x$ with examples $(1, 1)$, $(3, 3)$, and $(5, 5)$. However, the test examples assume that the slope is of the opposite sign with examples $(2, -2)$, $(4, -4)$, $(6, -6)$. The predicted values on the test examples are 2, 4, and 6. This leads to an RSS^o value of $4^2 + 8^2 + 12^2 = 224$. On the other hand, the value of TSS^o is only $2^2 + 0^2 + 2^2 = 8$. The value of R_o^2 is negative. Such gross inconsistencies are rarely observed in real-world settings and therefore one would never actually observe this type of situation in the real world. ■

7.9 Nonlinear Regression

The linearity assumption of regression is almost never completely true in practice. For certain types of data sets, the relationship between the predictor and the outcome is almost never linear. As a specific example, consider the case where a chemist wants to identify the relationship between the temperature of water and its volume. The chemist repeatedly runs experiments and creates a scatterplot of the volume versus the temperature. The corresponding scatterplot is illustrated in Figure 7.5(a). It is immediately evident that the volume of water has a nonlinear relationship with the temperature. Another interesting example is shown in Figure 7.5(b) in which the variation in salary with age and education index is shown. The case of multiple predictors is particularly interesting because they can *interact* with one another in a variety of ways in order to create complicated relationships between the predictors and the outcome. There are several approaches for nonlinear regression with different strengths and weaknesses. The following approaches are examined in this chapter:

1. *Directly interpretable feature engineering:* This is the simplest approach for generalizing linear regression to the nonlinear case. In this approach, new variables are added to the data that are nonlinear functions of the original data. Applying linear regression to the expanded set of variables is equivalent to applying nonlinear regression on the original set of variables. The advantage of this approach is that the coefficients of the resulting features are often highly interpretable.
2. *Explicit feature engineering via similarity matrices:* An indirect way of representing an $n \times d$ data matrix D is with its similarity matrix $S = DD^T$ of dot products. The symmetric factorization of a dot product matrix S yields a matrix U containing

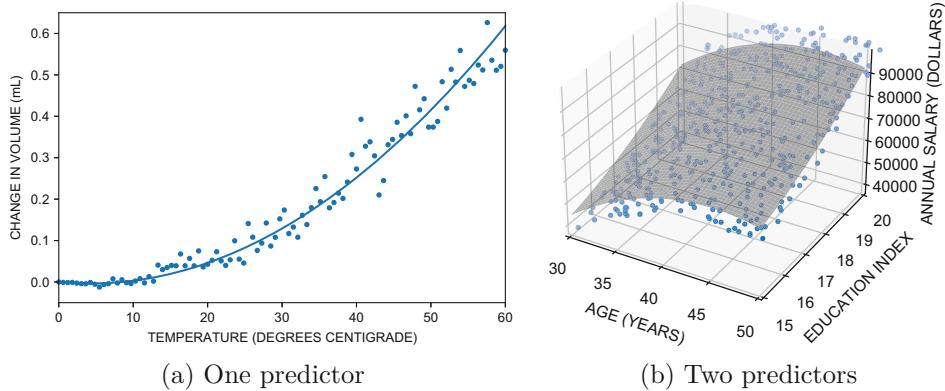


Figure 7.5: Simple examples of nonlinear regression

rotated/reflected versions of the rows of the data matrix D according to the following relationship:

$$S = UU^T$$

Here, U is an $n \times n$ matrix in which at least $\max\{n - d, 0\}$ columns will take on zero values and the remaining columns contain rotated/reflected versions of the rows of D . Note that one can factorize $S = UU^T$ in an infinite number of possible ways, all of which contain rotoreflections of D (including D itself). Whether one performs linear regression using data matrix D or with data matrix U , the final predictions of the regression on any (appropriately transformed) test data point will be exactly the same (although the optimal coefficients will be different). This is because U is a linearly transformed representation of D (e.g., rotation of points by 30°), which does not change the results of linear regression from a geometric point of view in spite of a different basis of representation (see Example 7.4). All one needs to do is to perform the same transformation on the test points (e.g., rotation of points by 30°).

What happens when one replaces the dot product function in each entry of S with a higher-order nonlinear function of the dot product before performing the factorization? It turns out that the symmetric factorization of such a similarity matrix yields an expanded set of features that are nonlinear functions of the original features. Furthermore, it is possible for more than d columns of U to be nonzero in order to accommodate the expanded feature set.

However, the precise functional relationships of these derived features with respect to the original features is often hard to represent in closed form. As a result, the approach loses its interpretability. Nevertheless, since the features are explicitly materialized, it presents several advantages in enabling preprocessing of features (to enhance the quality of predictions). However, the approach can be computationally expensive both in terms of processing and memory.

3. *Implicit feature engineering via similarity matrices:* It is possible to skip the step of factorizing the similarity matrix S — instead, one can directly use the entries of the similarity matrix for making predictions. This approach is equivalent to the use of explicit materialization. Skipping the explicit materialization step has the disadvantage that the underlying features cannot be preprocessed to enhance performance. However, the approach is computationally efficient, and therefore it is very popular. These techniques are referred to as *kernel methods*.

The following subsections will discuss the different nonlinear methods in detail.

7.9.1 Interpretable Feature Engineering

In this method, it is possible to create nonlinear representations of the original features on which it is possible to apply linear regression. In order to understand this point, consider the example of Figure 7.5(a) in which the volume of water is shown to be nonlinearly related to the temperature. In this case, the relationship between the outcome and the predictor is very close to being a quadratic polynomial. For the single predictor case, one can generate functions of increasing complexity by creating new features corresponding to the i th power of the original variable (for increasing i). By using the newly engineered features and treating the problem as one of performing linear regression with multiple features, it is possible to discover various types of nonlinear relationships mapping the predictor(s) to the outcome variable. For example, in order to discover a quadratic relationship between the outcome y_i and the predictor x_1 , we regress y_i against the two variables x_i and x_i^2 (along with the constant feature value of 1 whose coefficient is the bias):

$$y_i = w_1 + w_2 x_i + w_3 x_i^2 + \epsilon_i$$

Therefore, the data matrix needs to be expanded by adding a column corresponding to the feature value x_i^2 in each row.

A case involving two predictors is shown in Figure 7.5(b), in which the outcome (salary) depends on the age and the education index (which quantifies the level of education). The salary increases with both the age and the education index but the relationship is clearly nonlinear. This case is more challenging, because one cannot simply regress the outcome against a separable function of the polynomial features of the original variables. For example, consider the following function, which models the salary outcome y against the variables x_a for age and x_e for education index:

$$y \approx w_0 + w_{a1} x_a + w_{a2} x_a^2 + w_{e1} x_e + w_{e2} x_e^2$$

The main problem with the aforementioned model is that it is separable in age and education index. Therefore, it does not account for how age and education index interact with one another. For example, even though salary generally increases with age, the effect is less significant for people with low levels of education (i.e., unskilled workers). On the other hand, increasing age might present greater opportunities for advancement to highly educated workers. This type of effect can be addressed by introducing a multiplicative feature $x_a x_e$ that captures the interaction of age with the education index:

$$y \approx w_0 + w_{a1} x_a + w_{a2} x_a^2 + w_{e1} x_e + w_{e2} x_e^2 + w_{ae} x_a x_e$$

The coefficient w_{ae} tells us how much a combination of increased age and education contributes on average to increased salary. A positive value of w_{ae} indicates that a *combination* of increased age and increased education results in a higher salary than can be explained by the *individual additive effects* of age and salary. While it is possible to include interactions among more than two variables in the model, it is common to limit such settings to quadratic interactions among variables. Such models are referred to as *second-order models*. The number of possible interactions between d features increases as $O(d^2)$, which regulates the number of learned coefficients. Regularization is also very important in nonlinear models in order to avoid overfitting in the presence of an expanded feature set.

The aforementioned model illustrates the interaction between two numeric predictors. Interactions between categorical and numeric predictors are particularly insightful. For example, consider a situation where one has two predictors corresponding to age and race. The predictor for age is denoted by x_a . The race attribute has three values, corresponding to White, Black, and Hispanic. Assume that the reference category is White, and therefore, two dummy variables x_b and x_h are introduced (which take on the values of 1 for the Black and Hispanic categories, respectively). First, note that since the dummy variables are binary, there is nothing to be gained by squaring these values. Therefore, one can represent the second-order model corresponding to categorical-numeric interaction (i.e., age-race interactions for modeling salary) as follows:

$$y \approx w_0 + w_{a1}x_a + w_{a2}x_a^2 + w_{b1}x_b + w_{h1}x_h + w_{ab}x_ax_b + w_{ah}x_ax_h$$

It is helpful to understand the physical significance of coefficients such as w_{ab} and w_{ah} . First note that since the reference category is White, the quadratic coefficients w_0 , w_{a1} , and w_{a2} correspond to the effects of age for White individuals. Therefore, the model $y = w_0 + w_{a1}x_a + w_{a2}x_a^2$ represents the quadratic model for White individuals in which both x_b and x_h take on the value of 0. The coefficient w_{b1} indicates the addition to the intercept w_0 for Black individuals (over White individuals), and the coefficient w_{ab} indicates the additional first-order slope effects for Black individuals over White individuals. For example, a negative value of w_{ab} indicates that Black individuals gain $|w_{ab}|$ units less salary on average with one unit of increasing age than do White individuals. Similarly, a negative value of w_{b1} indicates a constant and age-independent disadvantage for Black individuals as compared to White individuals, although the specific numeric value of w_{b1} is hard to interpret in this case because it occurs at the intercept, where age is 0!

The main advantage of constructing nonlinear models with explicit feature engineering is the high degree of interpretability resulting from the model. One can interpret how the individual factors and their various combinations affect the outcome. The main disadvantage of the approach is that it is difficult to construct any model that uses more sophisticated relationships than second-order interactions. Furthermore, the increased number of coefficients can make the approach computationally expensive. For example, a data-matrix with 200 features will contain $\binom{200}{2} = 199,000$ second-order features, which can result in expensive model construction.

Example 7.19 Suppose you have a single feature and the values of independent-dependent pairs are expressed in the form (x_i, y_i) . There are no repetitions across different rows of x_i . Discuss why any training data of n instances can be expressed as a polynomial of at most the $(n - 1)$ th order, so that zero error is achieved on the training data. Is it advisable to use such a polynomial for regression?

Solution: One can generate n regressors as 1, x_i , x_i^2 , ..., x_i^{n-1} for each x_i . These regressors can be used to generate an $n \times n$ matrix D in which each row contains a set of polynomial regressors. The matrix D is a Vandermonde matrix, which can be shown to be invertible as long as there are no repetitions of x_i . Therefore one can find the regression coefficients to be $\vec{w} = D^{-1}\vec{y}$, where \vec{y} is a column vector containing the dependent variables. Note that this coefficient vector yields zero error on the training data since $D\vec{w} = DD^{-1}\vec{y} = \vec{y}$. Therefore, zero error can be achieved on the training data. It is not advisable to use a polynomial of such a high degree because the generalization performance will be poor on the test data. ■

Problem 7.10 Suppose you use the one-hot encoding approach (rather than the reference category approach) for creating binary variables from the race category in the age/race-to-salary regression on page 342. How many binary variables are created? How would you interpret the coefficients of the binary variables in this case, and how is it different from the reference category approach?

7.9.2 Explicit Feature Engineering with Similarity Matrices

The $n \times n$ similarity matrix S containing the dot products between the pairs of points in an $n \times d$ data matrix D is an indirect representation of the data set D . If the (i, j) th entry in the similarity matrix S contains the dot product between points (rows) i and j of D , then the similarity matrix S is related to the data matrix D using the following multiplication data matrix D with its transpose:

$$S = DD^T$$

Here, a key point is that even if one were given only the similarity matrix S rather than the data matrix D , one could recover a linear transformation of matrix D using symmetric matrix factorization:

$$S = UU^T$$

Here, U can be an $n \times d$ matrix, since using an $n \times n$ matrix for U will result in $(n - d)$ columns with zeros. The choice of U is not unique. One possible choice of U is D , but any rotoreflection of D (such as $U = DP$ for orthogonal⁷ matrix P) might be recovered as well. A rotoreflection of D refers to the matrix constructed by applying the same series of rotations and reflections to each row of D . One specific example of such a matrix is extracted using the eigen-decomposition of the matrix $S = DD^T$ into orthogonal matrix Q and diagonal matrix Σ as follows:

$$S = Q\Sigma^2Q^T = \underbrace{(Q\Sigma)}_U(Q\Sigma)^T$$

The matrix $Q\Sigma$ is the specific rotoreflection corresponding to *principal component analysis* in which all second-order correlations among feature variables are removed. *Regressing the outcome against the rows of a linear transformation U of the data matrix D will yield the same predictions as regressing the outcome against the rows of D .* This is because transforming the observations with a set of rotations and reflections can be compensated for by applying the corresponding inverse transformation to the weight vector \vec{w} (see Example 7.4).

So far, we have only seen that a linear transformation of the similarity matrix S can be used to extract a representation (via eigen-decomposition) of D that does not affect the results of regression. How is this result useful for non-linear regression? A useful result from linear algebra [6] is as follows:

Lemma 7.5 Let $S = DDT$ be an $n \times n$ matrix of dot products between the rows of $n \times d$ matrix D . Suppose that each entry s_{ij} of $S = [s_{ij}]$ is replaced by the entry $f(s_{ij})$, where $f(\cdot)$ is a polynomial of degree d to create the new matrix S_f . Then, for large enough $\lambda > 0$, the matrix $(S_f + \lambda I)$ can also be obtained by applying dot products to the rows of some $n \times s$ matrix D_f for some $s \leq n$. In other words, we have $S_f + \lambda I = D_f D_f^T$. The i th row of D_f contains multivariate polynomial transformations $[g_1(\vec{x}_i), g_2(\vec{x}_i), \dots, g_s(\vec{x}_i)]$ of the i th row \vec{x}_i of data matrix D .

⁷A $d \times d$ orthogonal matrix P is a square matrix satisfying $PP^T = P^TP = I$. Multiplying any data matrix D with P results in a series of rotations and reflections, depending on P [6].

The addition of λI to the diagonal is referred to as *conditioning*. Functions that require conditioning are sometimes not practically useful because they might convert large positive similarities to large negative similarities and vice versa. An example of such a function is $f(s_{ij}) = -s_{ij}^2$. As a practical matter, the function $f(s_{ij})$ should magnify the similarities in matrix entries in a superlinear way without changing the relative order of similarities in a significant way. A useful class of functions, referred to as *positive semi-definite kernels*, do not require conditioning and λ can be set to 0. This section will only focus on such (practically useful) functions and therefore $S_f = D_f D_f^T$.

The i th row of matrix D_f contains the engineered representation $[g_1(\vec{x}_i), g_2(\vec{x}_i), \dots, g_s(\vec{x}_i)]$ of the i th row \vec{x}_i of D . The usefulness of the engineered representation depends on the nature of the function $f(\cdot)$. The following are the two most common kernels used in the machine learning literature:

$$f(s_{ij}) = \begin{cases} (c + s_{ij})^{\dim} & [\text{Polynomial kernel}] \\ \exp(s_{ij}/(2\sigma^2)) & [\text{Gaussian kernel}] \end{cases}$$

It is noteworthy that a Gaussian kernel can also be considered a polynomial in infinite dimensions. The hyper-parameters $c \geq 0$, $\dim \in \mathbb{N}$, and σ regulate the effectiveness of the kernel for the specific problem at hand. These hyper-parameters are tested on held out portions of the training data (i.e., portions of the training data not used to calculate the vector \vec{w}) in order to compute the RMSE and select those values that minimize the RMSE.

We will now describe the nonlinear regression technique based on the discussion above and the kernel function $f(s_{ij})$. Consider the situation, where one has a data matrix D , and the corresponding dot-product matrix is $S = DD^T$. We construct the kernel matrix $S_f = [f(s_{ij})]$, in which the (i, j) th entry is $f(s_{ij})$. This matrix S_f is decomposed via eigen-decomposition in order to extract an $n \times n$ matrix U with rows containing engineered features:

$$S_f = Q\Sigma^2Q^T = \underbrace{(Q\Sigma)}_U(Q\Sigma)^T$$

Here, Q and Σ are $n \times n$ matrices, and one can represent the diagonal matrix in squared form (Σ^2) because it is positive semidefinite with nonnegative eigenvalues. In the event that some of the diagonal entries of Σ are 0, the corresponding columns in U will be 0 and can⁸ be dropped. One can also write U as $Q_0\Sigma_0$, where Σ_0 is a smaller $p \times p$ matrix containing only nonzero diagonal entries, and Q is an $n \times p$ matrix whose columns contain only nonzero eigenvectors. The rows of matrix U contain the engineered representation of the data. The number of columns of the matrix $U = Q_0\Sigma_0$ after dropping the zero columns is p . When none of the diagonal entries of Σ are 0, it is possible for the rows of U to be n -dimensional (corresponding to the engineered rows of D). Note that the number of observations n is usually larger than the (original) dimensionality d , and therefore the engineered representation can have a higher dimensionality than the original data. This is consistent with our observations in the previous section, where the number of interaction variables can be quadratically related to the dimensionality d .

Once the $n \times p$ matrix U has been constructed, a p -dimensional weight vector \vec{w} can be learned using the following relationship between the features and the n -dimensional column vector \vec{y} :

$$U\vec{w} = \vec{y}$$

⁸For example, using the identity function for $f(\cdot)$ results in at most d columns of U to be nonzero. This is because the resulting representation is a rotoreflection of the original data.

One use gradient-descent to find \vec{w} . Let D_t be a $t \times d$ matrix containing the out-of-sample test instances whose outcomes need to be predicted. First, we need to construct the engineered representations of D_t in a manner consistent with how the training matrix D was mapped to U (and therefore the engineered representations of the out-of-sample data will also contain p variables). First, we construct the $t \times n$ matrix S_t containing the similarities between the out-of-sample and in-sample data. This similarity matrix is obtained by applying the kernel function $f(\cdot)$ to each of the entries in $D_t D^T$. Let U_t be the $t \times p$ engineered representation of the out-of-sample data. Then, U_t must satisfy the following relationship:

$$U_t U^T = S_t$$

This is because the dot products between the engineered representations of the out-of-sample and in-sample data are contained in $U_t U^T$. Substituting $U = Q_0 \Sigma_0$, one obtains the following:

$$U_t \Sigma_0 Q_0^T = S_t$$

On multiplying both sides with $Q_0 \Sigma_0^{-1}$ and using the orthonormality of the columns of Q_0 , one obtains the following:

$$U_t = S_t Q_0 \Sigma_0^{-1}$$

Note that U_t is a $t \times p$ matrix, and each row contains the engineered representation of a test instance. Since the weight vector \vec{w} has been found with respect to this engineered representation, the predictions for the t test instances may be found in the entries of the t -dimensional vector $U_t \vec{w}$. The i th entry of this vector contains the prediction for the test instance i .

This approach is able to capture higher-order interactions (rather than only second-order interactions) between attributes. However, it is not interpretable because the individual columns of U do not have comprehensible relationships with the columns in the original data matrix D . Therefore, no special interpretation attaches to the p coefficients in \vec{w} . A second problem with the approach is that the matrix U can potentially be of size $n \times n$, which is rather large even for modest values of n . In the modern era, a data set containing a million observations would be considered very modest, but a matrix of size $10^6 \times 10^6$ would require 4 Terabytes even at 4 bytes per entry. This can be a computational problem. One issue here is that it is not necessary to use all the columns in U . Rather, one can use only the top few columns of Q corresponding to the largest eigenvalues in order to generate Q_0 and Σ_0 . In many cases, the columns corresponding to the smallest eigenvalues contain noisy and outlier deviations, which are not relevant for applications like regression. Another advantage of this approach is that it is possible to drop some of the features based on low correlations with the target vector \vec{y} . After the features with low eigenvalues are dropped, the coefficient of correlation is computed between each column of U and the target vector \vec{y} . Only those columns that have large (absolute) correlations with the target vector are retained in Q_0 and Σ_0 to construct both U and U_t .

Example 7.20 Suppose you have n paintings (where n is even) and you inspect them pairwise to rate them as similar to one another (rating of 1) or dissimilar to one another (rating of -1). You create an $n \times n$ similarity matrix using these ratings. This matrix can be divided into four $(n/2) \times (n/2)$ blocks. The two blocks along the diagonal have 1s, whereas all other values are -1. Note that all self-similarity values are 1. Find a 1-dimensional representation of the paintings.

Solution: This matrix can be shown to have a single eigenvector in which the first $n/2$ values are 1 and the remaining $n/2$ values are -1 . Therefore, the first $n/2$ paintings are represented by 1 and the remaining $n/2$ paintings are represented by -1 . Usually, the $n \times k$ eigenvector matrix Q is scaled as $Q\Sigma$ in order to adjust the relative influence of different eigenvectors in the representation. The scaling of the eigenvector in this case is unimportant since it is a 1-dimensional representation. ■

Problem 7.11 You have a set of three expensive paintings and you define pairwise similarity between them using the 3×3 matrix S . You want to regress the individual paintings against a dependent variable corresponding to the price. The matrix S is as follows:

$$S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

Self-similarity values (on the diagonal of S) are always set to 1. Find a reasonable 2-dimensional representation of the paintings for regression. You are now given a fourth painting for which the similarities to the other three paintings in same order as columns of S are 0.5, -0.5 , and 0.5. Predict the price of the new painting using regression.

A hint for solving the problem is to express S approximately by UU^T for 3×2 matrix U .

7.9.3 Implicit Feature Engineering with Similarity Matrices

The main disadvantage of the approach discussed in the previous section is that it is computationally expensive. It turns out that it is not necessary to explicitly compute the features as long as one does not need to perform additional enhancements like dropping features with low eigenvalues or low correlations to the target vector. As long as one is willing to drop only features with zero eigenvalues, the approach discussed in the previous section can be implemented very efficiently. Furthermore, while it makes sense to execute the approach of the previous section with gradient-descent, the closed-form solution is useful because it can be implemented with the use of similarity matrices.

Let U and U_t be the engineered representations of the training and test data matrices using the notations of the previous section. Then, the weight vector \vec{w} can be represented in closed form as follows:

$$\vec{w} = (U^T U)^{-1} U^T \vec{y} \quad (7.10)$$

A key point is that the *push-through identity* in linear algebra [6] implies the following:

$$(U^T U)^{-1} U^T = U^T (U U^T)^{-1}$$

Therefore, the weight vector may be expressed as follows:

$$\vec{w} = U^T (U U^T)^{-1} \vec{y}$$

One can then multiply the weight vector \vec{w} with U_t in order to create a t -dimensional vector \vec{y}_t of predictions on test instances:

$$\hat{\vec{y}}_t = U_t \vec{w} = \underbrace{(U_t U^T)}_{S_t} \underbrace{(U U^T)^{-1}}_{S^{-1}} \vec{y}$$

By using the fact that $U_t U^T$ is the $t \times n$ matrix of dot-product similarities between the engineered representations of testing and training instances, it can be inferred that $S_t = U_t U^T$. Similarly, $S = U U^T$ is the symmetric matrix of similarities between the engineered representations of training instances. Therefore, the predictions on test instances can be represented as follows:

$$\hat{y}_t = S_t S^{-1} \vec{y} \quad (7.11)$$

In other words, the predictions on test instances can be expressed purely in terms of the regressand vector of training instances as well as the similarities (a) between test and training instances, and (b) between pairs of training instances.

An observation is that this approach assumes that S_t is invertible. This may not always be the case. For example, when the similarity function contains vanilla dot products of the rows of an $n \times d$ data matrix D , the matrix S will not be invertible when $d < n$. This is because any dot product matrix $S = D D^T$ has rank no larger than that of D (and therefore rank at most d). In such cases, the model is still linear and S^{-1} would need to be replaced with its pseudo-inverse S^+ in order to compute the predictions:

$$\hat{y}_t = S_t S^+ \vec{y}$$

However, choosing a Gaussian kernel to create the similarity matrices S and S_t will typically lead to an invertible matrix S .

One issue with nonlinear regression is that it often provides excellent predictions on the training data but is unable to provide similar accuracy on the test data. Such a situation is often the result of over-fitting. Therefore, nonlinear models do not always result in better predictions, particularly when sufficient data are not available.

Example 7.21 Show that if the kernel regression model is applied to the n instances of the training data to make predictions in cases where the $n \times n$ training similarity matrix S is invertible, zero error is obtained on each instance of the training data. In other words, each regressand is predicted perfectly.

Solution: In this case, the prediction is being performed on the training instances, and therefore we have $S_t = S$. The prediction on the training instances is therefore as follows:

$$\hat{y} = S_t S^{-1} \vec{y} = [S S^{-1}] \vec{y} = \vec{y}$$

Therefore, the predicted vector \hat{y} is the same as the ground-truth vector \vec{y} . In many cases, perfect prediction on the training data is a sign of over-fitting. In such cases, regularization becomes particularly important for accurate predictions on out-of-sample data. ■

Example 7.22 (Regularized Nonlinear Regression) Show that Equation 7.6 for finding the weight vector in regularized linear regression can be used to create an expression for the prediction of the t -dimensional vector \vec{y}_t in regularized nonlinear regression as follows:

$$\vec{y}_t = S_t (S + \lambda I)^{-1} \vec{y}$$

Here, $\lambda > 0$ is the regularization parameter.

Solution: Let U and U_t be the engineered representations of the training and test data matrices (using the same notations as the unregularized version of the problem). Then, the weight vector \vec{w} for the regularized version of the problem can be expressed as follows:

$$\vec{w} = (U^T U + \lambda I)^{-1} U^T \vec{y}$$

The push-through identity in linear algebra [6] implies that the above expression is equivalent to the following:

$$\vec{w} = U^T (U U^T + \lambda I)^{-1} \vec{y}$$

By pre-multiplying the weight vector \vec{w} with U_t , one can create a t -dimensional vector \vec{y}_t of predictions on test instances:

$$\vec{y}_t = U_t \vec{w} = \underbrace{(U_t U^T)}_{S_t} \underbrace{(U U^T + \lambda I)^{-1}}_{(S + \lambda I)^{-1}} \vec{y}$$

Therefore, the predictions on test instances can be represented as follows:

$$\vec{y}_t = S_t (S + \lambda I)^{-1} \vec{y}$$

■

It is noteworthy that kernels are associated with parameters that need to be tuned. For example, the Gaussian kernel is associated with the bandwidth parameter σ whose optimal value needs to be determined. One way of performing the tuning is to hold out a small subset of the training instances and predict them with the remaining training data for varying values of the tunable parameter(s). The value(s) of the parameter(s) at which the RMSE is optimized is reported as the relevant one and then used for prediction on unseen test instances.

The main advantage of this approach is that it can be implemented efficiently. At first glance, it might seem that the inversion of the matrix S is expensive. First, note that the matrix inversion does not need to be performed on the full-rank matrix, but it can be performed approximately using the pseudo-inverse on a lower rank approximation. Specifically, the rank- k approximation S_k of S using its eigen-decomposition is as follows:

$$S_k = Q_k \Sigma_k^2 Q_k^T$$

Here, Q_k is an $n \times k$ matrix containing the top- k eigenvectors and Σ_k^2 is a diagonal matrix containing the nonnegative eigenvalues. Once the matrices Q_k and Σ_k have been computed, the predictions for the t test instances may be evaluated as follows:

$$\vec{y}_t \approx S_t [S_k^{-1} \vec{y}] = S_t [Q_k (\Sigma_k^{-2} [Q_k^T \vec{y}])]$$

The nested brackets indicated above show the order of multiplication of the various matrices. This order of multiplication ensures the most compact representations of the intermediate matrices during the materialization of \vec{y}_t . Although the solution above is more efficient than explicit feature engineering, the latter has the advantage that it is possible to preprocess the features (with techniques like correlation-based feature selection). It is not possible to implement such enhancements with implicit feature engineering. Implicit feature engineering is also referred to as the *kernel trick*.

7.10 Summary

This chapter introduces regression, which is one of the fundamental problems in linear algebra, statistics, and machine learning. The loss function of linear regression problem has a probabilistic interpretation based on maximum-likelihood estimation. Linear regression allows both closed-form solutions as well as solutions that require gradient descent. The problem of overfitting in linear regression was discussed along with solutions based on regularization. A connection between regularization and maximum a posteriori (MAP) estimation was presented. Nonlinear regression methods were introduced along with several alternative solutions, which provide different trade-offs between interpretability, preprocessing enhancements, and computational efficiency.

7.11 Further Reading

The problem of regression is discussed in different data mining and machine learning textbooks [1, 3, 10, 33]. A discussion of regression from the linear algebra perspective may be found in [6, 59]. A specialized discussion of regression analysis may be found in [20, 25]. The book by Fox [25] also provides insights into generalized linear models. The connections between regression and outlier detection are discussed in [55].

7.12 Exercises

1. You append a value of 2 (instead of a value of 1 as discussed in the text) to each observation as an additional attribute to model the bias. Discuss the difference of this model from one that uses a value of 1 for the additional engineered feature.
2. Consider the linear regression problem with a single non-trivial predictor and an engineered feature with a value of 1 for the bias. No regularization is used. Suppose that the data are normalized in such a way that the variance along the predictor and outcome variables is the same. Express the slope coefficient in terms of the coefficient of correlation between the predictor and outcome variables.
3. Consider a regression problem with a single non-trivial predictor variable. The covariance between predictor and outcome variable is 3 and the slope of the regression line is 1/3. Can you find the variance of the predictor or outcome variable or both?
4. Consider an unregularized regression setting in which the outcome variance on the training data is 16 and RMSE is 1.0. What is the R^2 -statistic if the number of predictors is negligibly small compared to the number of training instances?
5. Suppose that the variance on the outcome is 8. Estimate the RMSE for the case of a single non-trivial predictor when the coefficient of correlation between predictor and outcome is 0.73. Assume that the number of training instances is large.
6. Consider a linear regression problem in which you are given samples $\{(\vec{x}_i, y_i) : 1 \leq i \leq n\}$ to construct the training model. Suppose that you scale one of the predictors by a factor of 2 before building an optimized model to find the weight vector \vec{w} . Will the optimal value of the least-squares loss change because of this modification to the training data? Why or why not? Assume that no regularization is used.

7. How would your answer to the previous exercise change if regularization is used?
8. The US government releases the purchasing manager's index (PMI) each month, which is a numerical index of economic activity. The PMI for a given month is known to be influenced by the history of past PMIs, although any data more than six months earlier is generally considered too stale to be relevant. Discuss how you can predict the next month's PMI using regression modeling.
9. Consider a variation of the problem in Exercise 8, in which the PMI, inflation data, and unemployment rates are released on the same day each month. As in Exercise 8, data more than six months old has little predictive power. These data are known to be related to one another. Propose a regression model that predicts all three values to be released next month.
10. Consider a regression data set of 10 points with a single independent variable X and a dependent variable Y . A bias is also learned using an engineered feature value of 1. The data set has the following characteristics: (i) $\sum_{i=1}^{10} x_i^2 = 25$; (ii) $\sum_{i=1}^{10} x_i y_i = 30$; (iii) $\hat{\mu}_X = 1$; and (iv) $\hat{\mu}_Y = 2$. Find the optimal linear regression line.
11. Consider a simple regression problem in which you standardize the dependent and independent variables and call them Y and X , respectively. You find that the mean value of $x_i y_i$ over all points is 0.5. What is the optimal regression line?
12. The volume of water varies with temperature $t \in [0, 100]$ based on a polynomial dependency. You obtain several hundred temperature-volume pairs of 1 cm³ of 0° Celsius water using experimentation. Discuss a regression approach to learn the polynomial function $f(t)$ that provides this volume. What considerations should you use in setting the maximum degree of this polynomial?
13. Consider a simple regression problem with a (single) standardized independent variable X . The dependent variable Y has not been standardized and its mean is 2.7. The value of $\hat{\sigma}_{XY}$ is 0.9 in your data set. What is the optimal regression line?
14. For the problem in Exercise 13, you multiply the dependent variable by 2 and the independent variable by 3. What is the optimal regression line after the modification?
15. Suppose that you find the optimal regression line for a single predictor problem to have a slope of 0.3, where both the independent variable X and dependent variable Y are standardized. You find out that you had switched the data for the dependent and independent variables by mistake. What is the correct optimal regression line?
16. Suppose that you have n independent-dependent training pairs (x_i, y_i) . Jim claims that polynomial regression of degree n will always give zero error on the training data. Give a counter-example to show that this is not always true. [Hint: The training error is almost always zero. You will have to find a case in which the rows of polynomial regressors are linearly dependent in spite of the fact that the rows of Vandermonde's matrix (cf. Example 7.19) are linearly independent.]
17. Consider a d -dimensional regression problem with n whitened training instances. In other words, the feature means, variances, and covariances are 0, 1, and 0, respectively. Show that the optimal weight vector \vec{w} of linear regression is $\sum_{i=1}^n y_i \vec{x}_i / n$.

- 18.** Propose an algorithm that uses regression to find outlier entries in an $n \times d$ data set.
[Hint: What does a large residual in regression imply?]
- 19.** A k -nearest neighbor regressor outputs the average regressand of the k nearest points to the test instance. Suppose that you predict the regressand of test point \vec{z} from n training points denoted by (\vec{x}_i, y_i) as $\beta \sum_{i=1}^n y_i K_h(\vec{z}, \vec{x}_i)$ where $\beta = 1 / \sum_{i=1}^n K_h(\vec{z}, \vec{x}_i)$. Here, $K_h(\vec{z}, \vec{x}_i)$ is a kernel function drawn from kernel density estimation with bandwidth h . Show why this approach is a generalized version of k -nearest neighbor regressor in which the bandwidth plays a similar role to the value of k .
- 20.** Let $S = [K(\vec{x}_i, \vec{x}_j)]$ be the $n \times n$ training similarity matrix defined for kernel regression with regressand vector $\vec{y} = [y_1, \dots, y_n]^T$. Argue that the prediction function of kernel regression (cf. Equation 7.11) is a generalized version of k -nearest neighbor regression (cf. Exercise 19) in which the prediction of the test point \vec{z} is $\sum_{j=1}^n \beta_j(\vec{z}) y_j$. How can you compute the function $\beta_j(\vec{z})$ in terms of training-training and training-test similarities?
- 21.** Suppose that you create fine-grained clusters $\mathcal{G}_1 \dots \mathcal{G}_k$ from the independent variables of a regression data set. The clusters have average regressand values of $y'_1 \dots y'_k$ respectively. Can you make the approach in Exercise 19 more efficient by using ideas drawn from Exercise 27 in Chapter 6?
- 22.** Suppose that you have the $n \times n$ binary friendship matrix $A = [a_{ij}]$ for a social network with n actor nodes (where $a_{ij} = 1$ when actors i and j are friends). Nodes exhibit *homophily* because connected nodes have similarity in attribute values. A subset of the nodes contain the value for a particular numeric attribute. Discuss how you can use kernel regression to predict this attribute value for nodes where it is missing.
- 23.** Discuss the steps for implementing Exercise 22 with explicit feature engineering.



Chapter 8

Classification: A Probabilistic View

Knowledge is the small part of ignorance we arrange and classify. — Ambrose Bierce

8.1 Introduction

The previous chapter introduces the regression modeling problem, which predicts numeric outcomes from predictor variables. What happens in cases where the outcome variable is binary or categorical? Such a change to the regression problem definition results in an important and different problem, which is referred to as that of *classification*.

In the classification problem, we have a set of *labeled training instances*, denoted by $(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_n, y_n)$. Each \vec{x}_i is a d -dimensional row vector of numeric *feature variables* or *predictors*, and y_i contains a *categorical* or *binary class variable*. By default, we will assume that \vec{x}_i is a numeric vector, although it is not difficult to generalize most models to cases in which \vec{x}_i contains categorical components. The goal of the classification problem is to build a model, so that y_i is predicted as a function of \vec{x}_i :

$$y_i \approx f(\vec{x}_i)$$

The output of $f(\cdot)$ is either a binary value from $\{+1, -1\}$ or a 1-hot encoded vector of k dimensions in which the component corresponding to 1 contains the class prediction out of k possible classes. Therefore, classification is a very similar problem to that of regression, except that the dependent variable is now binary or categorical. After the function $f(\cdot)$ has been learned, it is typically used to make predictions on out-of-sample test data. In other words, for a given test instance \vec{z} , its class label is predicted as follows:

$$\hat{y} = f(\vec{z})$$

The classification problem is naturally defined in settings where the data is segmented into labeled groups. A real-world example is a credit-approval application in which the observations $\vec{x}_1 \dots \vec{x}_n$ correspond to the financial transactions of individuals and each outcome

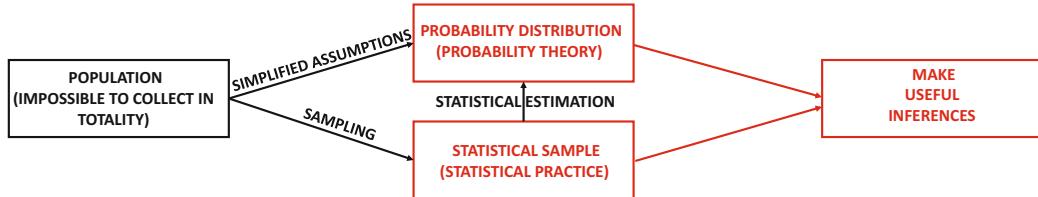


Figure 8.1: Revisiting Figure 1.6: How reconstructed distributions can be used for making useful inferences in machine learning.

variable y_i is either *normal* or *high-risk*. The outcome variable is encoded as a binary value. The goal is to create a model relating the observations \vec{x}_i to the risk-level y_i . In this case, the class label y_i is binary, and is often encoded as is $\{-1, +1\}$. In the case of categorical labels, it is assumed that there are a total of k labels, and the indices of the k labels are drawn from $\{1, 2, \dots, k\}$. Both types of models are discussed in this chapter.

In order to build classification models, the distributions of either the feature variables in each class or the distributions of feature-class relationships need to be reconstructed. This overall process of making inferences from reconstructed distributions is illustrated in Figure 8.1). This chapter will focus on the part marked in red in the figure. In *generative models*, it is assumed that the data are created by a specific generative process such as a mixture model in which each class is a component of the mixture. The parameters of the distributions in this generative process are estimated (using the maximum-likelihood estimation techniques discussed in Chapter 6). Other than generative models, some techniques model the probability distributions of the class labels (or their errors) as a function of the feature variables, and these are referred to as *discriminative models*. An example of a discriminative model is the *logistic regression classifier*.

8.1.1 Chapter Organization

This chapter is organized as follows. The next section introduces generative probabilistic models with a particular focus on the Bayes classifier. Loss-based formulations are discussed in section 8.3. Many popular techniques in machine learning, such as least-squares classification, logistic regression, and multinomial logistic regression, fall into this category. Section 8.4 moves beyond the classification problem and introduces methods for ordinal regression with the use of the ordered logit model. A summary is given in section 8.5.

8.2 Generative Probabilistic Models

Generative probabilistic models assume that each of the classes in the data is generated from a specific probability distribution that varies by class. Such models are referred to as *mixture models*, where the different classes correspond to different components of the mixture. A detailed discussion of the generative process for mixture models is provided in section 4.13.2. The following presentation will replicate that discussion in the context of the classification problem (i.e., with annotations that are specific to the classification problem). Furthermore, the discussion below is generic and it will not assume any specific form of the probability distribution of each component of the mixture. Subsequently, the cases corresponding to different types of probability distributions will be discussed. The specific form of the probability distribution of each component depends heavily on the type of data

domain from which the feature variables are drawn. For example, the features of a low-dimensional numeric data set are often assumed to be drawn from a Gaussian distribution, whereas the features of a data set containing binary features are assumed to be drawn from a Bernoulli distribution. Similarly, a high-dimensional sparse data set containing sparse (integer) features are assumed to be drawn from a multinomial distribution with as many categories as the number of features. The probability distribution of the full population is denoted by $f_{\vec{X}}(\vec{x})$ (corresponding to all the classes). The full probability distribution can be decomposed into mixture components that have their own conditional probability distributions. Therefore, the entire data set is a union of mixture components. The following discussion will assume that the population contains k classes, and each class has a different generating process corresponding to its conditional probability distribution. For example, a credit-risk classification application may contain individuals belonging to the *high-risk* class and the *low-risk* class, and therefore the value of k is 2. In such cases, the probability distribution of the financial transactions of a person from the *high-risk* class would be different from those of a person belonging to the *normal* class. The data set can, therefore, be assumed to be a mixture of two distributions with relative proportions defined by the percentage of high-risk individuals.

We assume that the different mixture components (representing the conditional probability distributions of the k classes) are denoted by $\mathcal{G}_1, \dots, \mathcal{G}_k$, with corresponding to conditional probability distributions denoted by $f_{\vec{X}|\mathcal{G}_1}(\vec{x}), f_{\vec{X}|\mathcal{G}_2}(\vec{x}), \dots, f_{\vec{X}|\mathcal{G}_k}(\vec{x})$. Therefore, there are a total of k mixture components. In the case when the feature variables in data are discrete (e.g., binary attributes), one can use probability mass functions $p_{\vec{X}|\mathcal{G}_1}(\vec{x}), p_{\vec{X}|\mathcal{G}_2}(\vec{x}), \dots, p_{\vec{X}|\mathcal{G}_k}(\vec{x})$ instead of probability density functions. We will give examples of both types of data sets, in which the individual mixture components are continuous (i.e., represented by probability density functions) and discrete (i.e., represented by probability mass functions). The specific choice of the probability distribution depends heavily on the data domain at hand. For example, a data set containing word frequencies of text documents in its rows would be represented by a very different choice of data distribution as compared to a data set containing numerical values of financial transactions. The probability that the i th mixture component is selected in the generation of an instance is given by $P(\mathcal{G}_i)$. Therefore, we have:

$$\sum_{i=1}^k P(\mathcal{G}_i) = 1$$

The probability $P(\mathcal{G}_i)$ is the *prior* probability of an observed data point belonging to mixture component i , because it refers to the probability that we would predict an observation to belong to mixture component i without knowing anything about the data point. Intuitively, one can view these values as the relative frequencies of the different classes in the observed data. For example, in the credit-risk classification application, the number of members in the *high-risk* class may be far fewer than the number of members of the *normal* class. Therefore, even without knowing anything about the attributes of a particular observation (i.e., the financial transactions of a particular individual), one can assign a higher prior probability to the *low-risk* class as compared to the *high-risk* class. Of course, the goal in classification is to predict the *posterior* probabilities of test instances belonging to one of these classes after having access to their attribute values (i.e., financial transactions). The basic generative process for a single observation in the training data is as follows:

1. Roll a biased die whose k sides have probabilities $P(\mathcal{G}_1) \dots P(\mathcal{G}_k)$. Let the outcome of the die roll be the side r , which provides the identity of the mixture component from

which the observation is generated. In other words, the observation is generated from class r .

2. Sample the data point \vec{x} from the probability distribution $f_{\vec{X}|\mathcal{G}_r}(\vec{x})$. The point \vec{x} is the output of one iteration of the generative model.

In the classification problem, the assumption is that this generative process can be repeated again and again in order to generate a data set of exactly the same size as the number of training samples. Furthermore, the outcome of the first step of the generative process (i.e., the roll of the die yielding the class identity) of each training sample is available in the training data (as class labels). A *maximum-likelihood estimation process* is used to estimate the parameters of the distribution and the prior probabilities. The availability of the class labels with the training data is very helpful in making estimations of distribution parameters. The above generative process is an example of how a classification data set is often assumed to be an outcome of a compound distribution represented by the aforementioned pair of generative steps.

The clustering problem can also be modeled by the same generative process. The availability of the outcome of the first roll of the die in the form of the class labels in the training data is the main difference between the classification and clustering problems. The lack of availability of labels in clustering creates challenges in the parameter estimation process and necessitates the use of an *iterative* parameter estimation process (cf. section 6.4), which is referred to as the expectation-maximization algorithm. As we will see in this section, the parameter estimation process in the case of the classification problem is much simpler.

After the prior probabilities and the parameters of the probability distributions have been estimated, the Bayes rule can be used to compute the posterior probabilities of individual test instances. This process is not very different from the 1-dimensional Bayes classifier discussed in section 3.8.3 of Chapter 3. The main difference is that the classifier in section 3.8.3 acquires its probability distribution parameters and prior probabilities through domain-specific inputs, whereas the predictive approach of this section acquires its distribution parameters with the use of maximum-likelihood estimation over the observed data in the context of a generative model.

Consider an out-of-sample test instance \vec{z} that needs to be classified into one of the k mixture components (classes) denoted by $\mathcal{G}_1 \dots \mathcal{G}_k$. Therefore, we want to determine the probability $P(\mathcal{G}_r|\vec{X} = \vec{z})$. This probability can be determined using the Bayes rule (cf. Equation 4.8 in section 4.13.3 of Chapter 4):

$$P(\mathcal{G}_r|\vec{X} = \vec{z}) = \frac{P(\mathcal{G}_r)f_{\vec{X}|\mathcal{G}_r}(\vec{z})}{\sum_{s=1}^k P(\mathcal{G}_s)f_{\vec{X}|\mathcal{G}_s}(\vec{z})} \quad (8.1)$$

The key point here is that all quantities on the right-hand side can be estimated using maximum-likelihood estimation on the training data. Therefore, the posterior probability can also be estimated from these quantities by using the above relationship. In the case of discrete random variables, the notation $f_{\vec{X}|\mathcal{G}_r}(\vec{z})$ for a probability density function is replaced with the probability mass function $p_{\vec{X}|\mathcal{G}_r}(\vec{z})$ in the above equation.

An important point is that the parameters of the joint distribution $f_{\vec{X}|\mathcal{G}_r}(\vec{x})$ can become challenging to estimate during maximum-likelihood estimation as the dimensionality of the data increases. Joint distributions often have a notoriously large number of parameters. In order to reduce the number of parameters in the joint distribution, it is common to make the assumption of *conditional independence*, wherein a joint probability density (or mass) function is the product of the corresponding marginal probability density (or mass) functions.

In other words, the joint probability density (or probability mass function) of a sample observation $[x_1, x_2, \dots, x_d]$ corresponding to the random variables vector $[X_1, X_2, \dots, X_d]$ satisfies the following relationship:

$$p_{\vec{X}|\mathcal{G}_r}(\vec{x}) = \prod_{j=1}^d p_{X_j|\mathcal{G}_r}(x_j) \quad [\text{Discrete Random Variables}]$$

$$f_{\vec{X}|\mathcal{G}_r}(\vec{x}) = \prod_{j=1}^d f_{X_j|\mathcal{G}_r}(x_j) \quad [\text{Continuous Random Variables}]$$

The logic for the use of conditional independence is provided in section 3.8.2 of Chapter 3. When the notion of conditional independence is applied in the context of a Bayes classifier, it is referred to as a *naïve Bayes classifier*. The use of the word “naïve” is indeed an acknowledgement of the fact that the conditional independence assumption is somewhat of an oversimplification; nevertheless, it is a reasonable one and tends to give good results in practice. It is also noteworthy that the need to use the naïve assumption depends heavily on the complexity of the underlying joint distribution — in some cases, this assumption is dropped.

In the following, three different settings of the Bayes classifier will be discussed. The first setting corresponds to continuous numeric data in which a Gaussian distribution is used to model each component of the mixture distribution. The second setting corresponds to binary data in which a multivariate Bernoulli distribution (with conditional independence) is used to model each component of the mixture distribution. The third setting corresponds to discrete and sparse numeric data in which multinomial distributions are used to model different components of the mixture. All these methods heavily use the machinery for reconstructing distributions, as discussed in Chapter 6. The main difference in all these cases is the way in which the distribution is reconstructed with the use of maximum likelihood estimation. When limited data are available, one can also switch to maximum a posteriori (MAP) estimation for better reconstruction. Both the Gaussian and the Bernoulli distributions use conditional independence. However, it is not natural to use conditional independence with the multinomial distribution, which explicitly uses the dependence between different dimensions to compute joint probabilities. Furthermore, it is also possible to use the Gaussian distribution without the conditional independence assumption. This is because the number of parameters required to represent a generic Gaussian (joint) distribution is quadratic in the dimensionality (in order to represent its covariance matrix), which can often be manageable for data sets of moderate dimensionality.

8.2.1 Continuous Numeric Data: The Gaussian Distribution

In the case when the feature variables are continuous numeric values in d dimensions, it is natural to use a d -dimensional Gaussian distribution to represent the data. It is also common to use a Gaussian distribution with conditional independence among the attributes. Consider a data set containing the n observations $\vec{x}_1 \dots \vec{x}_n$. The i th data point \vec{x}_i contains d dimensions denoted by $[x_{i1}, x_{i2}, \dots, x_{id}]$. The joint density function of the conditional distribution of the r th mixture component is denoted by $f_{\vec{X}|calG_r}(\vec{x}_i)$, which can be expressed as the product of d marginal density functions (because of the conditional independence assumption). Each of these marginal distributions is a 1-dimensional normal (Gaussian) distribution. Assume that the parameters of the marginal probability density function $f_{X_j|\mathcal{G}_r}(\cdot)$ of the Gaussian distribution for the r th mixture model along the j th dimension are μ_{jr} and

σ_{jr} , respectively. Therefore, the probability density of the i th d -dimensional observation $\vec{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]$ is obtained using the conditional independence assumption as follows:

$$f_{\vec{X}|\mathcal{G}_r}(\vec{x}_i) = \prod_{j=1}^d f_{X_j|\mathcal{G}_r}(x_{ij}) = \prod_{j=1}^d \frac{\exp(-(x_{ij} - \mu_{jr})^2/(2\sigma_{jr}^2))}{\sqrt{2\pi}\sigma_{jr}} \quad (8.2)$$

$$= \frac{\exp(-\sum_{j=1}^d [(x_{ij} - \mu_{jr})^2/(2\sigma_{jr}^2)])}{(2\pi)^{(d/2)} \prod_{j=1}^d \sigma_{jr}} \quad (8.3)$$

The conditional distribution of the r th mixture component is used for a particular training point \vec{x}_i when $y_i = r$. The availability of the class labels y_i allows one to cleanly partition the training data into the different classes corresponding to the k conditional Gaussian distributions. Therefore, each of these distributions can be reconstructed (i.e., their parameters can be learned) using the training points that are specific to those distributions. Furthermore, since the different dimensions are independent of one another, the parameters of each dimension can be estimated independently using the statistics of that dimension (cf. Lemma 6.1 in section 6.3.9 of Chapter 6). Therefore, μ_{jr} is estimated as the sample mean of the j th dimension for training examples belonging to the r th class. Similarly, σ_{jr} is estimated as the sample standard deviation of the j th dimension for training examples belonging to the r th class.

In addition to the distribution parameters, the prior probabilities need to be estimated. The set of n class memberships in the training data can be viewed as samples from a categorical distribution, which is the k -way generalization of the Bernoulli distribution. In fact, for binary classification problems, one needs to estimate the parameters of a Bernoulli distribution. Both the Bernoulli and categorical distributions can be reconstructed in a similar way from training samples (cf. section 6.3.2 of Chapter 6). Specifically, the prior probability $P(\mathcal{G}_r)$ is estimated as the fraction of instances belonging to the r th class, which is the maximum-likelihood estimate for the Bernoulli/categorical distributions.

Problem 8.1 (Gaussian Components without Independence Assumption)
Extend the maximum-likelihood estimation of your Gaussian mixture model to the case where the individual dimensions are not independent of one another. You may use the results in section 6.3.10 of Chapter 6 for distribution reconstruction. Does this mixture model satisfy the naïve assumption?

8.2.1.1 Prediction

After the maximum-likelihood estimation of the parameters has been performed, the posterior probabilities of test instances can be predicted by substituting the Gaussian probability density function (cf. Equation 8.3) in the Bayes rule for posterior probabilities (cf. Equation 8.1). As discussed in Equation 8.1, the posterior probability of out-of-sample instance $\vec{z} = [z_1, \dots, z_d]$ belonging to class r is evaluated as follows:

$$\begin{aligned} P(\mathcal{G}_r | \vec{X} = \vec{z}) &= \frac{P(\mathcal{G}_r) f_{\vec{X}|\mathcal{G}_r}(\vec{z})}{\sum_{s=1}^k P(\mathcal{G}_s) f_{\vec{X}|\mathcal{G}_s}(\vec{z})} \\ &= \frac{P(\mathcal{G}_r) \cdot \left[\exp(-\sum_{j=1}^d [(z_j - \mu_{jr})^2/(2\sigma_{jr}^2)]) / \prod_{j=1}^d \sigma_{jr} \right]}{\sum_{s=1}^k P(\mathcal{G}_s) \cdot \left[\exp(-\sum_{j=1}^d [(z_j - \mu_{js})^2/(2\sigma_{js}^2)]) / \prod_{j=1}^d \sigma_{js} \right]} \end{aligned}$$

It is noteworthy that some constant factors in front of the Gaussian distribution have been omitted from both the numerator and the denominator. Furthermore, the denominator is useful for normalization of the class-specific values into probabilities, but since it is the same normalization factor across classes, it does not change the relative order of the estimated posterior probabilities of the classes. Therefore, if one only wants to report the class with the largest value of the posterior probability, one can ignore the proportionality factor in the denominator as follows:

$$P(\mathcal{G}_r | \vec{X} = \vec{z}) \propto P(\mathcal{G}_r) \cdot \left[\exp\left(-\sum_{j=1}^d [(z_j - \mu_{jr})^2 / (2\sigma_{jr}^2)]\right) / \prod_{j=1}^d \sigma_{jr} \right] \quad (8.4)$$

The class with the largest (proportional) value of the posterior probability can be reported as the relevant one for the test instance.

It is noteworthy that the proportional form of the above relationship is sufficient for comparing different classes but it is not sufficient for comparing different test instances. In some applications, one is given a set of test instances and it is desirable to rank them in order of the propensity to belong to a particular (desirable) class. Such scenarios arise commonly in imbalanced classification or supervised anomaly detection problems.

An example of a setting in which test instances need to be ranked is a target mailing application in which the class labels are “*will buy*” or “*will not buy*. The goal of the e-commerce merchant (mailer) is to identify individuals most likely to buy a particular product, but the volume of mail is constrained to a specific amount. Therefore, the test instances need to be ranked and the top candidates are sent the mail. Since the proportionality factor is the same across classes but not across test instances, one cannot use the proportional relationship when comparing different test instances in terms of their probability of belonging to a particular (desirable) class. In such cases, the proportionality factor for each test instance is separately computed by using the fact that the posterior probabilities of the different classes must sum to 1.

Problem 8.2 (Laplace Model for Bayes Classification) *The Laplace distribution, which is introduced on page 312, is very similar to the Gaussian distribution but uses the normalized L_1 -distance in the exponent. Specifically, a 1-dimensional Laplace distribution is as follows:*

$$f_Z(z) = \frac{1}{2s} \exp\left(-\frac{\|z - \mu\|_1}{s}\right)$$

Like the Gaussian distribution, the Laplace distribution contains a mean parameter μ but a different dispersion parameter s based on the L_1 -distance instead of the variance σ^2 . Rework the details of the Bayes classifier in this section using a d-dimensional Laplacian distribution with conditional independence in lieu of the Gaussian distribution. Provide the details of both the training (i.e., estimating mean and dispersion parameter) and the prediction steps.

8.2.1.2 Handling Overfitting

As discussed in section 6.6 of Chapter 6, robust reconstruction of distributions requires a significant amount of data. In the case of the Bayes model, one is simultaneously reconstructing two types of distributions. The first is the categorical/Bernoulli distribution for modeling prior probabilities, whereas the second one is the Gaussian distribution for each component of the mixture. Such overfitting issues are addressed using maximum a posteriori (MAP) estimation of the parameters of the two types of distributions, wherein an *a priori* assumption is made on the parameters of the two types of distributions. In general, many

modeling techniques in machine learning that reduce overfitting can be thought of in terms of switching from MLE models to MAP models.

For the prior (categorical) distribution, the MAP estimation process results in Laplacian smoothing (cf. section 6.6.2 of Chapter 6). In this case, the assumption is that the categorical distribution parameters (prior parameters) follow a beta distribution. If n_r be the number of training instances belonging to the r th class out of n training instances, then MAP estimation with Laplacian smoothing parameter $\lambda > 0$ yields the following estimate (cf. Equation 6.23 of Chapter 6):

$$P(\mathcal{G}_r) = \frac{n_r + \lambda}{n + k\lambda} \quad (8.5)$$

Similarly, MAP estimation can be used to determine the parameters of the Gaussian distribution. Conceptually, MAP estimation can be thought of in terms of adding perturbations to the data by either adding fake training samples or in adding noise to existing training samples. As discussed in section 6.6.2 of Chapter 6, Laplacian smoothing can be thought of in terms of adding fake samples to the training data. The additional noise in the data usually results in the imposition of some type of structure on the parameters. For example, Laplacian smoothing pushes the parameters of the categorical distribution towards equal probabilities in the absence of evidence to the contrary.

In the following, we discuss the estimation of the Gaussian parameters in a manner that reduces overfitting. Here, we provide a heuristic that conceptually adds noise to the data (although it can be shown to be connected to MAP estimation). For each point belonging to class r , the maximum-likelihood estimate of the Gaussian variance parameter (i.e., sample variance without the Bessel correction) is multiplied by $(1 + \beta/n_r)$ for some value of $\beta > 0$, which is typically slightly greater than 1. This is conceptually equivalent to adding Gaussian noise to the j th attribute of each observation in the r th cluster; this noise has zero mean and variance $\beta\hat{\sigma}_{jr}^2/n$. Choosing a value of $\beta = 1$ brings the estimate closer to the Bessel-corrected value of the variance estimate (i.e., the unbiased estimate), although it will always be slightly less than the under-corrected value.

Example 8.1 You have a training data set of individuals with two attributes (not including the class label) corresponding to age and salary. Each person in this data set is either in a managerial position or not in a managerial position, which is the binary class label. Exactly 10% of the points correspond to managers. You calculate the sample mean and standard deviation on both attributes by class group. You find the following: (i) Managers have mean age 44 and standard deviation of 12. (ii) Managers have mean salary of 100,000 and standard deviation of 10,000. (iii) Non-managers have mean age 39 and standard deviation of 11. (iv) Non-managers have mean salary of 65,000 and standard deviation of 15,000. John is a 50 year old employee with a salary of 80,000. Find the probability that he is a manager assuming that you use a Gaussian Bayes classifier with the independence assumption across attributes.

Solution: Let A be the event that John is a manager and \vec{a} be his attributes in terms of age and salary. The attributes are denoted by the 2-dimensional random variable \vec{X} for which \vec{a} is an instantiation. We want to find the probability $P(A|\vec{X} = \vec{a})$. This

probability can be computed using the Bayes rule as follows:

$$\begin{aligned} P(A|\vec{X} = \vec{a}) &= \frac{P(A)f_{\vec{X}|A}(\vec{a})}{P(A)f_{\vec{X}|A}(\vec{a}) + P(A^c)f_{\vec{X}|A^c}(\vec{a})} \\ &= \frac{0.1 \frac{1}{12 \times 10000} \exp(-0.5^2/2 - (-2)^2/2)}{0.1 \frac{1}{12 \times 10000} \exp(-0.5^2/2 - (-2)^2/2) + 0.9 \frac{1}{11 \times 15000} \exp(-1^2/2 - (1)^2/2)} \end{aligned}$$

The factor $1/\sqrt{2\pi}$ in the Gaussian distribution has already been removed from both the numerator and denominator by cancelation. The values inside the exponents are the negative sum of the squares of John's group-specific Z-values (after dividing by 2). One can simplify the above expression to the following:

$$P(A|\vec{X} = \vec{a}) = \frac{0.1 \times 11 \times 15000 \times \exp(-2.125)}{0.1 \times 11 \times 15000 \times \exp(-2.125) + 0.9 \times 12 \times 10000 \times \exp(-1)}$$

The expression above can be further simplified as follows:

$$P(A|\vec{X} = \vec{a}) = \frac{33 \times \exp(-1.125)}{33 \times \exp(-1.125) + 216} = \frac{10.7135}{226.7135} \approx 0.047$$

Therefore, the probability that John is a manager is 0.047 based on the prediction of the Bayes classifier. ■

Example 8.2 It is known that the maximum-likelihood estimate $\hat{\sigma}_{jr}^2$ of the variance σ_{jr}^2 of attribute j in class r (without adjustments for overfitting) is simply the class-specific sample variance of that attribute without the Bessel correction (which underestimates it). A common heuristic correction is to use the regularized variance estimate $\hat{\sigma}_{jr}^2(1 + \beta/n_r)$ for hyper-parameter $\beta > 0$. This approach is equivalent to performing maximum likelihood estimation on the data set after adding i.i.d. noise to each entry in the data matrix with variance β/n_r . For what value of β does the regularized estimate reduce to the Bessel-corrected sample variance?

Solution: The Bessel correction essential multiplies the uncorrected sample variance by a factor of $n_r/(n_r - 1)$. Note that this correction factor is greater than 1. On the other hand, regularization also multiplies the uncorrected sample variance with the factor $(1 + \beta/n_r)$ (which is also greater than 1). For the two correction factors to be the same, we have the following condition:

$$(1 + \beta/n_r) = n_r/(n_r - 1)$$

On simplifying, it can be shown that $\beta = n_r/(n_r - 1)$, which is very close to 1 (except for small data sets). ■

8.2.2 Binary Data: The Bernoulli Distribution

The Bernoulli distribution is relevant when the data attributes take on the values of 0 or 1. For example, *implicit-feedback data* contains 0-1 attribute-values indicating customer buying or not buying specific items. The Bernoulli distribution is also used for cases in which the data attributes contain sparse nonnegative values that are not too different from one another. In such cases, the distinction between zero and nonzero values is greater than that between the different nonzero values. A specific example of this domain is text, where the dimensionality d of the data is equal to the size of the vocabulary from which the documents are drawn. The documents are represented as a vector of word frequencies — since the frequencies of most words are zero in any given document and the variation in word frequency within any given document (of statistically relevant words other than common parts of speech) varies by a factor of no more than 10, a reasonable approximation is to treat the word frequencies as 0-1 values. Therefore, the approach is particularly suitable for short text documents.

The Bernoulli distribution is useful in domains beyond text. It is possible to transform any data set containing a mixture of categorical and numeric values to a binary data set. First, the numeric values are transformed to categorical values using the process of *discretization*, wherein each attribute domain is divided into ranges and the attribute value of an observation is assigned to the category corresponding to its range. After the mixed-attribute data set has been transformed to a purely categorical data set, each categorical attribute can be transformed to multiple binary attributes via the process of one-hot encoding (cf. section 7.5 in Chapter 7). Note that this process results in an expansion of the number of attributes, since each categorical attribute maps to multiple binary attributes.

As in the case of numeric data, it is assumed that the n training pairs are denoted by $(\vec{x}_1, y_1), \dots, (\vec{x}_i, y_i), \dots, (\vec{x}_n, y_n)$. The i th data point \vec{x}_i contains d dimensions denoted by $[x_{i1}, x_{i2}, \dots, x_{id}]$, all of which are binary values. For example, the set of attributes for a 4-dimensional observation might be $[1, 0, 1, 1]$. The joint probability mass function of the r th mixture component (class) is denoted by $p_{\vec{X}|\mathcal{G}_r}(\vec{x}_i)$, although conditional independence ensures that the joint probability mass function is the product of the marginal probability mass functions over the different dimensions. Each of these probability mass functions corresponds to a Bernoulli distribution for a particular dimension of the r th mixture component (class) distribution. Assume that the parameter (success probability) of the marginal probability mass function $p_{X_j|\mathcal{G}_r}(\cdot)$ of the Bernoulli distribution for the r th mixture model along the j th dimension is p_{jr} . Therefore, the probability of the i th d -dimensional observation $\vec{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]$ is obtained using the independence assumption as follows:

$$p_{\vec{X}|\mathcal{G}_r}(\vec{x}_i) = \prod_{j=1}^d p_{X_j|\mathcal{G}_r}(x_{ij}) = \prod_{j=1}^d [p_{jr}^{x_{ij}} (1 - p_{jr})^{(1-x_{ij})}] \quad (8.6)$$

As in the case of numeric data, the value of y_i is used to decide which mixture component \mathcal{G}_r is relevant for the i th training point \vec{x}_i (i.e., \mathcal{G}_r is relevant when $y_i = r$). Each class-specific Bernoulli distributions can be reconstructed by learning its success parameter with the help of the training points belonging to that class. Furthermore, since the different dimensions are independent of one another in each such joint distribution, the parameters of each dimension can be estimated independently using the statistics of that dimension in the training data (cf. Lemma 6.1 in section 6.3.9 of Chapter 6). Therefore, p_{jr} is estimated as the fraction of training examples in the r th class that take on the value of 1 for the j th dimension (cf. section 6.3.2 of Chapter 6). In other words, if n_r is the number of training

examples belonging to the r th class, then p_{jr} is estimated as follows:

$$\hat{p}_{jr} = \frac{\sum_{i:y_i=r} x_{ij}}{n_r} \quad \forall j, r \quad (8.7)$$

The aforementioned relationship provides the maximum likelihood estimate of the Bernoulli distribution parameters. In addition to the distribution parameters, the prior probabilities need to be estimated. As in the case of Bayes classifier with the Gaussian distribution, the prior probability $P(\mathcal{G}_r)$ is estimated as the fraction of instances belonging to the r th class.

Problem 8.3 (Categorical Distribution) Extend the maximum-likelihood estimation of your mixture model to the case where the individual mixture components are drawn from the categorical distribution. This model is useful for Bayes classification of categorical data sets, without having to go through the intermediate step of one-hot encoding.

Problem 8.4 What kind of mixture distribution would you use to perform Bayes classification of data sets that contain both categorical and numeric attributes?

8.2.2.1 Prediction

After the maximum-likelihood estimation of the parameters has been performed, the posterior probabilities of test instances can be predicted by substituting the Bernoulli probability mass function (cf. Equation 8.6) in the Bayes rule for posterior probabilities (cf. Equation 8.1). As discussed in Equation 8.1, the posterior probability of out-of-sample instance $\vec{z} = [z_1, \dots, z_d]$ belonging to class r is evaluated as follows:

$$P(\mathcal{G}_r | \vec{X} = \vec{z}) = \frac{P(\mathcal{G}_r) f_{\vec{X}|\mathcal{G}_r}(\vec{z})}{\sum_{s=1}^k P(\mathcal{G}_s) f_{\vec{X}|\mathcal{G}_s}(\vec{z})} = \frac{P(\mathcal{G}_r) \cdot \left[\prod_{j=1}^d [p_{jr}^{z_j} (1 - p_{jr})^{(1-z_j)}] \right]}{\sum_{s=1}^k P(\mathcal{G}_s) \cdot \left[\prod_{j=1}^d [p_{js}^{z_j} (1 - p_{js})^{(1-z_j)}] \right]}$$

As in the case of numeric data, the constant factor in the denominator can be ignored when comparing across different classes:

$$P(\mathcal{G}_r | \vec{X} = \vec{z}) \propto P(\mathcal{G}_r) \cdot \left[\prod_{j=1}^d [p_{jr}^{z_j} (1 - p_{jr})^{(1-z_j)}] \right]$$

The class with the largest value of the posterior probability is reported as the relevant one for that test instance.

8.2.2.2 Handling Overfitting

When there is paucity of data, one needs to use MAP estimation in order to make the estimations more robust. There are two distributions for which MAP estimation needs to be performed. The first is the categorical/Bernoulli distribution for modeling prior probabilities, whereas the second one corresponds to the Bernoulli distribution for each component of the mixture. Both types of distributions are very similar and require Laplacian smoothing in the context of MAP estimation (cf. Equation 6.23 of Chapter 6).

The Laplacian smoothing of the prior probabilities has already been discussed on page 360. If n_r be the number of training instances belonging to the r th class out of n

training instances, then MAP estimation with Laplacian smoothing parameter $\lambda > 0$ yields the following estimate:

$$P(\mathcal{G}_r) = \frac{n_r + \lambda}{n + k\lambda} \quad (8.8)$$

The MAP estimation of the distribution-specific parameters is also similar because it is drawn from a Bernoulli distribution. In this case, the Laplacian smoothing parameter is $\beta > 0$, and the corresponding MAP estimation of the Bernoulli parameter p_{jr} for attribute j and mixture component r is as follows:

$$\hat{p}_{jr} = \frac{\beta + \sum_{i:y_i=r} x_{ij}}{2\beta + n_r} \quad (8.9)$$

The main difference from Equation 8.7 is that β has been added to the numerator and 2β has been added to the denominator. This setting corresponds to an equal prior probability of a binary attribute taking on a value of 0 or 1. It is possible to improve the estimation further by observing that the two values of the binary attribute are not equally likely (particularly when such evidence is available from a lot of other attributes). For example, most attributes are sparse and therefore an attribute is more likely to take on a value of 0 rather than a value of 1. Let f be the fraction of 1s out of all $n \cdot d$ binary values in the training data (i.e., all values over all attributes and observations). Then, one can assume that the prior value of the probability p_{jr} is f rather than $1/2$. In such a case, MAP estimation yields the following:

$$\hat{p}_{jr} = \frac{\beta + \sum_{i:y_i=r} x_{ij}}{\beta/f + n_r} \quad (8.10)$$

This approach yields more robust estimates of the Bernoulli probability of each attribute-mixture combination because it accounts for the sparsity of the attributes in the estimation process.

Example 8.3 You have a training data set of individuals included in the US labor force with two binary attributes (not including the class label) corresponding to whether they are college educated and whether they live in a neighborhood with average household salary above \$40,000. Each person in this data set is associated with a binary class label indicating whether they are unemployed. The unemployment rate in the data set is 4.0%. You calculate the education and salary statistics by class group. You find the following: (i) 35% of unemployed people are college educated. (ii) 10% of unemployed people live in a neighborhood with average household salary above \$40,000. (iii) 70% of employed people are college educated. (iv) 80% of employed people live in a neighborhood with average household salary above \$40,000. Tom is not college educated and lives in a neighborhood with average household salary greater than \$40,000. Find the probability that he is unemployed using a Bernoulli distribution on a Bayes classifier.

Solution: Let A be the event that Tom is unemployed and \vec{a} be his binary attributes in terms of education and neighborhood household salary. The attributes are denoted by the 2-dimensional random variable \vec{X} for which \vec{a} is an instantiation. We want to find the probability $P(A|\vec{X} = \vec{a})$. This probability can be computed using the Bayes

rule as follows:

$$\begin{aligned} P(A|\vec{X} = \vec{a}) &= \frac{P(A)p_{\vec{X}|A}(\vec{a})}{P(A)p_{\vec{X}|A}(\vec{a}) + P(A^c)p_{\vec{X}|A^c}(\vec{a})} \\ &= \frac{0.04 \times 0.65 \times 0.1}{0.04 \times 0.65 \times 0.1 + 0.96 \times 0.3 \times 0.8} \end{aligned}$$

One can simplify the above expression to the following:

$$P(A|\vec{X} = \vec{a}) = \frac{13}{13 + 1152} \approx 0.0116$$

Therefore, the probability that Tom is unemployed is 0.0116. Even though Tom is not college educated, the fact that he lives in a neighborhood with high average salary causes his probability of being unemployed to be lower than the general unemployment probability of 0.04. ■

8.2.3 Sparse Numeric Data: The Multinomial Distribution

An important case that is very relevant to text data is that of sparse and numeric data, in which the observations are vector space representations of documents. The dimensionality corresponds to the number of words in the vocabulary, and the attributes contain the frequencies of the individual words in the documents. In such cases, the frequencies of words in the documents can be modeled with the multinomial distribution. The generative process also assumes that the numeric data values are discrete numeric values, although it is possible to extend the model to continuous values. While the Bernoulli distribution is also used to model the frequencies of words in documents are approximated by using binary values to model presence/absence of words, it does not work very well when the underlying documents are very long and repeated occurrences of words become important. In such cases, the binary approximation of word frequencies starts becoming very inaccurate (as it ignores the significant variations in nonzero word frequencies).

In the multinomial model, it is assumed that the frequencies of words in each text document are generated by multiple throws of a biased die, wherein the number of throws is equal to the number of words in the document. The number of faces of the die is equal to the number of words in the vocabulary (i.e., lexicon size), and each throw adds one to the frequency of a word depending on which face of the die shows up. Therefore, multiple throws of the die are required to generate the frequencies of the words in a single document. The biased die that is used is specific to the mixture component at hand. For example, a mixture component (class) corresponding to the “Sports” label will have very different probability values on its faces than a mixture component corresponding to the “Politics” label because the words of the documents in these different classes will have different relative frequencies. The probabilities of the die faces will be biased heavily in favor of sports-related words in the first case, whereas the face probabilities will be heavily biased in favor of sports-related words in the second case.

It is assumed that the n training pairs are denoted by $(\vec{x}_1, y_1) \dots (\vec{x}_i, y_i), \dots (\vec{x}_n, y_n)$. The i th data point \vec{x}_i contains the d -dimensional row vector $[x_{i1}, x_{i2}, \dots, x_{id}]$, all of which are discrete numeric values. The data is typically high dimensional and sparse, corresponding to the fact that most of the values of x_{ij} are zeros. The joint probability mass function of

the r th mixture component (class) is denoted by $p_{\vec{X}|\mathcal{G}_r}(\vec{x}_i)$. The multinomial distribution is one of the few examples of the Bayes classifier in which there is no conditional independence assumption among the attributes. Each of these probability mass functions corresponds to a multinomial distribution, although one of the parameters (number of die throws) is different for different documents. Therefore, we have an additional random variable L , corresponding to the number of die throws. This random variable has a probability mass function $P_L(l)$, although we will see later that the specific form of this mass function does not matter when comparing different classes for the same document. A key assumption is that the length L is independent of the choice of the mixture component (or else the parameter estimation and prediction process in the model will become more complex). The basic assumption here is that all classes contain documents of the same average length.

Aside from the number of die throws, the multinomial distribution is also defined by the parameters corresponding to the probabilities of the different die faces. It is assumed that the probability of the j th die face (i.e., j th word in vocabulary) for the r th mixture model is denoted by p_{jr} . A key point is that the probability of a particular observation \vec{x}_i is obtained by first ensuring that the length $l_i = \sum_{j=1}^d x_{ij}$ of the document is drawn from $p_L(l)$, and then using a multinomial distribution whose length parameter is fixed to l_i . The conditional probability of the i th d -dimensional observation $\vec{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]$ (given the r th mixture component) is obtained by multiplying the probability of the document length l_i with the probability that the observation $[x_{i1}, x_{i2}, \dots, x_{id}]$ is obtained by throwing a die l_i times with face probabilities $p_{i1} \dots p_{id}$:

$$p_{\vec{X}|\mathcal{G}_r}(\vec{x}_i) = p_L(l_i) \cdot p_{[\vec{X}|\mathcal{G}_r, L=l_i]}(\vec{x}_i) \quad (8.11)$$

$$= p_L(l_i) \frac{l_i!}{\prod_{j=1}^d x_{ij}!} \prod_{j=1}^d p_{jr}^{x_{ij}} \quad (8.12)$$

Note that the specific form of the probability mass function $p_L(l_i)$ is not specified, although it is often assumed to be drawn from a Poisson distribution. It turns out that the specific form of the distribution of L does not matter as long as it is independent of the choice of mixture component.

The parameters of each mixture component of the multinomial distribution can be adapted using the approach for the multinomial distribution discussed in section 6.3.5 of Chapter 6. One can adapt Equation 6.10 of section 6.3.5 to this setting by performing the estimation separately for the training examples belonging to the r th mixture component (i.e., training examples with $y_i = r$). Adapting Equation 6.10 to the notations used in this section, the probability parameter p_{jr} may be estimated as follows:

$$\hat{p}_{jr} = \frac{\sum_{i:y_i=r} x_{ij}}{\sum_{i:y_i=r} l_i} \quad (8.13)$$

Note that the estimation is performed separately for the training examples of each mixture component. The parameters of $p_L(l)$ (document length) do not need to be estimated. As we will see in the next section, the expression for the posterior probability does not use the document length. The prior probability $P(\mathcal{G}_r)$ is estimated as the fraction of instances belonging to the r th class.

8.2.3.1 Prediction

After the maximum-likelihood estimation of the parameters has been performed, the posterior probabilities of test instances can be predicted by substituting the multinomial proba-

bility mass function (cf. Equation 8.12) in the Bayes rule for posterior probabilities (cf. Equation 8.1). As discussed in Equation 8.1, the posterior probability of out-of-sample instance $\vec{z} = [z_1, \dots, z_d]$ belonging to class r is evaluated as follows:

$$\begin{aligned} P(\mathcal{G}_r | \vec{X} = \vec{z}) &= \frac{P(\mathcal{G}_r) f_{\vec{X}|\mathcal{G}_r}(\vec{z})}{\sum_{s=1}^k P(\mathcal{G}_s) f_{\vec{X}|\mathcal{G}_s}(\vec{z})} = \frac{P(\mathcal{G}_r) \cdot p_L(\sum_j z_j) \left[\frac{l_i!}{\prod_{j=1}^d z_j!} \prod_{j=1}^d p_{jr}^{z_j} \right]}{\sum_{s=1}^k P(\mathcal{G}_s) \cdot p_L(\sum_j z_j) \left[\frac{l_i!}{\prod_{j=1}^d z_j!} \prod_{j=1}^d p_{js}^{z_j} \right]} \\ &= \frac{P(\mathcal{G}_r) \cdot \left[\prod_{j=1}^d p_{jr}^{z_j} \right]}{\sum_{s=1}^k P(\mathcal{G}_s) \cdot \left[\prod_{j=1}^d p_{js}^{z_j} \right]} \end{aligned}$$

Note that all the factors involving the length of the document have disappeared from the above expression, which is why we do not need to know the specific form of the length distribution $p_L(l)$. Furthermore, the constant factor in the denominator can be ignored when comparing across different classes:

$$P(\mathcal{G}_r | \vec{X} = \vec{z}) \propto P(\mathcal{G}_r) \cdot \left[\prod_{j=1}^d p_{jr}^{z_j} \right]$$

The class with the largest value of the posterior probability is reported as the relevant one for that test instance.

8.2.3.2 Handling Overfitting

When there is paucity of data, one needs to use maximum a posteriori (MAP) estimation in order to make the estimations more robust. There are two distributions for which the MAP estimation needs to be performed. The first is the categorical/Bernoulli distribution for modeling prior probabilities, whereas the second one corresponds to the multinomial distribution for each component of the mixture. Both types of distributions are very similar and require Laplacian smoothing in the context of maximum MAP estimation.

The Laplacian smoothing of the prior probabilities has already been discussed in page 360. If n_r be the number of training instances belonging to the r th class out of n training instances, then MAP estimation with Laplacian smoothing parameter $\lambda > 0$ yields the following estimate:

$$P(\mathcal{G}_r) = \frac{n_r + \lambda}{n + k\lambda} \quad (8.14)$$

The MAP estimation of the multinomial distribution is performed with the Laplacian smoothing parameter $\beta > 0$, and the corresponding MAP estimation of the multinomial parameter p_{jr} for attribute j and mixture component r is as follows:

$$\hat{p}_{jr} = \frac{\beta + \sum_{i:y_i=r} x_{ij}}{d\beta + \sum_{i:y_i=r} l_i} \quad (8.15)$$

The basic assumption in the MAP estimation of the die0face probabilities is that each of the d faces of the die has equal prior probability.

8.2.3.3 Extending Multinomial Distributions to Real-Valued Data

The aforementioned discussion assumes that the numeric data contains only nonnegative integers. Since the generative process is based on throws of the die, it is natural to have only integer outcomes. However, it is possible to *interpolate* the generative process to real-valued features, although the resulting features no longer have a generative interpretation. First, note that the integer nature of the attributes is emphasized by the fact that all the intermediate expressions in estimation use the factorial function $x!$, which is naturally defined for integers. However, the posterior probabilities do not actually use the factorial function, and can be evaluated for real-valued frequencies.

In some unusual cases, it may be desired to compute the generative probabilities of individual data points, which does require the computation of the factorial function. In such cases, one can interpolate the factorials of real-valued quantities with the use of the *gamma* function:

$$\Gamma(x + 1) = \int_{y=0}^{\infty} y^x \exp(-y) dy$$

Note that the gamma function can be computed for any real-valued x , although it satisfies the following relationship for integer values of x :

$$\Gamma(x + 1) = x!$$

Therefore, in order to interpolate a factorial to real values, one simply needs to replace it with the gamma function.

8.2.4 Plate Diagrams for Generative Processes

The Bayes classifier assumes that the data set is constructed using a generative process — the principle behind such a process is that it is assumed that *a sequence of probabilistic steps are used to generate the observed data*. For example, in the generative process for the training data of the Bayes classifier, the identifier of a mixture component is first generated from a categorical distribution with k possible values. Subsequently, an observation \vec{x}_i is generated from the conditional distribution corresponding to the selected mixture component. A plate diagram shows precisely this sequence of steps in graphical notation. The advantage of using a plate diagram over a verbal description is that it provides a more intuitive way to represent generative processes and also compare processes with similar probabilistic procedures.

Each generative step of the instantiations of a random variable is indicated in a plate diagram by a round circle containing the variable. A sequence of generation of variables will correspond to multiple such circles, which are connected by arrows indicating the sequential order of generation and the logical flow of the process. For example, the training data for classification is generated by first generating the class identifier r using the prior probabilities $\alpha_1 \dots \alpha_k$ (where $\alpha_s = P(\mathcal{G}_s)$). Subsequently, this class identifier r is used to generate the d -dimensional observation \vec{x}_i using the joint distribution (e.g., Gaussian distribution) for mixture component r . This process is shown in Figure 8.2(a). Note the arrow showing the sequence of steps. Furthermore, since the process is repeated to obtain multiple training points, it is encased in a square plate with the right corner indicating the number of times that independent samples are drawn from this generative process. It is also noteworthy that both circles in Figure 8.2(a) are shaded in the case of the training data. A shaded circle indicates a random variable whose outcome is observable. Since both the feature variables and class variable are *visible* to the analyst in the case of the training data, both the circles for the mixture component outcome and the feature generation are shaded.

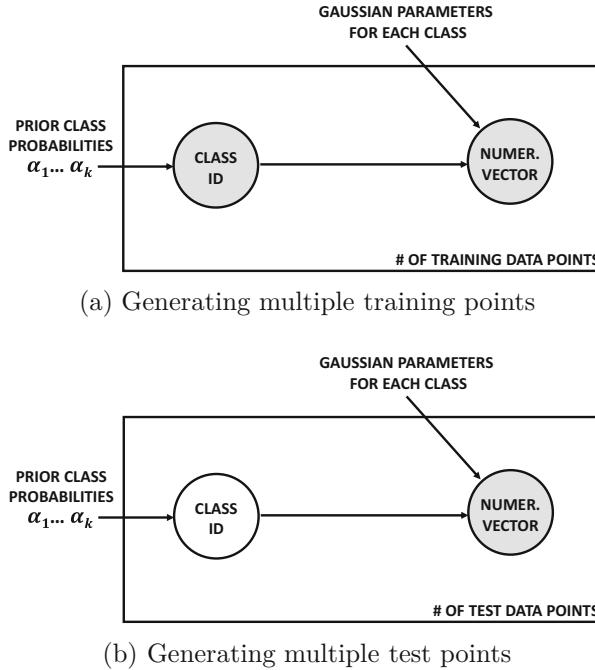


Figure 8.2: The generative process for training and test data in terms of plate diagrams

It is also instructive to examine the case of test data, which has an identical generative process but the class labels are not visible. The corresponding plate diagram is shown in Figure 8.2(b). Note that this plate diagram is similar to that of the training data but the circle for the class identifier is not shaded. This is because the class labels are not visible for test instances. It is noteworthy that the plate diagram for the generation of instances in unsupervised clustering is identical to that of the generation of test instances in classification. The main difference is that the first step of the process generates a cluster identifier rather than a class identifier because mixture components are associated with cluster identifiers in that case. The corresponding plate diagram for clustering is shown in Figure 8.3. The expectation-maximization algorithm (cf. section 6.4 of Chapter 6) is based on this generative process.

The reason that plate diagrams are useful is that they can be used to provide a schematic of a sequence of probabilistic steps as well as the specific number of observations generated by each step. The observations from all compound distributions (such as mixture distributions) are created using a sequence of probabilistic steps. Therefore, plate diagrams are very useful for generative models. The basic components of plate diagrams are shown in Figure 8.4. Each node in a plate diagram contains a variable (or vector of variables), and the presence of the node indicates the generation of an instance of that variable. An edge from a node containing variable A to a node containing the variable B indicates that the generation of variable B is conditional on the variable A . For example, in mixture modeling, the generation of \vec{x}_i is conditional on the specific mixture component that was selected. Therefore, an edge between two nodes shows the probabilistic dependency between two nodes. The shading of a node indicates whether or not the variable generated by that node is visible. For example, the class label is visible during the generation of the training data

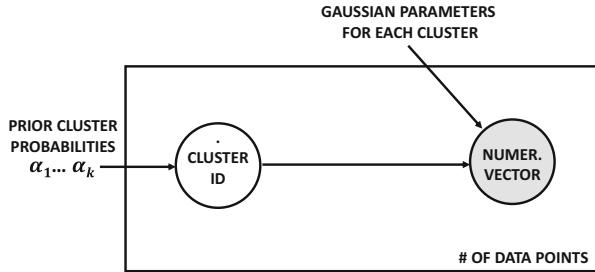


Figure 8.3: The generative process for unsupervised clustering

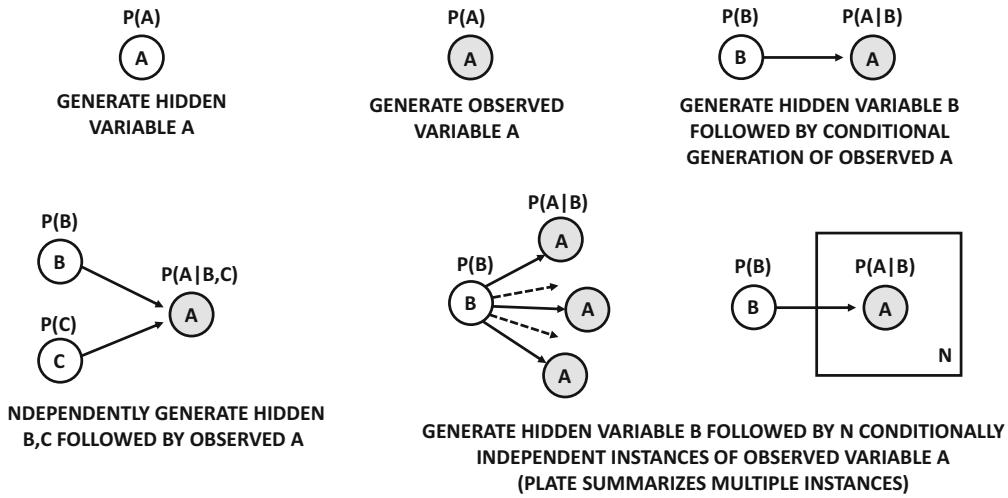


Figure 8.4: The basic scheme for plate diagrams

but not during the generation of the test data. This difference shows up as a difference in shading between Figures 8.2(a) and (b). Furthermore, when the same generative process is repeated multiple times to create many observations, it is shown with a plate [see lower right illustration of Figure 8.4] with the number of instantiations of the variable indicated within the plate.

8.3 Loss-Based Formulations: A Probabilistic View

The generative models of the previous section do not directly incorporate the class-labels into a loss based formulation, but do so indirectly by learning the parameters of the underlying probability distributions (via the loss functions of maximum likelihood estimation behind the scenes). For example, a Bayes classifier with the Gaussian mixture assumption estimates the means and variances of the Gaussian distribution using sample means and variances; this type of estimation is based on results known from maximum likelihood estimation of Gaussian distributions. However, the parameters of each class are estimated separately, and the class label does not appear in the loss-based formulation.

In loss-based formulations for classification, the observed class labels are compared with the predicted class labels directly in order to create loss functions. Each classification prob-

lem, therefore, requires one to directly solve this optimization model that directly incorporates the class labels. The simplest such formulation is that of the extension of the least-squares regression model to classification. This model can be refined further with the use of techniques that are tailored towards the categorical nature of the underlying class labels.

All the models that we will be looking at in this section are referred to as *generalized linear models*, and most of the models are binary classifiers in which the class labels are drawn from $\{-1, +1\}$. Given the training instances (\vec{x}_1, y_1) , (\vec{x}_2, y_2) , \dots , (\vec{x}_n, y_n) , the *predictive model* of y_i from \vec{x}_i is as follows:

$$\hat{y}_i = \text{sign}\{\vec{w} \cdot \vec{x}_i^T\} \quad (8.16)$$

Note the circumflex over \hat{y}_i to indicate that it is a predicted value. It is noteworthy that an explicit bias term is missing from the above formulation, which implies that the feature engineering trick for incorporating the bias variable has been used. In particular, a single feature with a fixed value of 1 is always included as one of the features of each \vec{x}_i and its coefficient in \vec{w} provides the bias parameter.

The weight vector \vec{w} needs to be learned in a data-driven manner in order to maximize the consistency between the sign of $\vec{w} \cdot \vec{x}_i^T$ and y_i on the training data. This consistency is imposed with the use of a loss function. One can interpret the process of learning the weight vector \vec{w} as that of finding a linear separator between the multidimensional representation of the points. For example, consider the case where the vector \vec{x} contains 2-dimensional features along with the bias variable (resulting in a total of three features). In such a case, one can consider \vec{w} to contain the coefficients of a linearly separating hyperplane between the two classes. Points satisfying $\vec{w} \cdot \vec{x}_i^T = 0$ will usually be rare and correspond to those points lying on the hyperplane. Such points represent instances that satisfy characteristics of both classes, and are therefore hard to classify. Points satisfying $\vec{w} \cdot \vec{x}_i^T < 0$ are the "negatively" classified points on one side of the hyperplane, whereas points satisfying $\vec{w} \cdot \vec{x}_i^T > 0$ are the positively classified points. For a particular training data set, is it possible to find a linear separator vector \vec{w} , so that all points with a label of $+1$ lie on the positive side of the hyperplane, and all points with label of -1 lie on the negative side of the hyperplane? The answer to this question depends on the complexity of the data set at hand. For some data sets, such as the one shown on the left-hand side of Figure 8.5, this is indeed possible, because the two classes are *linearly separable*. On the other hand, it is not possible to find such a separator for the data set on the right-hand side of Figure 8.5. Since this section discusses classification models based on linear separators, these models work effectively only for certain restricted types of data sets that are similar to the data set on the left-hand side of Figure 8.5. However, the models can be extended to general settings by using kernel methods and feature transformations.

The prediction of Equation 8.16 uses a sign function. In practice, since the sign function is not differentiable, some differentiable function of y_i and $\vec{w} \cdot \vec{x}_i^T$ needs to be used to enforce "closeness" between these two quantities. This surrogate is essentially a loss function, which usually has a probabilistic interpretation. The fact that y_i is drawn from $\{-1, +1\}$ does have consequences in terms of how this closeness is measured. For example, the simplest approach would be to use the regression loss function directly for classification. This approach is referred to as *least-squares classification*. As we will see later, making this type of assumption is often not realistic because the probabilistic assumptions on the error (i.e., Gaussian error) do not naturally apply to binary class labels. This mismatch shows up as a weakness in the classification model.

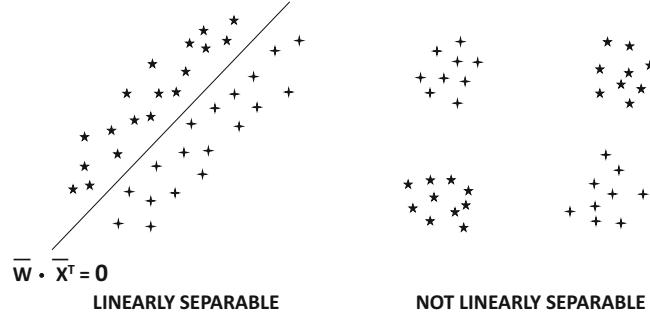


Figure 8.5: Examples of classes that are linearly separable or not separable. One can assume that the points marked ‘+’ correspond to the class label +1, whereas the points marked ‘*’ correspond to the class label −1.

8.3.1 Least-Squares Classification

The least-squares classification model is a direct extension of the least-squares regression model to classification. As in the case of regression, the loss function of least-squares classification is as follows:

$$\text{Minimize}_{\vec{w}} J = \sum_{i=1}^n (y_i - \vec{w} \cdot \vec{x}_i^T)^2$$

In this case, one is trying to minimize the squared distance between y_i and $\vec{w} \cdot \vec{x}_i^T$ by simply treating y_i as a real value, as in regression. Of course, we know that y_i is not a real value, but is drawn from $\{-1, +1\}$. One can also express the above objective function with the use of the $n \times d$ matrix D whose rows are $\vec{x}_1 \dots vecx_n$ and the n -dimensional class vector \vec{y} :

$$\text{Minimize}_{\vec{w}} J = \|D\vec{w} - \vec{y}\|^2$$

In such a case, one can use the solution methodology of least-squares regression in order to learn the value of the weight vector \vec{w} . We express this gradient in matrix calculus notation in the denominator layout:

$$\frac{\partial J}{\partial \vec{w}} = 2 D^T (D\vec{w} - \vec{y}) = \vec{0}$$

Rearranging the optimality condition, one obtains the following:

$$(D^T D)\vec{w} = D^T \vec{y}$$

For now, let us assume that the matrix $D^T D$ is invertible. In such a case, the solution for \vec{w} is as follows:

$$\vec{w} = (D^T D)^{-1} D^T \vec{y}$$

Given this solution vector \vec{w} , the out-of-sample test instance \vec{z} can simply be predicted into its outcome as $\text{sign}\{\vec{w} \cdot \vec{z}^T\}$. This solution is identical to the closed-form solution of least-squares regression discussed in section 7.4.1 of Chapter 7, except that the sign function is used for the final predictive step.

Example 8.4 Consider a classification problem with a single non-trivial predictor variable in which the feature-class pairs are as follows:

$$\{(4, 1), (3, 1), (2, 1), (1, 1), (-1, -1), (-2, -1), (-3, -1), (-4, -1), (1, 1), (-1, 1)\}$$

Note that the class variable tend to be positive when the feature variable is positive and vice versa, although the final training point is an exception. Furthermore, the training point $(1, 1)$ is repeated. Find the optimal least-squares classification in the form $y = \text{sign}\{mx + b\}$, where x is the single non-trivial predictor. Technically, the problem has two predictors (including the engineered predictor) although only one of them is non-trivial and is a part of the data.

Solution: The above data is presented in pairs (x_i, y_i) including the class. The data does not include the feature-engineered variable for bias. We compute the 10×2 independent variable data matrix D whose rows contain the independent variable row vectors (including the engineered variable value of 1 for the bias). The rows of the 10×2 matrix D are the 2-dimensional tuples $\vec{x}_i = [x_i, 1]$ of the set below in that specific order:

$$\{[4, 1], [3, 1], [2, 1], [1, 1], [-1, 1], [-2, 1], [-3, 1], [-4, 1], [1, 1], [-1, 1]\}$$

One can show that the matrix $D^T D$ and its inverse are as follows:

$$D^T D = \begin{bmatrix} 62 & 0 \\ 0 & 10 \end{bmatrix} \quad (D^T D)^{-1} = \begin{bmatrix} 1/62 & 0 \\ 0 & 1/10 \end{bmatrix}$$

The dependent variable vector $\vec{y} = [1, 1, 1, 1, -1, -1, -1, -1, 1, 1]^T$ is used to compute $D^T \vec{y}$:

$$D^T \vec{y} = \begin{bmatrix} 20 \\ 2 \end{bmatrix}$$

One can then compute the weight vector as follows:

$$\vec{w} = (D^T D)^{-1} D^T \vec{y} = \begin{bmatrix} 10/31 \\ 1/5 \end{bmatrix}$$

One can then use this weight vector to define the optimal equation of least-squares classification as follows:

$$y_i = \text{sign} \left\{ \frac{10}{31} x_i + \frac{1}{5} \right\}$$

The final training point is classified incorrectly by this approach (because it is most likely a noisy point). The reader is encouraged to examine the similarity of this solution to Example 7.5 of the previous chapter in order to explore the similarity of least-squares classification with regression. ■

The above solution can have overfitting problems when the number of rows of D is small. In such cases, it is possible for $D^T D$ to not be invertible because its rank is bounded above by the number of rows in D . As in least-squares regression, one can use a regularization

term in order to modify the objective function as follows:

$$\text{Minimize}_{\vec{w}} J = \frac{1}{2} \|D\vec{w} - \vec{y}\|^2 + \frac{\lambda}{2} \|\vec{w}\|^2$$

Here, $\lambda > 0$ is the regularization parameter. The corresponding closed-form solution of least-squares classification is as follows:

$$\vec{w} = (D^T D + \lambda I)^{-1} D^T \vec{y}$$

One can also find \vec{w} with the use of gradient descent or stochastic gradient descent. The approach is identical to the gradient descent methods discussed in section 7.4.3 of Chapter 7. Let S contain the indices (numbered between 1 and n) of a subset of training instances. Then, mini-batch stochastic gradient descent is expressed as follows (see derivations in section 7.4.3 for least-squares regression):

$$\vec{w} \leftarrow \vec{w}(1 - \alpha\lambda) - \alpha \sum_{i \in S} \vec{x}_i^T \underbrace{(\vec{w} \cdot \vec{x}_i^T - y_i)}_{\text{Error } i}$$

Here, $\alpha > 0$ is the learning rate and the set S contains the indices of the randomly selected training instances in the current mini-batch. When the set S contains all training instances, the solution corresponds to gradient descent. On the other hand, when the set S contains exactly one training instance, the solution corresponds to pure stochastic gradient descent. In practice, the set S typically contains a few hundred points. As in the case of least-squares regression, the magnitude and directions of the updates are detected by the magnitude and direction of the error [in addition to the feature values of the training point(s) at hand]. The following example is a variation from Example 7.5 in the chapter on regression.

Example 8.5 Consider the single non-trivial predictor problem of Example 8.4 in which the independent-dependent pairs are as follows:

$$\{(4, 1), (3, 1), (2, 1), (1, 1), (-1, -1), (-2, -1), (-3, -1), (-4, -1), (1, 1), (-1, 1)\}$$

We know from the solution to Exercise 8.4 that the optimal solution is $w_1 = 10/31$, $w_2 = 1/5$. Here, w_2 is the bias coefficient of the engineered feature value of 1. Consider the situation, where the initial weight vector is $[3, -1]^T$. Show that the gradient descent step moves the weight vector closer to the optimal solution for appropriate choice of step size?

Solution: The pairs above are denoted by (x_i, y_i) . The data points $\vec{x}_i = [x_i, 1]$ along with the engineered variable value of 1 (corresponding to the bias coefficient) can be expressed in the following set S :

$$S = \{[4, 1], [3, 1], [2, 1], [1, 1], [-1, 1], [-2, 1], [-3, 1], [-4, 1], [1, 1], [-1, 1]\}$$

Gradient descent requires the computation of error on each data point by current solution $\vec{w} = [3, -1]^T$. On plugging in the above data pairs into the error equation $e_i = \vec{w} \cdot \vec{x}_i^T - y_i = (3x_i - 1 - y_i)$, one obtains the following list of errors:

$$10, 7, 4, 1, -3, -6, -9, -12, 1, -5$$

On computing $\sum_i e_i \vec{x}_i$ one obtains the following:

$$\sum_{\vec{x}_i \in S} e_i \vec{x}_i = [166, -12] \propto [83, -6]$$

Here, the summation is over the engineered features \vec{x}_i in set S . Therefore, the gradient descent update will be as follows:

$$\vec{w} = [3, -1]^T - \alpha [83, -6]^T$$

For small step sizes α , both components of \vec{w} move in the correct direction since w_1 is decreased and w_2 is increased from the initial solution. This would be step in the right direction because the target solution is $[10/31, 1/5]$. For example, choosing $\alpha = 0.035$ results in a new weight vector of $[0.095, -0.79]$, which is closer to the target solution for both components of the vector \vec{w} . Choosing the step to be too small might cause slow progress, whereas choosing very large step sizes will cause the solution to overshoot and oscillate about the optimum. In some cases, choosing large step sizes could even lead to divergence from the optimal solution. A key strategy that is sometimes used in such cases is *line search* [6]. ■

8.3.1.1 The Probabilistic Interpretation and Its Problems

The key probabilistic assumption is that each residual $\epsilon_i = y_i - \vec{w} \cdot \vec{x}_i$ is drawn from a normal distribution defined by random variable \mathcal{E} with zero mean and variance σ^2 . If \vec{X} and Y be the predictor and outcome random variables respectively, we have:

$$\mathcal{E} = Y - \vec{w} \cdot \vec{X}^T \sim \mathcal{N}(0, \sigma^2)$$

Therefore, we have the following:

$$\begin{aligned} f_{\mathcal{E}}(\epsilon_i) &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\epsilon_i^2}{2\sigma^2}\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \vec{w} \cdot \vec{x}_i^T)^2}{2\sigma^2}\right) \end{aligned}$$

Note that this distribution is exactly the same as that of least-squares regression with numerical outcome variables. In such a case, interpreting the error ϵ_i to belong to a Gaussian distribution is problematic, because the value y_i is either 1 or -1 . Therefore, if multiple observations containing very similar values of \vec{x}_i are collected, the errors for these observations will get tightly clustered into two groups depending on whether y_i is $+1$ and -1 — this type of tightly clustered error distribution cannot realistically be considered to be a Gaussian distribution. Therefore, the probabilistic interpretation of least-squares classification is not realistic, which sometimes shows up in the errors in prediction.

8.3.1.2 Practical Issues with Least Squares Classification

The aforementioned problems with probabilistic modeling reflect in qualitative problems in classifying new test instances because the loss function does not reflect the goals of the application at hand. One issue here is that the prediction for a new test instance is

performed as $\hat{y}_{test} = \text{sign}\{\vec{w} \cdot \vec{z}\}$ to (accurately) reflect the binary nature of the prediction, whereas the prediction $\hat{y}_i \approx \vec{w} \cdot \vec{x}_i^T$ for training instances does not use the sign function. This continuous-valued prediction is used to set up the error $\epsilon_i = y_i - \hat{y}_i$, which is modeled as a Gaussian distribution. This mismatch between the treatment of training and test prediction shows up in a practical sense, because the learned weight vector \vec{w} does not reflect the goals of the application at hand.

For example, consider a training instance (\vec{x}_i, y_i) in which the value of y_i is +1. Now imagine a situation in which $\vec{w} \cdot \vec{x}_i^T > 1001$ for 1% of the positive training instances (i.e., $y_i = 1$) and $\vec{w} \cdot \vec{x}_i^T$ exactly matches y_i for other training instances. This is a perfectly accurate model if the training instances were to be classified with the sign function (like test instances) — in fact, the large magnitudes of the training predictions can be interpreted as confidences in the observed values (and the predictions are correct as well). However, the way in which the loss function is set up, the absolute training error for each of the 1% of training instances with $\vec{w} \cdot \vec{x}_i^T > 1001$ is at least 10^6 , which will dominate the learning process of \vec{w} at the expense of other instances. As a result, a computational algorithm such as gradient descent will always try to rapidly change the value of \vec{w} to reduce the error on the 1% of training instances with large error. This will eventually cause classification error for other positive training instances.

Now let us consider a different scenario in which the absolute magnitude of each prediction \hat{y}_i is less than 1, but every single training instance is classified with the incorrect sign. This is a terrible situation from the perspective of prediction of test instances (with the sign function), but the absolute training error per instance is less than 2; this value is much less than the error of the previous case, where all training instances were predicted correctly by sign. Therefore, the loss in this case of fully incorrect predictions will be less than the loss of the case when all predictions are correct. This type of problem ultimately arises because the goal of predicting test instances to binary values does not align with treating the training error as Gaussian random variables. These weaknesses of least-squares classification prompted many changes to the basic regression model. Such models are referred to as generalized linear models because they generalize the linear models of regression to other types of dependent variables.

Example 8.6 Discuss the probabilistic interpretation of regularization in least-squares classification, drawing parallels to the probabilistic interpretation of regularization in regression. The probabilistic interpretation of regularization in regression is discussed in section 7.7 of the previous chapter.

Solution: The probabilistic interpretation of regularization is to add the following prior on the weight vector \vec{w} :

$$f_{\vec{W}}(\vec{w}) = \left(\frac{1}{2\pi\sigma_0^2} \right)^{d/2} \exp\left(-\frac{\|\vec{w}\|^2}{2\sigma_0^2}\right)$$

Here, σ_0 is a hyper-parameter, which is often expressed indirectly in terms of the regularization parameter $\lambda \propto 1/\sigma_0^2$. According to Equation 6.20 of Chapter 6, incorporating such a prior will result in adding the negative logarithm of the prior to the objective function, which is $\lambda\|\vec{w}\|^2/2$ (ignoring constant terms). This is exactly the same probabilistic interpretation as that used for regularization in linear regression. ■

Problem 8.5 Generalize the L_1 -loss regression discussed in the previous chapter (for regression with numerical outcome variables) to the classification setting with labels drawn from $\{-1, +1\}$. You can ignore regularization. Provide a maximum-likelihood interpretation of this model. Discuss the problems associated with this probabilistic interpretation, drawing parallels with least-squares classification.

8.3.2 Logistic Regression

Logistic regression is a probabilistic method that is superficially similar to least-squares classification in terms of using a d -dimensional coefficient (column) vector \vec{w} to classify a d -dimensional test point \vec{z} (expressed as a row vector) to the sign of the dot product $\vec{w} \cdot \vec{z}^T$. However, the way in which the probabilistic modeling is done in order to compute \vec{w} is quite different.

Consider a training data set with n training pairs $(\vec{x}_1, y_1) \dots (\vec{x}_n, y_n)$, so that each y_i is a binary class label drawn from $\{-1, +1\}$. The least-squares classification model assumes that $y_i - \vec{w} \cdot \vec{x}_i^T$ is distributed with Gaussian errors. One can alternatively say that the random variable Y_i (for which the instantiation is y_i) is distributed in Gaussian fashion with mean $\vec{w} \cdot \vec{x}_i^T$ and variance σ^2 . In other words, the probabilistic assumption of the least-squares classification model may be expressed as follows:

$$Y_i \sim \mathcal{N}(\vec{w} \cdot \vec{x}_i^T, \sigma^2) \quad [\text{Least-squares classification}]$$

As discussed earlier in this chapter, there is a problem with using the assumption of a Gaussian distribution in a discrete setting where each Y_i is drawn from $\{-1, +1\}$. Gaussian distributions do not produce discrete values, and the chance of producing an integer like -1 or $+1$ is infinitesimally small. Many of the practical weaknesses of least-squares classification arise from this flawed assumption because the observed data does not fit the assumed distribution very well.

Logistic regression makes a more reasonable assumption by assuming that Y_i is drawn from a discrete Bernoulli distribution. However, Bernoulli distributions are drawn on $\{0, 1\}$, whereas y_i is drawn from $\{-1, +1\}$. We refer to this modified Bernoulli distribution as the M-Bernoulli distribution. The M-Bernoulli distribution returns $+1$ in case of a success and -1 in case of a failure.

As in the case of least-squares classification, the parameters of the distribution from which Y_i are drawn is defined by $\vec{w} \cdot \vec{x}_i^T$. however, since a modified Bernoulli distribution is being used in this case, we need to map $\vec{w} \cdot \vec{x}_i^T$ to a probability value that serves as the parameter of the distribution. In this respect, the *sigmoid function* is very useful:

$$\text{Sigmoid}(\vec{w} \cdot \vec{x}_i^T) = \frac{1}{1 + \exp(-\vec{w} \cdot \vec{x}_i^T)}$$

Note that the sigmoid function of $\vec{w} \cdot \vec{x}_i^T$ yields a value close to 1 when the argument is highly positive and a value close to 0 when the argument is highly negative. This sigmoid function yields the success parameter of the Bernoulli distribution for $y_i^{(0/1)}$. In other words, the probabilistic assumption of logistic regression is as follows:

$$Y_i \sim \text{M-Bernoulli}(\text{Sigmoid}(\vec{w} \cdot \vec{x}_i^T)) \quad [\text{Logistic regression}]$$

This model is more realistic than least-squares classification because it uses a discrete probability distribution like the Bernoulli distribution to model y_i rather than an unrealistic

continuous distribution like the Gaussian distribution. This model sets the stage for the maximum-likelihood estimation of the parameters in the coefficient vector \vec{w} . The loss function for logistic regression is derived from this principle.

8.3.2.1 Maximum Likelihood Estimation for Logistic Regression

The maximum likelihood estimation in logistic regression computes the product of the probabilities of each class label y_i based on the modified Bernoulli distribution model. Note that if y_i is 1, the probability value of this class label is $\text{Sigmoid}(\vec{w} \cdot \vec{x}_i^T)$ (since $y_i = 1$ is a success event). On the other hand, if $y_i = -1$, the probability value of the class label is $1 - \text{Sigmoid}(\vec{w} \cdot \vec{x}_i^T)$. In discriminative models, it is these conditional probabilities that are treated as likelihoods:

$$\text{Likelihood}(\vec{x}_i, y_i, \vec{w}) = \begin{cases} \text{Sigmoid}(\vec{w} \cdot \vec{x}_i^T) & \text{if } y_i = 1 \\ 1 - \text{Sigmoid}(\vec{w} \cdot \vec{x}_i^T) & \text{if } y_i = -1 \end{cases}$$

The aforementioned case-wise statement can be rewritten in a somewhat simpler form by making the following observation about the sigmoid function:

$$\text{Sigmoid}(-v) = \frac{1}{1 + \exp(v)} = 1 - \frac{1}{1 + \exp(-v)} = 1 - \text{Sigmoid}(v)$$

This property allows the likelihood to be rewritten in a consolidated form:

$$\text{Likelihood}(\vec{x}_i, y_i, \vec{w}) = \begin{cases} \text{Sigmoid}(\vec{w} \cdot \vec{x}_i^T) & \text{if } y_i = 1 \\ \text{Sigmoid}(-\vec{w} \cdot \vec{x}_i^T) & \text{if } y_i = -1 \end{cases} = \text{Sigmoid}(y_i \vec{w} \cdot \vec{x}_i^T) \quad (8.17)$$

The products of the likelihoods for the individual training points can be used to create a likelihood fit $\mathcal{L}(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n, \vec{w})$ of logistic regression over the entire training data set, which is as follows:

$$\mathcal{L}(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n, \vec{w}) = \prod_{i=1}^n \text{Sigmoid}(y_i \vec{w} \cdot \vec{x}_i^T)$$

As in all cases of maximum-likelihood estimation, it is numerically convenient to minimize the negative log-likelihood function $\mathcal{LL}(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n, \vec{w})$, which is the negative logarithm of the likelihood fit, and is expressed as follows:

$$\mathcal{LL}(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n, \vec{w}) = - \sum_{i=1}^n \ln (\text{Sigmoid}(y_i \vec{w} \cdot \vec{x}_i^T)) = \sum_{i=1}^n \ln (1 + \exp(-y_i \vec{w} \cdot \vec{x}_i^T))$$

This yields the loss function of logistic regression, which needs to be minimized with respect to \vec{w} :

$$\text{Minimize}_{\vec{w}} J = \sum_{i=1}^n \ln (1 + \exp(-y_i \vec{w} \cdot \vec{x}_i^T))$$

As in the case of least-squares classification, one can avoid overfitting by adding L_2 -regularization. The regularized objective function may be written as follows:

$$\text{Minimize}_{\vec{w}} J = \sum_{i=1}^n \ln (1 + \exp(-y_i \vec{w} \cdot \vec{x}_i^T)) + \frac{\lambda}{2} \|\vec{w}\|^2$$

Note that the probabilistic interpretation of regularization is exactly the same as in the case of least-squares regression (discussed in section 7.7 of Chapter 7). In essence, regularization adds a Gaussian prior to the distribution of the coefficient vector \vec{w} .

Example 8.7 Show that the L_2 -regularized version of logistic regression is derived using MAP estimation on \vec{w} , where a Gaussian prior is imposed on \vec{w} .

Solution: The proof of the above result is similar to that of the MAP estimation result discussed in section 7.7 of Chapter 7. As discussed in that section, the Gaussian prior for L_2 -regularization is as follows:

$$f_{\vec{W}}(\vec{w}) = \left(\frac{1}{2\pi\sigma_0^2} \right)^{d/2} \exp\left(-\frac{\|\vec{w}\|^2}{2\sigma_0^2}\right)$$

Here, σ_0 is a hyper-parameter selected by the analyst that controls the degree of regularization. The negative log-likelihood function $\mathcal{LL}(\cdot)$ has already been derived in this section. Based on the relationship between the negative log-likelihood function and the negative log-posterior function (cf. Equation 6.20 of Chapter 6), the following holds:

$$\mathcal{LP}(\vec{x}_1 \dots \vec{x}_n, y_1, \dots, y_n, \vec{w}) = -\ln(f_{\vec{W}}(\vec{w})) + \mathcal{LL}(\vec{x}_1 \dots \vec{x}_n, y_1, \dots, y_n, \vec{w}) + C_0$$

On plugging in the prior density function as well as the negative log-likelihood function, the following is obtained:

$$\mathcal{LP}(\vec{x}_1 \dots \vec{x}_n, y_1, \dots, y_n, \vec{w}) = \frac{\|\vec{w}\|^2}{2\sigma_0^2} + \sum_{i=1}^n \ln(1 + \exp(-y_i \vec{w} \cdot \vec{x}_i^T)) + C_1$$

Note that the constant C_0 has been changed to C_1 to absorb additional constant terms from the prior. On ignoring the constant term for the purpose of optimization and setting the regularization parameter λ to $1/\sigma_0^2$, the following objective function is obtained:

$$J = \sum_{i=1}^n \ln(1 + \exp(-y_i \vec{w} \cdot \vec{x}_i^T)) + \frac{\lambda \|\vec{w}\|^2}{2}$$

The above objective function exactly defines the formulation for regularized logistic regression. ■

Example 8.8 You are about to embark on learning a 3-dimensional logistic regression model, but the amount of data you have is very limited. Mr. Know-It-All comes to you and tells you that he has some personal domain knowledge into the problem because of which he knows that the optimal weight vector \vec{w} for logistic regression is close to $\vec{w}_0 = [2, -1, 3]^T$. You trust Mr. Know-It-All. Discuss how you would use Mr. Know-It-All's domain knowledge in the form of regularization. What is the probabilistic interpretation of this type of regularization?

Solution: In order to use Mr. Know-It-All's domain knowledge, the regularization term should be changed in order to encourage the optimal weight vector to be closer to $\vec{w}_0 = [2, -1, 3]^T$, rather than simply make it have a small L_2 -norm. Therefore, the regularized objective function should be modified as follows:

$$J = \sum_{i=1}^n \ln(1 + \exp(-y_i \vec{w} \cdot \vec{x}_i^T)) + \frac{\lambda \|\vec{w} - \vec{w}_0\|^2}{2}$$

The probabilistic interpretation of this regularization is the weight vector has a prior distribution with a mean of \vec{w}_0 . The regularization parameter is inversely proportional to the variance of this prior distribution and therefore controls the strength of the regularization. ■

8.3.2.2 Gradient Descent and Stochastic Gradient Descent

The derivation of the gradient-descent updates in logistic regression is similar to the case of linear regression. The aforementioned loss function J needs to be differentiated with respect to the vector \vec{w} in order to perform the update:

$$\vec{w} \leftarrow \vec{w} - \alpha \frac{\partial J}{\partial \vec{w}}$$

here, $\alpha > 0$ is the learning rate, and the derivative of the scalar J with respect to the vector \vec{w} is expressed in the denominator layout of matrix calculus. The derivative of the objective function J in the previous section may be expressed as follows:

$$\frac{\partial J}{\partial \vec{w}} = \sum_{i=1}^n \frac{-\exp(-y_i \vec{w} \cdot \vec{x}_i^T) y_i \vec{x}_i^T}{1 + \exp(-y_i \vec{w} \cdot \vec{x}_i)} + \lambda \vec{w} = \sum_{i=1}^n \frac{-y_i \vec{x}_i^T}{1 + \exp(y_i \vec{w} \cdot \vec{x}_i^T)} + \lambda \vec{w}$$

Applying the aforementioned derivative to the gradient-descent update step, one obtains the following:

$$\vec{w} \leftarrow \vec{w}(1 - \alpha \lambda) + \alpha \sum_{i=1}^n \frac{y_i \vec{x}_i^T}{1 + \exp(y_i \vec{w} \cdot \vec{x}_i^T)}$$

One can also implement mini-batch stochastic gradient descent by sampling a subset S of points, where S contains the indices of the sampled training points between 1 and n . The corresponding mini-batch stochastic gradient descent update is as follows:

$$\vec{w} \leftarrow \vec{w}(1 - \alpha \lambda) + \alpha \sum_{i \in S} \frac{y_i \vec{x}_i^T}{1 + \exp(y_i \vec{w} \cdot \vec{x}_i^T)}$$

When the size of S reduces to a single point, one obtains pure stochastic gradient descent.

8.3.2.3 Interpreting Updates in Terms of Error Probabilities

The updates of both least-squares classification and regression can be expressed in terms of the errors of prediction as follows:

$$\vec{w} \leftarrow \vec{w}(1 - \alpha \lambda) - \alpha \sum_{i \in S} \vec{x}_i^T \underbrace{(\vec{w} \cdot \vec{x}_i^T - y_i)}_{\text{Error}(i)} \quad [\text{Least-squares classification}]$$

How is logistic regression related to the errors? In order to understand this point, first note that the likelihood of the i th training point (cf. Equation 8.17) is essentially the probability that the classification is correct. The corresponding probability of an error on the i th training point may be expressed as follows:

$$\begin{aligned} P(\text{incorrect prediction of } \vec{x}_i) &= 1 - p_{[Y_i | \vec{X}_i = \vec{x}_i]}(y_i) \\ &= 1 - \text{Sigmoid}(y_i \vec{w} \cdot \vec{x}_i) \\ &= \frac{1}{1 + \exp(y_i \vec{w} \cdot \vec{x}_i^T)} \end{aligned}$$

Given this expression for the probability of error on the training instance \vec{x}_i , it is instructive to examine the update of logistic regression and compare it to the expression for the above error:

$$\vec{w} \leftarrow \vec{w} (1 - \alpha \lambda) + \alpha \sum_{i \in S} \frac{y_i \vec{x}_i^T}{1 + \exp(y_i \vec{w} \cdot \vec{x}_i^T)}$$

It is easy to see that the update can be expressed in terms of the probability of errors as follows:

$$\vec{w} \leftarrow \vec{w} (1 - \alpha \lambda) + \alpha \sum_{i \in S} y_i \vec{x}_i^T P(\text{incorrect prediction of } \vec{x}_i) \quad (8.18)$$

Therefore, while least-squares regression/classification performs the updates based on the *magnitudes* of the errors, logistic regression performs the updates based on the *probabilities* of the errors. When the probabilities of the errors on the training data are small, logistic regression will also make small updates. This is intuitively reasonable, and follows a similar principle to least-squares regression, which makes small updates when the errors are small. However, since the errors are modeled in a more realistic way with Bernoulli probabilities, logistic regression tends to provide superior results.

Example 8.9 Consider a 20-dimensional logistic regression problem (including the engineered bias feature value of 1) in which two training points have feature values of $[2, 1]$ and $[3, 1]$, and class labels of -1 and 1 , respectively. Note that the second feature value is 1 in both training instances because it is an engineered feature capturing the bias. The logistic regression problem uses a training data set of a million points to find the coefficient vector to be $\vec{w} = [2, -2]^T$. Find the probability of misclassification that logistic regression assigns to each of the above training points using the sigmoid function. Rationalize the magnitude of these probabilities in comparison with 0.5.

Solution: Logistic regression assigns a probability of error equal to 0.5 for points lying on the separator (i.e., satisfying $\vec{w} \cdot \vec{x}_i^T = 0$). This is because the probability of error for such points is $1/(1 + \exp(0)) = 0.5$. For points in which the sign of $\vec{w} \cdot \vec{x}_i^T$ matches y_i , the probability of error is less than 0.5, and vice versa.

For the first training point, the value of $\vec{w} \cdot \vec{x}_i^T$ is $2 \times 2 - 2 \times 1 = 2$. Note that the sign of this value is opposite to the class label, which is -1 for the first point. Therefore, we expect the probability of error to be greater than 0.5. The probability of error is given by the following:

$$\text{Error Probability} = \frac{1}{1 + \exp(y_i \vec{w} \cdot \vec{x}_i^T)} = \frac{1}{1 + \exp(-2)} \approx 0.88$$

For the second training point, the value of $\vec{w} \cdot \vec{x}_i^T$ is $3 \times 2 - 1 \times 2 = 4$. Note that the sign of this value is the same as the class label, which is +1 for the second point. Therefore, we expect the probability of error to be less than 0.5. The probability of error is given by the following:

$$\text{Error Probability} = \frac{1}{1 + \exp(y_i \vec{w} \cdot \vec{x}_i^T)} = \frac{1}{1 + \exp(4)} \approx 0.0180$$

■

8.3.3 Multinomial Logistic Regression

Least-squares classification and logistic regression are both designed for binary classification, wherein each y_i is drawn from $\{-1, +1\}$. However, many classification settings are multiway settings in which the class variable may be drawn from one of k values denoted by $\{1, \dots, k\}$. Therefore, the value of y_i takes on one of these k categorical values $\{1, \dots, k\}$, which are assumed to be unordered. The fact that these values are unordered is important because it implies that one cannot use numeric models (like regression) that use the ordering between the dependent variable values in order to define the distribution of y_i .

The multinomial logistic regression model is a direct k -way generalization of the logistic regression model. In order to classify a test instance \vec{z} into one of k classes, the model uses k coefficient vectors denoted by $\vec{w}_1 \dots \vec{w}_k$ and then computes $\vec{w}_j \cdot \vec{z}^T$ for each $j \in \{1 \dots k\}$. The class with the largest value of $\vec{w}_j \cdot \vec{z}^T$ is reported as the relevant one. It is also possible to predict the probability of \vec{z} belonging to class j to be proportional to $\exp(\vec{w}_j \cdot \vec{z}^T)$. In the next section, we will introduce the probabilistic model for multinomial logistic regression.

8.3.3.1 The Probabilistic Model

The multinomial logistic regression is a direct k -way generalization of the binary logistic regression model. Therefore, the probability model for generation of y_i is the k -way generalization of the Bernoulli distribution, which happens to be the categorical distribution. For each training point \vec{x}_i , the k different values $\vec{w}_1 \vec{x}_i^T, \vec{w}_2 \cdot \vec{x}_i^T, \dots, \vec{w}_k \cdot \vec{x}_i^T$ are computed, which are used to compute the categorical probabilities. The class variable y_i is assumed to be drawn from a categorical distribution, in which the probability of the j th class is proportional to $\exp(\vec{w}_j \cdot \vec{x}_i^T)$. Therefore, the random variable Y_i with realized value y_i is assumed to be drawn from the following distribution:

$$Y_i \sim \text{Categorical} \left(\frac{\exp(\vec{w}_1 \cdot \vec{x}_i^T)}{\sum_{r=1}^k \exp(\vec{w}_r \cdot \vec{x}_i^T)}, \dots, \frac{\exp(\vec{w}_j \cdot \vec{x}_i^T)}{\sum_{r=1}^k \exp(\vec{w}_r \cdot \vec{x}_i^T)}, \dots, \frac{\exp(\vec{w}_k \cdot \vec{x}_i^T)}{\sum_{r=1}^k \exp(\vec{w}_r \cdot \vec{x}_i^T)} \right)$$

The above model can be used to compute the probability that a training instance \vec{x}_i belongs to the j th class:

$$P(\vec{x}_i \text{ in } j\text{th class}) = \frac{\exp(\vec{w}_j \cdot \vec{x}_i^T)}{\sum_{r=1}^k \exp(\vec{w}_r \cdot \vec{x}_i^T)}$$

However, since the class label for training instances is already known to belong to class y_i , the above equation can be transformed into a likelihood value for the training instance (\vec{x}_i, y_i) simply by replacing the class index j with y_i in the above equation as follows:

$$\text{Likelihood}(\vec{x}_i, y_i) = \frac{\exp(\vec{w}_{y_i} \cdot \vec{x}_i^T)}{\sum_{r=1}^k \exp(\vec{w}_r \cdot \vec{x}_i^T)}$$

The above computation of the likelihood can be used for maximum likelihood estimation.

8.3.3.2 Maximum Likelihood Estimation

The likelihood fit $\mathcal{L}(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n, \vec{w}_1, \dots, \vec{w}_k)$ of multinomial logistic regression over the entire training data set can be computed as the products of the likelihoods over the individual training points as follows:

$$\mathcal{L}(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n, \vec{w}_1, \dots, \vec{w}_k) = \prod_{i=1}^n \text{Likelihood}(\vec{x}_i, y_i, \vec{w}_1, \dots, \vec{w}_k) = \prod_{i=1}^n \frac{\exp(\vec{w}_{y_i} \cdot \vec{x}_i)}{\sum_{r=1}^k \exp(\vec{w}_r \cdot \vec{x}_i)}$$

The corresponding negative log-likelihood fit is the negative logarithm of the aforementioned likelihood fit, which is as follows:

$$\mathcal{LL}(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n, \vec{w}_1, \dots, \vec{w}_k) = - \sum_{i=1}^n \vec{w}_{y_i} \cdot \vec{x}_i^T + \sum_{i=1}^n \ln \left(\sum_{r=1}^k \exp(\vec{w}_r \cdot \vec{x}_i^T) \right)$$

In order to regularize the objective function, the squared sum of the norms on the weight parameters are added as penalties to the objective function. We leave the derivation of L_2 -regularization as an exercise:

Example 8.10 Show that when MAP estimation is applied with Gaussian priors to multinomial logistic regression, it is equivalent to performing L_2 -regularization on the objective function derived from maximum likelihood estimation.

Solution: The Gaussian prior on the weight parameters can be written as the products of the Gaussians on the individual weight parameters:

$$f_{\vec{W}_1, \dots, \vec{W}_k}(\vec{w}_1, \dots, \vec{w}_k) = \left(\frac{1}{2\pi\sigma_0^2} \right)^{d \cdot k / 2} \prod_{r=1}^k \exp \left(-\frac{\|\vec{w}_r\|^2}{2\sigma_0^2} \right)$$

Here, the hyper-parameter σ_0 controls the degree of regularization. Small values of σ_0 will cause greater regularization. On adding the negative of the log-prior to the negative log-likelihood function, one can obtain the negative log-posterior after ignoring constant terms (using Equation 6.20 of Chapter 6). On setting the regularization parameter λ to $1/\sigma_0^2$, the following objective function is obtained:

$$J = - \sum_{i=1}^n \vec{w}_{y_i} \cdot \vec{x}_i^T + \sum_{i=1}^n \ln \left(\sum_{r=1}^k \exp(\vec{w}_r \cdot \vec{x}_i^T) \right) + \frac{\lambda}{2} \sum_{r=1}^k \|\vec{w}_r\|^2$$

Since $\lambda = 1/\sigma_0^2$, large values of λ cause greater regularization. The probabilistic interpretation of regularization is exactly the same as that in the case of least-squares regression (discussed in section 7.7 of Chapter 7). ■

8.3.3.3 Gradient Descent and Stochastic Gradient Descent

Let $I(a, b)$ be an indicator function with categorical (class label) arguments, which takes on the value of 1 when $a = b$, and 0, otherwise. The derivative of J with respect to the vector

\vec{w}_j can be computed as follows:

$$\begin{aligned}\frac{\partial J}{\partial \vec{w}_j} &= -\sum_{i=1}^n I(y_i, j) \vec{x}_i^T + \sum_{i=1}^n \frac{\vec{x}_i^T \exp(\vec{w}_j \cdot \vec{x}_i^T)}{\sum_{r=1}^k \exp(\vec{w}_r \cdot \vec{x}_i^T)} + \lambda \vec{w}_j \\ &= -\sum_{i=1}^n \vec{x}_i^T [I(y_i, j) - P(\text{Predict } \vec{x}_i \text{ to class } j)] + \lambda \vec{w}_j\end{aligned}$$

The above derivative is expressed in the denominator layout of matrix calculus, and can be used to make gradient-descent updates.

The aforementioned derivative of the log-likelihood function can be used in order to make gradient-descent updates. The gradient-descent update is as follows:

$$\vec{w}_j \leftarrow \vec{w}_j - \alpha \frac{\partial J}{\partial \vec{w}_j} \quad \forall j \in \{1 \dots k\}$$

here, $\alpha > 0$ is the learning rate. Substituting for the derivative in the above expression, the update is as follows:

$$\vec{w}_j \leftarrow \vec{w}_j (1 - \alpha \lambda) + \alpha \sum_{i=1}^n \vec{x}_i^T [I(y_i, j) - P(\text{Predict } \vec{x}_i \text{ to class } j)] \quad \forall j \in \{1 \dots k\}$$

The above update can also be expressed in the form of mini-batch stochastic gradient descent with a set S of training points. Here, the set S contains indices of a few hundred randomly sampled points from the training data. The mini-batch stochastic gradient descent update is as follows:

$$\vec{w}_j \leftarrow \vec{w}_j (1 - \alpha \lambda) + \alpha \sum_{i \in S} \vec{x}_i^T [I(y_i, j) - P(\text{Predict } \vec{x}_i \text{ to class } j)] \quad \forall j \in \{1 \dots k\}$$

When the set S contains exactly one point, this update reduces to pure stochastic gradient descent.

8.3.3.4 Probabilistic Interpretation of Gradient Descent Updates

The updates of gradient descent are expressed in terms of the probabilities. Since all updates thus far have been connected in some form or other to either the magnitude of an error or the probability of an error, it is also useful to examine the update of gradient descent update. First note that the mini-batch stochastic gradient-descent update can be written in the following form:

$$\vec{w}_j \leftarrow \vec{w}_j (1 - \alpha \lambda) + \alpha \sum_{i \in S} \vec{x}_i^T P_{ij} \quad \forall j \in \{1 \dots k\}$$

Here, P_{ij} is a signed probability value drawn from $(-1, 1)$, which can be expressed as follows:

$$P_{ij} = [I(y_i, j) - P(\text{predict } \vec{x}_i \text{ to class } j)]$$

The interpretation of the signed probability P_{ij} depends on whether the categorical class label y_i of the i th point matches the index j of the weight vector \vec{w}_j being updated. In the event that $y_i = j$, the notation P_{ij} is equal to $(1 - P(\text{predict } \vec{x}_i \text{ to class } j))$, which is the probability that \vec{x}_i will be (incorrectly) not predicted to class j . On the other hand,

if $y_i \neq j$, the quantity P_{ij} is the negative of the probability that \vec{x}_i will be incorrectly predicted to class j . Therefore, if we create a new binary label $y_i^{(j)} \in \{-1, +1\}$ for each instance-class pair, depending on whether or not \vec{x}_i belongs to class j , one can express the update as follows:

$$\vec{w}_j \leftarrow \vec{w}_j(1 - \alpha\lambda) + \alpha \sum_{i \in S} y_i^{(j)} \vec{x}_i^T P(\text{incorrect prediction of } \vec{x}_i \text{ w.r.t. } y_i^{(j)}) \quad \forall j \in \{1 \dots k\}$$

This update is almost identical to the logistic regression update of Equation 8.18, except that the binary class labels y_i of logistic regression have been replaced by binary labels $y_i^{(j)}$ that are specific to class j in multinomial regression when updating \vec{w}_j . This is not particularly surprising, considering the fact that *logistic regression is a special case of multinomial logistic regression*. When multinomial regression is used for the case $k = 2$ and the weight vectors \vec{w}_1 and \vec{w}_2 of the two classes are initialized to the negatives of one another, the updates are identical to binary logistic regression — furthermore, the mutually negative relationship of the two weight vectors is maintained (see Exercises 12 and 13).

8.4 Beyond Classification: Ordered Logit Model

The discussion in this chapter has shown that the main difference between the regression model and discriminative classification models is the manner in which the dependent variable is modeled probabilistically. Since the dependent variable is numeric in the case of regression, the variable is modeled using a Gaussian distribution. Similarly, a categorical or Bernoulli distribution may be used to model a class label in the case of the classification setting (such as in the case of logistic regression). Ordered logit regression extends the idea to ordinal variables. Ordinal attribute values are *ordered categories*. For example, a degree-type variable with value instantiations such as bachelors, masters, or doctorate is seen as ordinal; there is a clear ordering among the attribute values but the precise distances among different attribute values cannot be known. This general class of models is referred to as *ordinal regression*, of which ordered logit regression is a special case. Ordinal regression is sometimes referred to as ordinal classification, because the dependent variables take on the characteristics of both numeric and categorical data.

In ordered logit regression, it is assumed that we have n training pairs denoted by $(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_n, y_n)$. Each y_i is an ordinal variable with k possible ordered values denoted by $a_1 \dots a_k$. It is important to note that these values $a_1 \dots a_k$ are not numerical values but they are ordered *categorical* values satisfying the following ordered relationship:

$$a_1 < a_2 < \dots < a_{k-1} < a_k$$

In order to model the variable y_i , a total of $(k - 1)$ threshold parameters are used, which are denoted by $\theta_1, \theta_2, \dots, \theta_{k-1}$, satisfying the following relationship:

$$\theta_1 < \theta_2 < \dots < \theta_{k-1}$$

The variable y_i satisfies the following relationship:

$$P(y_i \leq a_j) = \text{Sigmoid}(\theta_j - \vec{w} \cdot \vec{x}_i^T) \quad \forall j \in \{1 \dots k - 1\}$$

The sigmoid function is defined in the same manner as discussed in section 8.3.2 on logistic regression. Note that the above equation does not include the case where $j = k$. This is

because we have the following:

$$P(y_i = a_k) = 1 - P(y_i \leq a_{k-1}) = 1 - \text{Sigmoid}(\theta_{k-1} - \vec{w} \cdot \vec{x}_i^T)$$

The above equations express the probability of the ordinal variable in a “cumulative” distribution form (insofar as one can talk about cumulative distributions with ordinal variables). It is also possible to write the ordinal variable as a probability mass function:

$$p_Y(a_j) = \begin{cases} \text{Sigmoid}(\theta_j - \vec{w} \cdot \vec{x}_i^T) & \text{if } j = 1 \\ \text{Sigmoid}(\theta_j - \vec{w} \cdot \vec{x}_i^T) - \text{Sigmoid}(\theta_{j-1} - \vec{w} \cdot \vec{x}_i^T) & \text{if } 2 \leq j < k \\ 1 - \text{Sigmoid}(\theta_{j-1} - \vec{w} \cdot \vec{x}_i^T) & \text{if } j = k \end{cases}$$

It is noteworthy that by using dummy parameter values $\theta_0 = -\infty$ and $\theta_k = +\infty$, it is possible to consolidate the aforementioned case-wise analysis into a single case as follows:

$$p_Y(a_j) = \text{Sigmoid}(\theta_j - \vec{w} \cdot \vec{x}_i^T) - \text{Sigmoid}(\theta_{j-1} - \vec{w} \cdot \vec{x}_i^T) \quad \forall j \in \{1, \dots, k\}$$

The simplified expression introduced above will be utilized for subsequent analysis. The parameters \vec{w} and $\theta_1 \dots \theta_{k-1}$ are learned using maximum-likelihood estimation. Given a test-instance \vec{z} , its class label is determined by computing $\vec{w} \cdot \vec{z}^T$, and determining the first index j for which $\vec{w} \cdot \vec{z}^T$ is no larger than θ_j (including the dummy values for θ_0 and θ_k). Then, the class label is reported as a_j . Next, the maximum-likelihood estimation process of the parameters is described.

8.4.1 Maximum Likelihood Estimation for Ordered Logit

As in the case of logistic regression, the maximum likelihood estimation of the ordered logit model computes the product of the probabilities of each ordinal label y_i over the training data. The likelihood value for the i th training data point is as follows:

$$\text{Likelihood of } (\vec{x}_i, y_i) = p_Y(y_i) = \sum_{j=1}^k I_{ij} p_Y(a_j)$$

Here, the indicator variable I_{ij} is 1 if $y_i = a_j$ and 0, otherwise. Substituting the value of the probability $p_Y(a_j)$, one obtains the following:

$$\text{Likelihood}(\vec{x}_i, y_i, \vec{w}, \vec{\theta}) = \sum_{j=1}^k I_{ij} [\text{Sigmoid}(\theta_j - \vec{w} \cdot \vec{x}_i^T) - \text{Sigmoid}(\theta_{j-1} - \vec{w} \cdot \vec{x}_i^T)]$$

It is noteworthy that the likelihood value depends on the parameters \vec{w} and $\vec{\theta} = [\theta_0, \dots, \theta_k]^T$. Note that θ_0 and θ_k are fixed to $-\infty$ and $+\infty$, respectively. The aforementioned point-specific likelihood value can be converted to the likelihood over the entire training data by using the products of the likelihoods for the individual training points:

$$\begin{aligned} \mathcal{L}(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n, \vec{w}, \vec{\theta}) &= \prod_{i=1}^n \text{Likelihood}(\vec{x}_i, y_i, \vec{w}, \vec{\theta}) \\ &= \prod_{i=1}^n \sum_{j=1}^k I_{ij} [\text{Sigmoid}(\theta_j - \vec{w} \cdot \vec{x}_i^T) - \text{Sigmoid}(\theta_{j-1} - \vec{w} \cdot \vec{x}_i^T)] \end{aligned}$$

The negative log-likelihood of the data can then be computed by taking the negative logarithm of the above expression:

$$\mathcal{LL}(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n, \vec{w}, \vec{\theta}) = - \sum_{i=1}^n \ln \left(\sum_{j=1}^k I_{ij} [\text{Sigmoid}(\theta_j - \vec{w} \cdot \vec{x}_i^T) - \text{Sigmoid}(\theta_{j-1} - \vec{w} \cdot \vec{x}_i^T)] \right)$$

Since exactly one of the values of I_{ij} is 1 for fixed i and varying j , one can bring the indicator variable and the summation outside the logarithm as follows:

$$\mathcal{LL}(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n, \vec{w}, \vec{\theta}) = - \sum_{i=1}^n \sum_{j=1}^k I_{ij} \ln (\text{Sigmoid}(\theta_j - \vec{w} \cdot \vec{x}_i^T) - \text{Sigmoid}(\theta_{j-1} - \vec{w} \cdot \vec{x}_i^T))$$

This expression can be used to compute the loss function of ordered logit regression. In other words, one needs to find a coefficient vector \vec{w} and vector $\vec{\Theta}$ that minimizes the following objective function:

$$\text{Minimize}_{\vec{w}, \vec{\theta}} J = - \sum_{i=1}^n \sum_{j=1}^k I_{ij} \ln (\text{Sigmoid}(\theta_j - \vec{w} \cdot \vec{x}_i^T) - \text{Sigmoid}(\theta_{j-1} - \vec{w} \cdot \vec{x}_i^T))$$

As in all the optimization models, one can avoid overfitting by adding L_2 -regularization. The regularized objective function may be written as follows:

$$\text{Minimize}_{\vec{w}, \vec{\theta}} J = - \sum_{i=1}^n \sum_{j=1}^k I_{ij} \ln (\text{Sigmoid}(\theta_j - \vec{w} \cdot \vec{x}_i^T) - \text{Sigmoid}(\theta_{j-1} - \vec{w} \cdot \vec{x}_i^T)) + \frac{\lambda}{2} (\|\vec{w}\|^2 + \|\vec{\theta}\|^2)$$

The regularization adds a Gaussian prior to the distribution of the coefficient vectors \vec{w} and $\vec{\Theta}$. We leave the derivation of MAP estimation as an exercise:

Example 8.11 Derive the additive term in L_2 -regularization of the ordered logit model as an additive term of MAP estimation over MLE estimation.

Solution: The prior on the parameters of the order logit model follow a Gaussian distribution with variance σ_0^2 across each parameter:

$$f_{\vec{\Theta}, \vec{W}}(\vec{\theta}, \vec{w}) \propto \exp \left(-\frac{\|\vec{w}\|^2 + \|\vec{\theta}\|^2}{2\sigma_0^2} \right)$$

Here, σ_0 is a hyper-parameter that controls the degree of regularization. Based on Equation 6.20 of Chapter 6, the negative logarithm of the prior needs to be added to negative logarithm of the likelihood to create the objective function of MAP estimation (ignoring constant terms). This additive term is given by the negative logarithm of the above, which is expressed as follows using $\lambda = 1/\sigma_0^2$:

$$\text{Additive Term} = \text{Constant} + \frac{\lambda}{2} (\|\vec{w}\|^2 + \|\vec{\theta}\|^2)$$

Ignoring constants, the above expression is exactly the L_2 -regularization term. ■

The gradient descent steps for ordinal regression can be derived in a manner similar the case of logistic regression. These steps are omitted as and left to the reader as an exercise.

Problem 8.6 (Gradient Descent for Ordinal Regression) *Using the loss function discussed in this section, work out the gradient-descent steps for ordinal regression. Extend your analysis to stochastic gradient descent. How does L_2 -regularization affect the updates?*

A key observation is that almost any type of dependent variable can be addressed with a discriminative model, as long as an appropriate probability distribution can be used to model the dependent variable.

8.5 Summary

This chapter introduces a probabilistic view of classification models. Probabilistic models for classification can be either generative or discriminative. Generative models create the entire data set using a pre-defined distribution, which is typically a mixture model of different classes. Discriminative models attempt to model the dependent variable probabilistically as a function of the feature variables and learnable parameters. In both cases, maximum-likelihood estimation is used to define a loss function. Minimizing this loss function yields the learnable parameters. Discriminative models can be generalized to any type of dependent variable, as long as an appropriate probability distribution for the dependent variable can be defined. A specific example in which the dependent variable is ordinal is the problem of ordinal regression. In this case, the ordered logit model is used to define a log-likelihood loss function.

8.6 Further Reading

The problem of classification has been discussed extensively in the machine learning literature [1, 3, 10, 33], although this book differs from earlier expositions in terms of providing a probabilistic view. The linear algebra perspective on classification is discussed in [6]. However, the optimization perspective discussed in [6] finds its basis in the (maximum-likelihood) probabilistic models discussed in this book. In general, many of the loss-based formulations in machine learning can be fully explained using probabilistic formulations. Generalized additive models are discussed in [34, 45] — these models extend discriminative models like logistic regression to other types of dependent variables. The ordered logit model discussed in this chapter is one such example. Another example of a classification model that uses a different probability distribution and link function to model a binary class variable is the *probit model* [45]. These models are also referred to as *generalized linear models*.

8.7 Exercises

1. Design a generative Bayes model for classification of multidimensional numeric data in which each numeric attribute is modeled using an exponential distribution. Where would such data arise?
2. Use the ideas discussed in the previous chapter to create kernelized versions of the classification models discussed in this chapter. Use explicit feature engineering, so that the ideas can be used in a general way across different classification models.
3. Discuss the probabilistic interpretation of L_1 -regularization when it is applied to logistic regression. How are the updates different when L_1 -regularization is used?

4. For the logistic regression model for binary classification, how would you interpret the sign of each component of the coefficient vector \vec{w} ?
5. For the multinomial logistic regression model for multiway classification, interpret the sign of each component of the coefficient vectors $\vec{w}_1, \dots, \vec{w}_k$ (based on notations discussed in the chapter).
6. Consider the binary classification setting with labels Y drawn from $+1$ and -1 . A single predictor and a feature engineered variable with value of 1 exists for the bias. The sole predictor X is standardized and the sample covariance $\hat{\sigma}_{XY}$ between the predictor and class variable is 2.3. The average of the class variable values is 0.3. Find the optimal least-squares classification model.
7. Describe the steps involved (and probabilistic expressions) in creating a Bayes classifier for continuous data using kernel density estimates at the test point (instead of a Gaussian model). Argue why this approach is closely related to a k -nearest neighbor classifier. A k -nearest neighbor classifier retrieves the closest k training points to a test instance and reports the class with the largest frequency among them.
8. An email spam classifier is constructed using the presence or absence of the five keywords **Free**, **Winner**, **Sweepstakes**, **Giftcard**, and **Deal**. The training data set contains 18% spam and 82% non-spam emails. The fraction of non-spam emails in which each of the five keywords are contained are 0.03, 0.04, 0.01, 0.05, and 0.07, respectively. The corresponding fractions for spam emails are 0.13, 0.18, 0.06, 0.18, and 0.16, respectively. Using the Bernoulli model for Bayes classification find the probability that an email containing only **Winner** and **Sweepstakes** is spam.
9. Consider a logistic regression problem in which the value of $\vec{w} \cdot \vec{x}_i$ is -0.7 , whereas the class label y_i is $+1$ for the training point (\vec{x}_i, y_i) . What is the probability that this training point has been misclassified by the logistic regression model? Write an equation for the unregularized update of \vec{w} in terms of \vec{x}_i for stepsize $\alpha = 0.1$.
10. This problem is a rework of Example 8.1 with the use of the Laplace distribution on attributes rather than the Gaussian distribution:

$$f_Z(z) = \frac{1}{2s} \exp\left(-\frac{\|z - \mu\|_1}{s}\right)$$

You have a training data set of individuals with two attributes (not including the class label) corresponding to age and salary. Each person in this data set is either in a managerial position or not in a managerial position, which is the binary class label. Exactly 10% of the points correspond to managers. You calculate the sample mean and mean absolute deviation on both attributes by class group. You find the following: (i) Managers have mean age 44 and mean absolute deviation of 12. (ii) Managers have mean salary of 100,000 and mean absolute deviation of 10,000. (iii) Non-managers have mean age 39 and mean absolute deviation of 11. (iv) Non-managers have mean salary of 65,000 and mean absolute deviation of 15,000. John is a 50 year old employee with a salary of 80,000. Find the probability that he is a manager assuming that you use a Bayes classifier with the independence assumption across attributes and a Laplace distribution on both attributes.

- 11.** Suppose that you whiten the features of a binary classification data set (i.e., transform them with PCA and then standardize them) so that the features have zero means, unit variances, and zero covariances. The labels are drawn from $\{-1, +1\}$. The (whitened) feature mean vectors of the negative and positive classes are $\vec{\mu}_0$ and $\vec{\mu}_1$, respectively. The fraction of points belonging to the positive class is f . Show that the optimal weight vector \vec{w} of unregularized least-squares classification is proportional to $f\vec{\mu}_1 - (1-f)\vec{\mu}_0$.
- 12.** As discussed in the chapter, the stochastic gradient-descent update of multinomial logistic regression For training point (\vec{x}_i, y_i) is as follows:
- $$\vec{w}_j \leftarrow \vec{w}_j + \alpha \vec{x}_i^T [I(y_i, j) - P(\text{predict } \vec{x}_i \text{ to class } j)] \quad \forall j \in \{1 \dots k\}$$
- Here, $I(y_i, j)$ is an indicator function that is 1 when $y_i = j$. Show that the updates to all the different \vec{w}_j for a single training point add up to 0.
- 13.** Consider the use of multinomial logistic regression for $k = 2$ classes. The weight vectors \vec{w}_0 and \vec{w}_1 of the negative and positive classes are initialized to the negatives of one another. Use the result of Exercise 12 to show that this property is maintained by updates. Suppose that binary logistic regression updates are implemented via stochastic gradient descent in the same order of processing of training instances as multinomial logistic regression, and the weight vector \vec{w} of binary logistic regression is initialized to $2\vec{w}_1$. Show that the relationship between the weight vectors is maintained by the updates of the two algorithms as long as the step-size in binary logistic regression is twice that in multinomial logistic regression. Furthermore, show that both algorithms provide the same probability of classification of test instances.
- 14.** The text discusses how to use the Bayes condition in proportional form (cf. Equation 8.4) in order to predict the class with the largest probability. Consider a setting of two classes in which one of the classes (e.g., *fraud*) is particularly important and you want to rank test instances based on probability of a fraud occurring. Discuss the problem associated with using the Bayes equation in proportional form in this case. How would you use a Bayes rule in such a setting?
- 15.** Consider a Bayes classifier built using the Gaussian assumption (without independence of attributes as in Problem 8.1). The training data for class 0 contains the points $(1, 2)$, $(-2, 1)$, $(1, -1)$, and $(0, -2)$. The training data for class 1 is $(3.1, 3)$, $(1, 1)$, $(2, 2.1)$, $(-1, -1)$, $(-2, -2.1)$, and $(-3.1, -3)$. What are the prior probabilities? What are the Gaussian distributions of the two classes obtained by using MLE? Find the probability that the test point belongs to class 0 using the Bayes method. Feel free to use internet calculators to compute the means and covariance matrices of the Gaussians.
- 16.** A data set contains d attributes corresponding to the counts of the number of network failures in each of d different computer systems (which are largely independent of one another). Each row contains the aggregate number of failures of each type in a month of data, and the row is tagged with a binary label indicating whether or not there was an intrusion attack on the full system. The number of failures of each type is modeled by a Poisson distribution. Propose the details of a Bayes classifier in this setting.
- 17.** Consider a binary classification problem with labels drawn from $\{-1, +1\}$. The population-level means of the numeric attributes for each class are known with certainty. Discuss how this additional information affects the design of a Bayes classifier in which each attribute follows the exponential distribution.

18. Suppose that you are not given the original $n \times d$ data matrix D but an $n \times n$ similarity matrix S containing dot products between rows of D . You are also given labels for the rows of D . Therefore, we have $S = DD^T$. Assume that D is mean-centered. Show how you can create a Bayes classifier starting from S rather than D . Assume that an $n_t \times n$ test-train similarity matrix S_t is available. A key point of this exercise is to understand how one can extract the PCA representation of D from S . You might want to review the material on explicit feature engineering in Chapter 7. Now discuss what this result means in the context of applying the Bayes classifier (or any other classifier like logistic regression) to objects that are not multidimensional.
19. Use the ideas in Exercise 7 to implement the Bayes classifier with kernel density estimation in Python. Your main problem will be that the classification of each test instance requires the scanning of all points. Use the ideas in Exercise 27 of Chapter 6 to speed up the approach.
20. For the setting in Exercise 19, propose a toy scenario with a few 1-dimensional points to show how different values of the kernel bandwidth can lead to different classifications of the same test instance. Relate this phenomenon to the bias-variance trade-off.



Chapter 9

Unsupervised Learning: A Probabilistic View

“There is a striking difference between going to school and getting an education.”
— Michael Bassey Johnson

9.1 Introduction

The previous two chapters have introduced algorithms for supervised learning in which the dependent variable has a significant influence on the learned model. In unsupervised learning, this type of supervision is not available. Rather, the model learns the trends and patterns in the underlying data in terms of carefully designed summaries (models). These models can be used to create new insights. Some types of unsupervised learning are also referred to as *self-supervised learning*, because the same attributes are considered independent and dependent variables; in other words, all attributes of the data can be predicted from themselves (approximately) after passing them through a compressed representation model.

There is a wide variety of ways in which unsupervised learning can be performed, depending on the goals of the application at hand. Some example of unsupervised models are as follows:

- *Clustering*: In the clustering problem, the observations are partitioned into similar groups. In the probabilistic setting, the membership of points in groups is defined by a random variable. Therefore, each point has a probability of belonging to a particular group. In generative models for clustering, the groups are defined in the context of a mixture model, and each point is generated from one of the components of the mixture in probabilistic fashion. An interesting connection with the previous chapter is that the mixture model has exactly the same generative process as the naïve Bayes classifier. The main difference is that examples of pre-defined groups are available in

classification, whereas no such examples are available in clustering. Furthermore, the mixture model parameters can be used to create a compressed representation of the data.

- *Matrix factorization:* In matrix factorization, the $n \times d$ data matrix D can be represented as $D \approx UV^T$, where U is an $n \times k$ matrix and V is a $d \times k$ matrix for $k \ll \min\{n, d\}$. Since the matrices U and V can be represented using $(n+d)k$ entries, which is typically much less than the $n \cdot d$ entries in matrix D , it is assumed to be a compressed representation of the data. The matrices U and V can be learned by minimizing the Frobenius norm $\|D - UV^T\|_F^2$ (i.e., sum of squares of the entries) of the residual matrix $(D - UV^T)$. It is noteworthy that this loss function can be treated in an analogous manner to least-squares regression by imposing a Gaussian probability distribution on the residual entries. In other words, it is assumed that each entry of $(D - UV^T)$ is generated from a zero-centered Gaussian distribution.
- *Outlier detection:* Outlier detection is the problem of identifying a small number of data points that have a different generative process than the data distribution from which the vast majority of the remaining points were generated. For example, if the generative process is that of mixture modeling, any point with a low generative probability is an outlier. In general, outlier detection may be considered a complementary problem to that of building compressed models like clustering and matrix factorization, which attempt to maximize the likelihood of points being generated from the compressed probabilistic model. Outliers can be generated as byproducts of the clustering and matrix factorization models, when some (outlier) points continue to have low likelihood of being generated from the model.

In some cases such as mixture-model clustering, a one-to-one correspondence exists with supervised models like the naïve Bayes model. For example, the Bayes model for classification with Gaussian mixtures is identical to the Gaussian mixture model for clustering (in terms of its generative characteristics); the main difference is that the assignment of points to mixture components is known in classification for training points (via labels), whereas the assignment is not available in clustering.

9.1.1 Chapter Organization

This chapter is organized as follows. The next section introduces mixture models for clustering. Different types of probability distributions are discussed in the context of clustering. Section 9.3 introduces matrix factorization along with its probabilistic interpretation. The outlier detection problem is discussed in section 9.4. A summary is given in section 9.5.

9.2 Mixture Models for Clustering

Mixture models for reconstructing distributions were introduced in section 6.4 of Chapter 6. The basic idea in these models is that the data are generated from multiple distributions. For each point in the data set, it is assumed that it is generated from a particular mixture component (i.e., probability distribution of a cluster), and then generated from this component. While the discussion in section 6.4 of Chapter 6 provides a generic description (without details of the distributions of the specific mixture components), this section provides more details of mixture models with specific types of mixture components. For completeness, the following discussion repeats some of the descriptions in section 6.4 of Chapter 6.

In the mixture model, it is assumed that the data is created by k different generating processes, each of which has its own probability distribution. It is common to assume that each generating process (mixture component or cluster) uses the same family of distributions (e.g., Gaussian distribution), although each component will have a different set of parameters. For example, each Gaussian component of a Gaussian mixture model will have a different center based on the location of the cluster. Similarly, different Gaussian clusters will have different variance corresponding to the “tightness” of that cluster.

The different mixture components may have different relative frequencies in the observed data. The number of mixture components is (typically) fixed up front and corresponds to the analyst’s estimate of the number of clusters in the data. The data is then used to estimate the relative frequency of each mixture component and the parameters of each of the mixture components. These parameters are estimated in a data-driven manner with the use of maximum-likelihood estimation. In addition, the probability of each point being generated by a particular mixture component is estimated (based on the values of its features). An important point about probabilistic clustering is that it does not create a “hard” partition of the data. Rather, points are probabilistically generated from different mixture components, and each point has a nonzero probability of being generated by the cluster corresponding to each mixture component.

It is noteworthy that the estimations of mixture parameters and the membership probabilities of points in clusters depend on each other, and they therefore need to be repeatedly derived from one another in an iterative manner. These two steps are referred to as the expectation and maximization steps. The estimation of mixture parameters is achieved with maximum-likelihood estimation in the maximization step, whereas the estimation of cluster assignment probabilities is achieved with the expectation step. The iterative approach starts by assigning random values of the parameters to each cluster and then repeats the following steps:

- *Expectation step:* Compute the probability of membership of each observed data point to each mixture component, assuming that the current parameters of different distributions are optimal. Thus, the *expected* membership is computed in this step.
- *Maximization step:* Assuming that the current probabilistic assignments of points to mixture components are correct, divvy up each point into different mixture components with weight equal to assignment probability. Perform maximum-likelihood estimation independently for each mixture component with its own set of weighted points (to derive the parameters for the different conditional distributions).

The expectation step is also referred to as the *E-step* and the maximization step is also referred to as the *M-step*. These two steps are iterated to convergence. The above description already provides an informal description of the *expectation-maximization algorithm*, which is also referred to as the *EM-algorithm*. Both sets compute different parameters of the generative process iteratively. The expectation step computes probabilities of membership of points in mixture components, whereas the maximization step computes the parameters of each mixture via the process of decomposition of the data set into different components (allowing fractional membership of points in components). Next, we will provide a more formal description with notations. We first recap the generating process of data created from a mixture distribution given in section 4.13 of Chapter 4.

We assume that the different mixture components are denoted by $\mathcal{G}_1, \dots, \mathcal{G}_k$, with corresponding conditional probability distributions denoted by $f_{\vec{X}|\mathcal{G}_1}(\vec{x}), f_{\vec{X}|\mathcal{G}_2}(\vec{x}), \dots, f_{\vec{X}|\mathcal{G}_k}(\vec{x})$. Therefore, there are a total of k mixture components. In the case when the data are dis-

crete, one can use probability mass functions $p_{\vec{X}|\mathcal{G}_1}(\vec{x})$, $p_{\vec{X}|\mathcal{G}_2}(\vec{x})$, ..., $p_{\vec{X}|\mathcal{G}_k}(\vec{x})$ instead of probability density functions. For greater generality, we will work with probability density functions rather than probability mass functions (although later sections will provide examples of cases where probability mass functions are used). The set of parameters associated with the density function $f_{\vec{X}|\mathcal{G}_j}(\vec{x})$ is the vector $\vec{\theta}_j$. For some distributions like the Bernoulli distribution for 1-dimensional data, this vector could contain only one element (Bernoulli success probability parameter), which makes it a scalar. The probability that the j th generative process is selected in the generation of an instance is given by $\alpha_j = P(\mathcal{G}_j)$. Therefore, we have:

$$\sum_{j=1}^k P(\mathcal{G}_j) = \sum_{j=1}^k \alpha_j = 1$$

The probability $P(\mathcal{G}_j)$ is the *prior* probability of an observed data point belonging to mixture component j , because it refers to the probability that we would predict an observation to belong to mixture component j without knowing anything about the data point. This prior probability simply reflects the relative frequencies of the points belonging to different mixture components. The prior probabilities are not known up front and therefore the vector of prior probabilities is treated as the k -dimensional parameter vector $\vec{\alpha} = [\alpha_1, \dots, \alpha_k]$, which needs to be estimated during the execution of the EM-algorithm. The basic generative process for a single observation in the data is as follows:

1. Roll a biased die whose k sides have probabilities $P(\mathcal{G}_1) \dots P(\mathcal{G}_k)$. Let the outcome of the die roll be the side j , which provides the identity of the mixture component from which the observation is generated.
2. Sample the data point \vec{x} from the probability distribution $f_{\vec{X}|\mathcal{G}_j}(\vec{x})$. The point \vec{x} is the output of one iteration of the generative model.

The aforementioned generative process can be repeated again and again in order to create a synthetic sample of the distribution that is as large as one needs. When using the model for clustering, one can assume that the generated data is the *outcome* of repeating this process n times, where n is the number of points in the data.

The unconditional density function for the generated data is given by the total probability rule of Chapter 3:

$$f_{\vec{X}}(\vec{x}) = \sum_{j=1}^k P(\mathcal{G}_j) f_{\vec{X}|\mathcal{G}_j}(\vec{x})$$

The negative log-likelihood function is therefore given by the following:

$$\mathcal{LL}(\vec{x}_1, \dots, \vec{x}_n, \vec{\theta}_1, \dots, \vec{\theta}_k, \vec{\alpha}) = - \sum_{i=1}^n \ln \left(\sum_{j=1}^k \alpha_j f_{\vec{X}|\mathcal{G}_j}(\vec{x}_i) \right)$$

Differentiating the negative log-likelihood function with respect to $\vec{\theta}_j$ in vector-calculus notation and setting to zero, one obtains the following (while recognizing that only the density function for the j th mixture component depends on $\vec{\theta}_j$):

$$\sum_{i=1}^n \underbrace{\frac{\alpha_j f_{\vec{X}|\mathcal{G}_j}(\vec{x}_i)}{\sum_{j=1}^k \alpha_j f_{\vec{X}|\mathcal{G}_j}(\vec{x}_i)}}_{\text{Bayes form}} \frac{1}{f_{\vec{X}|\mathcal{G}_j}(\vec{x}_i)} \frac{\partial f_{\vec{X}|\mathcal{G}_j}(\vec{x}_i)}{\partial \vec{\theta}_j} = \vec{0}$$

Here, a key observation is that one part of this expression is simply the posterior probability $\gamma_{ji} = P(\mathcal{G}_j | \vec{x}_i)$ based on the Bayes rule. Therefore, one can rewrite the above expression as follows:

$$\sum_{i=1}^n \gamma_{ji} \frac{1}{f_{\vec{X}|\mathcal{G}_j}(\vec{x}_i)} \frac{\partial f_{\vec{X}|\mathcal{G}_j}(\vec{x}_i)}{\partial \vec{\theta}_j} = \vec{0} \quad (9.1)$$

Note that this form of the optimality condition is a generalization of the condition for optimality with a single mixture distribution (cf. Equation 6.2); in this case, the j th mixture distribution masquerades as a single unconditional distribution, but the i th point has a weight equal to that of its posterior probability. Of course, since the posterior probability also depends on the distribution parameters $\vec{\alpha}$, $\vec{\theta}_1 \dots \vec{\theta}_k$, one cannot really solve the above equation in closed form (which is a different situation from all the distributions in section 6.3 without this weight). The EM algorithm gets around this problem with an iterative procedure of fixing the weights (posterior probabilities γ_{ji}) using the values of the parameters $\vec{\alpha}$, $\vec{\theta}_1 \dots \vec{\theta}_k$ in the last iteration and then using these posterior probabilities in the above equation to recompute the optimal values of these parameters in closed form. This is a common iterative approach used in solving equations that do not have neat closed-form solutions. Often, the closed-form solution to the above equation (with fixed posterior) probabilities will look similar to the single-distribution expressions computed in section 6.3, except that the contributions of points to these expressions are weighted down with their posterior probabilities. This pattern of weighting down the contributions is particularly true, when the MLE estimates are defined in terms of additive contributions of different points. Indeed, one can show that most of the distributions discussed in section 6.3 exhibit this property with the exception of the uniform distribution.

Next, we describe the expectation and maximization steps formally with notations. The purpose of the expectation step is to compute the *posterior* probability of the membership of a point in a cluster after one has already observed the values of the attributes in the data point. Therefore, the posterior probability incorporates information about the relative frequencies of points in mixture components *as well as* information about how the values of the points relate to the mixture components. The maximization component then recomputes all parameters with maximum-likelihood estimation, including the prior probabilities and the parameters of mixture components. The algorithm starts with setting the parameters to random values and then iterates through the following steps:

- *Expectation step:* Compute the probability of membership of each observed data point to each mixture component using the Bayes Rule:

$$P(\mathcal{G}_r | \vec{x}_i) = \frac{P(\mathcal{G}_r) \cdot f_{\vec{X}|\mathcal{G}_r}(\vec{x}_i)}{\sum_{s=1}^k P(\mathcal{G}_s) \cdot f_{\vec{X}|\mathcal{G}_s}(\vec{x}_i)} \quad \forall i, r$$

Recall that the posterior probability of membership of point i to mixture component r is denoted by γ_{ri} , and therefore the above equation yields γ_{ri} .

- *Maximization step:* First, note that the prior probability of each cluster is a parameter of a categorical distribution. As in the case of the Bernoulli distribution, the maximum-likelihood estimate of the probability of each category is derived as the fraction of points assigned to mixture component j . Since points have fractional memberships to different mixture components, this probability is computed as follows:

$$\alpha_j = P(\mathcal{G}_j) = \frac{\sum_{i=1}^n \gamma_{ji}}{n} \quad \forall j$$

Next, we discuss the derivation of the parameters of the different mixture components. Each mixture component is assumed to contain the same n points as the entire data set, albeit with different fractional weights γ_{ri} that sum to 1 for fixed point \vec{x}_i and varying mixture component \mathcal{G}_r . The optimality condition for the i th mixture component is contained in Equation 9.1, which is simply a posterior-weighted version of the optimality condition for a data set generated only by the i th mixture component. The key point is that the expectation step *has helped decompose the problem into k independent problems*, one for each mixture component. Depending on the specific choice of the distribution selected by the analyst for each mixture component, the specific details of the estimation step may be different.

The main design choice in the case of the expectation-maximization algorithm is the choice of the distribution to be used for each mixture component. This choice depends on the type of data. For example, a numeric data set requires the use of a Gaussian distribution for each mixture component. Many of these variations are discussed in this chapter.

9.2.1 Continuous Numeric Data: The Gaussian Distribution

In this case, the feature variables are continuous numeric values in d dimensions, and the common choice of distribution in this case is the d -dimensional Gaussian distribution. As in the case of the naïve Bayes model for classification, the assumption of conditional independence among the attributes is used. Consider a data set containing the n observations $\vec{x}_1 \dots \vec{x}_n$. The i th data point \vec{x}_i contains d dimensions denoted by $[x_{i1}, x_{i2}, \dots, x_{id}]$. The joint density function of the r th mixture component is denoted by $f_{\vec{X}}^{(r)}(\vec{x}_i)$, which can be expressed as the product of d marginal density functions (because of the conditional independence assumption). Each of these marginal distributions is a 1-dimensional normal (Gaussian) distribution. Assume that the parameters of the marginal probability density function $f_{X_j|\mathcal{G}_r}(\cdot)$ of the Gaussian distribution for the r th mixture model along the j th dimension are μ_{jr} and σ_{jr} , respectively. Therefore, the probability density of the i th d -dimensional observation $\vec{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]$ is obtained using the conditional independence assumption as follows:

$$f_{\vec{X}|\mathcal{G}_r}(\vec{x}_i) = \prod_{j=1}^d f_{X_j|\mathcal{G}_r}(x_{ij}) = \prod_{j=1}^d \frac{\exp(-(x_{ij} - \mu_{jr})^2/(2\sigma_{jr}^2))}{\sqrt{2\pi}\sigma_{jr}} \quad (9.2)$$

$$= \frac{\exp(-\sum_{j=1}^d [(x_{ij} - \mu_{jr})^2/(2\sigma_{jr}^2)])}{(2\pi)^{(d/2)} \prod_{j=1}^d \sigma_{jr}} \quad (9.3)$$

Note that this distribution is exactly the same as that used to model the r th mixture component in the case of the naïve Bayes model for classification (cf. section 8.2). The main difference is that class labels are not available for training purposes, which makes the problem much more challenging. An analogous relationship exists between the classes in the case of supervised learning and the cluster identifiers in the case of unsupervised learning.

These distributions can be reconstructed (i.e., their parameters can be learned) using the expectation-maximization algorithm. Unlike the supervised case in which the membership of points to different mixture components is known up front, the weighted assignment of points to mixture components is achieved with the use of the expectation step, which is an application of the Bayes rule. As in the case of classification, since the different dimensions are independent of one another, the parameters of each dimension can be estimated independently using the statistics of that dimension (cf. Lemma 6.1 in section 6.3.9 of Chapter 6).

Therefore, μ_{jr} is estimated as the weighted sample mean of the j th dimension for training examples belonging to the r th mixture component (where the weights are defined by the expectation step). Similarly, σ_{jr} is estimated as the (weighted) sample standard deviation of the j th dimension for training examples belonging to the r th mixture component.

In the expectation-step, the notation γ_{ri} denotes the intermediate posterior probability of point i belonging to the mixture component r , and is computed as follows:

$$\gamma_{ri} = P(\mathcal{G}_r | \vec{x}_i) = \frac{P(\mathcal{G}_r) \cdot f_{\vec{X}|\mathcal{G}_r}(\vec{x}_i)}{\sum_{s=1}^k P(\mathcal{G}_s) \cdot f_{\vec{X}|\mathcal{G}_s}(\vec{x}_i)} \quad \forall i, r$$

The Gaussian distribution is substituted in the right-hand side of the above expression in order to compute the expected probability of assignment for each data point. The following posterior probability is obtained from the expectation step:

$$\gamma_{ri} = \frac{\alpha_r \cdot \left[\exp(-\sum_{j=1}^d [(x_{ij} - \mu_{jr})^2 / (2\sigma_{jr}^2)]) / \prod_{j=1}^d \sigma_{jr} \right]}{\sum_{s=1}^k \alpha_s \cdot \left[\exp(-\sum_{j=1}^d [(x_{ij} - \mu_{js})^2 / (2\sigma_{js}^2)]) / \prod_{j=1}^d \sigma_{js} \right]} \quad (9.4)$$

The parameter set γ_{ri} is important from the perspective of clustering because it provides the probability of assignment of the data points to different clusters (at the end of the iterative process of repeating the E-step and M-step multiple times). In other words, the value γ_{ri} can also be interpreted as the strength of the soft assignment of data points to mixture components (clusters), which is one of the key outputs of the algorithm at termination.

The first expectation step uses randomly initialized parameters in the above equation in order to compute the assignment probabilities, which correspond to the weights γ_{ri} . Subsequently, these weights are used to compute $\hat{\alpha}_r = P(\mathcal{G}_r)$ as follows:

$$\hat{\alpha}_r = P(\mathcal{G}_r) = \frac{\sum_{i=1}^n \gamma_{ri}}{n} \quad \forall r$$

The mean and standard deviation of the r th mixture component are computed as follows (based on maximum-likelihood estimation):

$$\begin{aligned} \hat{\mu}_{jr} &= \frac{\sum_{i=1}^n \gamma_{ri} x_{ij}}{\sum_{i=1}^n \gamma_{ri}} \quad \forall j, r \\ \hat{\sigma}_{jr}^2 &= \frac{\sum_{i=1}^n \gamma_{ri} (x_{ij} - \hat{\mu}_{jr})^2}{\sum_{i=1}^n \gamma_{ri}} \quad \forall j, r \end{aligned}$$

Note that this estimation is exactly the same as that discussed in section 8.2.1 of Chapter 8, except that the points have been weighted by their posterior probabilities for maximum-likelihood estimation. The E-steps and M-steps are repeated to convergence, and the soft assignments of data points to clusters (i.e., the values of γ_{ri}) are reported as the outputs of the algorithm. The distribution parameters, such as the means and standard deviations provide information about the location and spread of each cluster. The plate diagram for the Gaussian mixture model is illustrated in Figure 9.1. It is noteworthy that this plate diagram is almost identical to the plate diagram for generation of test instances shown in Figure 8.2(b) of Chapter 8.

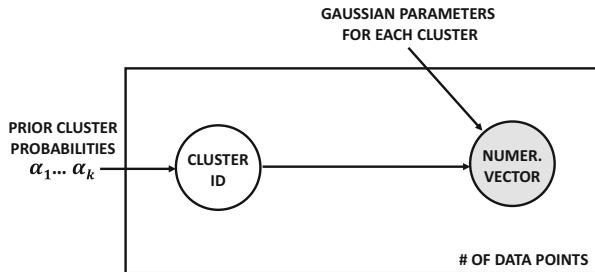


Figure 9.1: The generative process for Gaussian mixture modeling in terms of plate diagrams

Example 9.1 Suppose that you model gorilla heights as a mixture of two 1-dimensional normal distributions under the assumption that males and females will naturally separate out into two components. Your EM algorithm finds two mixture components \mathcal{G}_1 and \mathcal{G}_2 of means 4.0 feet and 6.9 feet, respectively. The corresponding standard deviations are 0.7 feet and 0.9 feet, respectively. The prior probabilities are estimated to be 0.6 and 0.4, respectively. You are given the heights of three gorillas from the data, which are Mr./Ms. Tiny at 3.5 feet, Mr./Ms. Middling at 5.4 feet, and Mr./Ms. Giant at 7.5 feet. What is the probability of each of these gorillas belonging to mixture components \mathcal{G}_1 and \mathcal{G}_2 ?

Solution: This problem is exactly identical to that of finding the probability of each class in a Bayes classifier. The only difference is that the parameters of each cluster were estimated using an EM algorithm rather than a more straightforward MLE approach using labels in the case of classification. Using the total Bayes rule in each case, and ignoring proportionality constants like $1/\sqrt{2\pi}$ in the Gaussian densities (which cancel out from the numerator and denominator), we obtain the following:

$$P(\mathcal{G}_1|Tiny) = \frac{0.6 \frac{1}{0.7} \exp\left(-\frac{(3.5-4.0)^2}{2*0.7^2}\right)}{0.6 \frac{1}{0.7} \exp\left(-\frac{(3.5-4.0)^2}{2*0.7^2}\right) + 0.4 \frac{1}{0.9} \exp\left(-\frac{(3.5-6.9)^2}{2*0.9^2}\right)} \approx \frac{0.664}{0.668} \approx 0.994$$

$$P(\mathcal{G}_1|Middling) = \frac{0.6 \frac{1}{0.7} \exp\left(-\frac{(5.4-4.0)^2}{2*0.7^2}\right)}{0.6 \frac{1}{0.7} \exp\left(-\frac{(5.4-4.0)^2}{2*0.7^2}\right) + 0.4 \frac{1}{0.9} \exp\left(-\frac{(5.4-6.9)^2}{2*0.9^2}\right)} \approx \frac{0.116}{0.227} \approx 0.511$$

$$P(\mathcal{G}_1|Giant) = \frac{0.6 \frac{1}{0.7} \exp\left(-\frac{(7.5-4.0)^2}{2*0.7^2}\right)}{0.6 \frac{1}{0.7} \exp\left(-\frac{(7.5-4.0)^2}{2*0.7^2}\right) + 0.4 \frac{1}{0.9} \exp\left(-\frac{(7.5-6.9)^2}{2*0.9^2}\right)} \approx 0$$

The probabilities of \mathcal{G}_2 can be calculated as the complements of the aforementioned probabilities. Note that *Middling* does not naturally belong to any of the clusters (with evenly split probabilities). As we will see later in this chapter, such points are often outliers. ■

Example 9.2 Consider a d -dimensional data set in which each attribute is a waiting time drawn from an exponential distribution. The 1-dimensional exponential distribution is as follows:

$$f_Z(z) = \lambda \exp(-\lambda z)$$

Rework the details of the expectation-maximization algorithm in this section using a d -dimensional exponential distribution with conditional independence. Assume that each mixture component and dimension has its own exponential parameter. Provide the details of both the E- and M-steps.

Solution: The notations used here are the same as used in the earlier section, such as the data points $\vec{x}_1 \dots \vec{x}_n$, number of mixture components k , mixture component notations $\mathcal{G}_1 \dots \mathcal{G}_k$, prior probabilities $\alpha_1 \dots \alpha_k$, and intermediate posterior assignment probabilities γ_{ri} . The exponential parameter for the j th dimension in the r th mixture component is λ_{jr} . The exponential parameters are initialized randomly to positive values. The value of each α_r can be initialized to $1/k$. In the E-step, the intermediate posterior assignment probabilities γ_{ri} are computed as follows:

$$\gamma_{ri} = \frac{\alpha_r \cdot \prod_{j=1}^d [\lambda_{jr} \exp(-\lambda_{jr} x_{ij})]}{\sum_{s=1}^k \alpha_s \cdot \prod_{j=1}^d [\lambda_{js} \exp(-\lambda_{js} x_{ij})]}$$

Once the fractional assignment of each point to each cluster has been computed in the form of γ_{ri} , it can be used for the M-step. The values of α_r and λ_{jr} are estimated as follows:

$$\alpha_r = \frac{\sum_{i=1}^n \gamma_{ri}}{n} \quad \forall r$$

$$\lambda_{jr} = \frac{\sum_{i=1}^n \gamma_{ri}}{\sum_{i=1}^n \gamma_{ri} x_{ij}}$$

The estimation of λ_{jr} is consistent with the formulas in Table 6.1. The E- and M-steps are iterated to convergence. ■

Example 9.3 (Deterministic EM Is k-Means) Consider the expectation-maximization algorithm of the previous section with the following simplifications: (i) Each prior probability α_s is set to $1/k$; (ii) each cluster standard deviation σ_{jr} is set to the same parameter σ along each dimension; and (iii) points are deterministically assigned to clusters with the highest assignment probability ($\gamma_{ri} \in \{0, 1\}$).

Solution: The same notations are used as in the description of the Gaussian mixture model. The only difference is that σ_{jr} is replaced by σ , since all variances are the same. On substituting $\alpha_s = 1/k$ and $\sigma_{jr} = \sigma$ in the expectation step of Equation 9.4, one obtains the following:

$$P(\mathcal{G}_r | \vec{x}_i) = \frac{\exp(-\sum_{j=1}^d [(x_{ij} - \mu_{jr})^2 / (2\sigma^2)])}{\sum_{s=1}^k \exp(-\sum_{j=1}^d [(x_{ij} - \mu_{js})^2 / (2\sigma^2)])}$$

The denominator of the above expression is the same for each cluster r (and fixed point \vec{x}_i), whereas the numerator is maximized when the expression in the exponent is least negative. Therefore, the above assignment probability is maximized for the cluster r for which the value of $\sum_{j=1}^d (x_{ij} - \mu_{jr})^2$ is small as possible for a particular point \vec{x}_i . In other words, *the deterministic expectation step assigns each point \vec{x}_i to the cluster r for which the distance to the cluster centroid is as small as possible, and therefore γ_{ri} is set to 1 for such a cluster-point pair.*

For the maximization step, the weights $\gamma_{ri} \in \{0, 1\}$ are used in the same manner as the algorithm discussed in the previous section. Therefore, the *maximization step sets the centroid of each cluster to the mean of the points assigned to each cluster*. The value of σ^2 does not need to be explicitly calculated for the algorithm to work, and its derivation is slightly different from the previous section (since this parameter is shared across different clusters unlike in the previous section where cluster variances are distinct). Nevertheless, the estimated value of σ can be shown to be equal to the mean-squared distance of the points from their cluster centroids along each dimension:

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^n \sum_{j=1}^d \sum_{r=1}^k \gamma_{ri} (x_{ij} - \hat{\mu}_{jr})^2}{n \cdot d}$$

The deterministic variation of the expectation-maximization algorithm is referred to as the *k-means algorithm*. The estimation of the variance parameter σ^2 is not essential for the implementation of the algorithm, but it can still be useful in some variations of the approach. This estimate is referred^a to as the normalized sum-of-squared distance (SSQ). SSQ is sometimes used as a measure of the quality of the clustering, where smaller values are better. As the algorithm proceeds to convergence, this value stabilizes and provides a termination signal to the iterative approach. ■

^aThe dimension- and point-wise normalization is often omitted by dropping the factor of $n \cdot d$ in the denominator of the expression for σ^2 . Dropping this factor yields the raw sum-of-squared distance measure.

Problem 9.1 (Laplace Distribution for Mixture Modeling) *The Laplace distribution (cf. page 312) is very similar to the Gaussian distribution but uses the normalized L_1 -distance in the exponent. Specifically, a 1-dimensional Laplace distribution is as follows:*

$$f_Z(z) = \frac{1}{2s} \exp\left(-\frac{\|z - \mu\|_1}{s}\right)$$

Rework the details of the expectation-maximization algorithm in this section using a d -dimensional Laplacian distribution, in which each dimension j and mixture component r has its own mean μ_{jr} and scale factor s_{jr} .

Problem 9.2 (Dimension-Normalized k-Means) *Example 9.3 considers a k-means algorithm in which all clusters have the same variance σ along each dimension. Work out a deterministic variant of the EM-algorithm in which the variance σ_{jr}^2 is specific to dimension j and mixture component r . The prior probability α_s is fixed to $1/k$ for each s and points are deterministically assigned to the cluster with the highest posterior probability γ_{ri} of membership. Work out the details of this variant of the k-means algorithm.*

9.2.2 Binary Data: The Bernoulli Distribution

The Bernoulli distribution is used to model binary (0/1) attributes such as implicit feedback data. The Bernoulli distribution is also used to approximately model sparse and nonnegative values. An example is the case of text data in which the presence or absence of words in documents is used to create sparse binary representations. It is also possible to transform any data set containing a mixture of categorical and numeric values to a binary data set. First, the numeric values are transformed to categorical values using the process of *discretization*, wherein each attribute domain is divided into ranges and the attribute value of an observation is assigned to the category corresponding to its range. After the mixed-attribute data set has been transformed to a purely categorical data set, each categorical attribute can be transformed to multiple binary attributes via the process of one-hot encoding (cf. section 7.5 in Chapter 7). Therefore, even though the Bernoulli distribution might seem quite simple at first glance, it is capable of modeling a variety of data sets after preprocessing.

The n data points are denoted by $\vec{x}_1 \dots \vec{x}_i, \dots \vec{x}_n$. The i th data point \vec{x}_i contains d dimensions denoted by $[x_{i1}, x_{i2}, \dots x_{id}]$, all of which are binary values. For example, the set of attributes for a 4-dimensional observation might be $[1, 0, 1, 1]$. The joint probability mass function of the r th mixture component is denoted by $p_{\vec{X}|\mathcal{G}_r}(\vec{x}_i)$. Conditional independence among attributes ensures that the joint probability mass function is the product of the marginal probability mass functions over the different dimensions. Each of these probability mass functions corresponds to a Bernoulli distribution for a particular dimension of the r th mixture component distribution. If the parameter (success probability) of the marginal probability mass function $p_{X_j|\mathcal{G}_r}(\cdot)$ of the Bernoulli distribution for the r th mixture model along the j th dimension is p_{jr} , the probability of the i th d -dimensional observation $\vec{x}_i = [x_{i1}, x_{i2}, \dots x_{id}]$ is obtained using the independence assumption as follows:

$$p_{\vec{X}|\mathcal{G}_r}(\vec{x}_i) = \prod_{j=1}^d p_{X_j|\mathcal{G}_r}(x_{ij}) = \prod_{j=1}^d [p_{jr}^{x_{ij}} (1 - p_{jr})^{(1-x_{ij})}] \quad (9.5)$$

Each Bernoulli distribution in the mixture can be reconstructed by learning its success parameter with the help of the assignment probabilities of the points to the different mixture components. Let γ_{ri} be the assignment probability of point \vec{x}_i to mixture component \mathcal{G}_r in the E-step. Therefore, the value of γ_{ri} is calculated as follows:

$$\gamma_{ri} = P(\mathcal{G}_r|\vec{x}_i) = \frac{P(\mathcal{G}_r) \cdot p_{\vec{X}|\mathcal{G}_r}(\vec{x}_i)}{\sum_{s=1}^k P(\mathcal{G}_s) \cdot p_{\vec{X}|\mathcal{G}_s}(\vec{x}_i)} \quad \forall i, r$$

On substituting the probability mass function of the Bernoulli distribution in the above equation, the following expression is obtained for the weight γ_{ri} :

$$\gamma_{ri} = \frac{\alpha_r \cdot \left[\prod_{j=1}^d [p_{jr}^{x_{ij}} (1 - p_{jr})^{(1-x_{ij})}] \right]}{\sum_{s=1}^k \alpha_s \cdot \left[\prod_{j=1}^d [p_{js}^{x_{ij}} (1 - p_{js})^{(1-x_{ij})}] \right]}$$

The above equation completes the description of the expectation step. It remains to describe the parameter estimation (maximization) step.

As in the case of the Gaussian distribution, the prior probabilities $\alpha_1 \dots \alpha_k$ may be computed as follows:

$$\hat{\alpha}_r = P(\mathcal{G}_r) = \frac{\sum_{i=1}^n \gamma_{ri}}{n} \quad \forall r$$

Furthermore, since the different dimensions are independent of one another in each such joint distribution, the parameters of each dimension can be estimated independently using the statistics of that dimension in the training data (cf. Lemma 6.1 in section 6.3.9 of Chapter 6). Therefore, p_{jr} is estimated as the (weighted) fraction of training examples in the r th mixture component that take on the value of 1 for the j th dimension (cf. section 6.3.2 of Chapter 6). In other words, the value of p_{jr} is estimated as follows:

$$\hat{p}_{jr} = \frac{\sum_{i=1}^n \gamma_{ri} x_{ij}}{\sum_{i=1}^n \gamma_{ri}}$$

The aforementioned relationship provides the maximum likelihood estimate of the Bernoulli distribution parameters. The E-step and M-step are repeated to convergence, and the soft assignments of data points to clusters (i.e., the values of γ_{ri}) are reported as the outputs of the algorithm. The estimated values of p_{jr} provide information about the frequencies of the different attributes in each cluster. For example, in the text domain, the value p_{jr} provides information about how frequently the word j is present in cluster r . This information can be used to identify which topics are relevant to cluster r .

Example 9.4 Discuss how you can add regularization to the Bernoulli model for clustering.

Solution: One can derive some ideas from the regularization of the Bayes classification model with the Bernoulli distribution in Chapter 8. The main problems associated with overfitting arise in the M-step where the prior probabilities of clusters and the Bernoulli success probabilities of die faces are estimated. Changing the M-step to MAP estimation with beta-distribution priors (instead of using MLE estimation) amounts to the use of Laplacian smoothing (cf. section 6.6.2 in Chapter 6).

The smoothed estimation of the prior probabilities is as follows for regularization parameter $\lambda_1 > 0$:

$$\hat{\alpha}_r = \frac{\lambda_1 + \sum_{i=1}^n \gamma_{ri}}{k\lambda_1 + n} \quad \forall r$$

The smoothed estimation of the Bernoulli success probabilities are as follows (for regularization parameter $\lambda_2 > 0$):

$$\hat{p}_{jr} = \frac{\lambda_2 + \sum_{i=1}^n \gamma_{ri} x_{ij}}{2\lambda_2 + \sum_{i=1}^n \gamma_{ri}}$$

Note that large values of λ_1 lead to prior values close to $1/k$, whereas large values of λ_2 lead to Bernoulli success probabilities that are close to 0.5. ■

Problem 9.3 (Categorical Distribution) Extend the clustering approach to the case where the individual mixture components are drawn from the categorical distribution. This model is useful for processing categorical data sets, without having to go through the intermediate step of one-hot encoding.

9.2.3 Sparse Numeric Data: The Multinomial Distribution

The generative process of the multinomial data distribution assumes that the numeric data values are discrete (i.e., natural numbers). However, it is possible to extend the model to

real values with minor modifications. The multinomial distribution is used to model text data. While the Bernoulli distribution assumes a binary representation (depending on the presence or absence of a word in a document), the multinomial distribution allows the use of nonnegative frequencies. The Bernoulli distribution is well suited to short documents, whereas the multinomial distribution is better suited to longer documents.

In the multinomial model, the frequencies of the words in each text document are generated by multiple throws of a biased die, wherein the number of throws is equal to the number of words in the document. The number of faces of the die is equal to the number of words in the vocabulary, and each throw adds one to the frequency of the word corresponding to the die face that shows up. Therefore, multiple throws of the die are required to generate the frequencies of the words in a single document. Here, it is noteworthy that the generative process needs to use the number of words in each document in order to define the number of die throws, which is itself a random variable — however, this particular aspect of the generative process turns out to be inconsequential in the final analysis. The biased die that is used is specific to the mixture component at hand. For example, a mixture component (cluster) that is dominated by sports-related documents will have very different probability values on its faces than a mixture component that is dominated by politics-related documents. The probabilities of the die faces will be biased heavily in favor of sports-related words in the first case, whereas the face probabilities will be heavily biased in favor of politics-related words in the second case.

The n data points to be clustered are denoted by $\vec{x}_1, \dots, \vec{x}_i, \dots, \vec{x}_n$. The i th data point \vec{x}_i contains d dimensions denoted by $[x_{i1}, x_{i2}, \dots, x_{id}]$, all of which are discrete numeric values. The data is typically high dimensional and sparse, corresponding to the fact that most of the values of x_{ij} are zeros. The joint probability mass function of the r th mixture component is denoted by $p_{\vec{X}|\mathcal{G}_r}(\vec{x}_i)$ for the i th data point \vec{x}_i . Each of the probability mass functions corresponds to a multinomial distribution, although one of the parameters (number of die throws) is different for different documents. Therefore, we have an additional random variable L , corresponding to the number of die throws. This random variable has a probability mass function $P_L(l)$, although the specific form of this mass function does not turn out to be consequential in the final analysis.

Aside from the number of die throws, the multinomial distribution is also defined by the parameters corresponding to the probabilities of the different die faces. It is assumed that the probability of the j th die face (i.e., j th word in vocabulary) for the r th mixture model is denoted by p_{jr} . A key point is that the probability of a particular observation \vec{x}_i is obtained by first ensuring that the length $l_i = \sum_{j=1}^d x_{ij}$ of the document is drawn from $p_L(l)$, and then using a multinomial distribution whose length parameter is fixed to l_i . The conditional probability of the i th d -dimensional observation $\vec{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]$ (given the r th mixture component) is obtained by multiplying the probability of the document length l_i with the probability that the observation $[x_{i1}, x_{i2}, \dots, x_{id}]$ is obtained by throwing a die l_i times with face probabilities $p_{i1} \dots p_{id}$:

$$p_{\vec{X}|\mathcal{G}_r}(\vec{x}_i) = p_L(l_i) \cdot p_{[\vec{X}|\mathcal{G}_r, L=l_i]}(\vec{x}_i) \quad (9.6)$$

$$= p_L(l_i) \frac{l_i!}{\prod_{j=1}^d x_{ij}!} \prod_{j=1}^d p_{jr}^{x_{ij}} \quad (9.7)$$

Note that the specific form of the probability mass function $p_L(l_i)$ is not specified, although it is often assumed to be drawn from a Poisson distribution. It turns out that the specific form of the distribution of L does not matter as long as it is independent of the choice of mixture component.

The probability parameters of the multinomial distribution can be estimated in a relatively simple manner from the training data once the expectation step computes the probability of assignment of points to mixture components. Therefore, if γ_{ri} is the probability of assignment of point \vec{x}_i to cluster r , it is computed using the probability mass function of the random variable as follows:

$$\gamma_{ri} = P(\mathcal{G}_r | \vec{x}_i) = \frac{P(\mathcal{G}_r) \cdot p_{\vec{X}|\mathcal{G}_r}(\vec{x}_i)}{\sum_{s=1}^k P(\mathcal{G}_s) \cdot p_{\vec{X}|\mathcal{G}_s}(\vec{x}_i)} \quad \forall i, r$$

On substituting the probability mass function of the multinomial distribution in the above equation, the following expression is obtained for γ_{ri} :

$$\gamma_{ri} = \frac{\alpha_r \cdot \left[\prod_{j=1}^d p_{jr}^{x_{ij}} \right]}{\sum_{s=1}^k \alpha_s \cdot \left[\prod_{j=1}^d p_{js}^{x_{ij}} \right]}$$

The above equation completes the description of the expectation step. Next, we provide a description of the maximization step.

The prior probabilities are estimated in the same manner as in the case of other distributions:

$$\hat{\alpha}_r = P(\mathcal{G}_r) = \frac{\sum_{i=1}^n \gamma_{ri}}{n} \quad \forall r$$

The maximum-likelihood estimation of the parameters of the multinomial distribution is discussed in section 6.3.5 of Chapter 6. One can adapt Equation 6.10 of section 6.3.5 to this setting by performing the estimation separately for the training examples belonging to the r th mixture component. Adapting Equation 6.10 to the notations used in this section (and performing the estimation separately for each mixture component), the probability parameter p_{jr} may be estimated as follows:

$$\hat{p}_{jr} = \frac{\sum_{i=1}^n \gamma_{ri} x_{ij}}{\sum_{i=1}^n \gamma_{ril_i}} \quad (9.8)$$

The aforementioned estimation is identical to Equation 6.10, except that the estimation is performed separately for each mixture component (and the notations are different to accommodate a mixture distribution). Note that the parameters of $p_L(l)$ (document length) do not need to be estimated because the expression for the posterior probability is independent of this value. At the end of the iterative process, each value of γ_{ri} provides the probability of assignment of a data point to a particular mixture component (i.e., posterior generative probability of data point $vec{x}_i$ from mixture component \mathcal{G}_r). The estimated value \hat{p}_{jr} provides the relative frequency of attribute j in mixture component r . For example, in the case of text data, \hat{p}_{jr} provides the relative frequency of word j in cluster r , and can be used to identify the topics relevant to cluster r by examining its frequent words.

Example 9.5 How would you incorporate regularization in clustering with the multinomial model?

Solution: This problem is very similar to the regularization of the Bernoulli model for clustering (cf. Example 9.4). The main problems associated with overfitting arise in the M-step where the prior probabilities of clusters and the multinomial prob-

abilities of die faces are estimated. Changing the M-step to MAP estimation with beta-distribution priors (instead of using MLE estimation) amounts to the use of Laplacian smoothing (cf. page 280 in Chapter 6).

The smoothed estimation of the prior probabilities is as follows for regularization parameter $\lambda_1 > 0$:

$$\hat{\alpha}_r = \frac{\lambda_1 + \sum_{i=1}^n \gamma_{ri}}{k\lambda_1 + n} \quad \forall r$$

The smoothed estimation of the multinomial distribution probabilities are as follows (for regularization parameter $\lambda_2 > 0$):

$$\hat{p}_{jr} = \frac{\lambda_2 + \sum_{i=1}^n \gamma_{ri} x_{ij}}{d\lambda_2 + \sum_{i=1}^n \gamma_{ri} l_i}$$

■

Problem 9.4 (Semi-Supervised Clustering) Consider a situation where some of the data points have labels corresponding to k classes, whereas others are unlabeled. How would you perform k -way clustering in this case? Specifically, how would the E-step and M-step change? Comment on the relationship of this method with both unsupervised clustering and the training process of the naïve Bayes classifier.

The key hint for solving this problem is to understand what the appropriate value of γ_{ri} should be for labeled points. All other steps are similar to unsupervised clustering.

9.3 Matrix Factorization

In the matrix factorization problem, the goal is to approximately factorize a matrix into two small matrices. In the scalar domain, factorization is a relatively simple problem that allows an infinite number of possible solutions. For example, the number 12 can be expressed as $6 * 2$, $4 * 3$, or $8 * 1.5$. Similarly, matrices can be factorized in multiple ways, such as the following:

$$\begin{bmatrix} 3 & 6 \\ 3 & 6 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 3 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix}$$

in the case discussed above, exact factorization of the matrix is possible. However, in the case of larger matrices, exact factorization may not be possible. For example, consider the case where 0.001 is added to the top-left entry of the above matrix to create the value 3.001. In such a case, the matrix becomes invertible, and therefore the 2×2 matrix can no longer¹ be factorized as UV^T , where U and V are both of size 2×1 . Nevertheless, the original factorization continues to be an excellent approximation:

$$\begin{bmatrix} 3.001 & 6 \\ 3 & 6 \end{bmatrix} \approx \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 3 & 6 \end{bmatrix}$$

In general, the approximate factorization of an $n \times d$ matrix D into two smaller matrices U and V of respective sizes $n \times k$ and $d \times k$ is referred to as *rank- k matrix factorization*:

$$D \approx UV^T$$

¹A square invertible matrix cannot be factorized into two matrices of lower rank, since the rank of the product of the matrices is no larger than the rank of both matrices.

Typically, the value of k is significantly smaller than $\min\{n, d\}$, and therefore exact factorization is (often) not possible. Nevertheless, excellent approximations are often possible in many real-world data sets for values of k that are significantly smaller than $\min\{n, d\}$. This is because real-world matrices are not random but contain significant correlations, causing the data to roughly align along low-dimensional hyperplanes. The basis vectors of these hyperplanes correspond to the columns in V , and the rows of U contain the compressed coordinates of the data points with the use of these bases vectors [6]. Such factorizations are also referred to as *low-rank* factorizations. These factorizations are compressed representations of the data because the total number of entries $(n + d)k$ in U and V is typically much smaller than the number $n \cdot d$ of entries in D . For example, a text corpus might be represented by a document-term corpus of size $10^6 \times 10^6$, but the value of k might only be of the order of a few hundred, corresponding to significant space savings.

Why is matrix factorization useful? First, it is used as a tool for dimensionality reduction, and in some cases, the reduction even has semantically interpretable characteristics for some domains. In the case of matrices with missing entries, it is used to create recommender systems. Some forms of matrix factorization are also useful for clustering. In the following, some common types of matrix factorization will be considered together with their probabilistic interpretations. For the purpose of this section, it will be assumed that the (i, j) th entry of the $n \times d$ matrix D is x_{ij} .

9.3.1 The Squared Loss Model

The simplest type of matrix factorization uses the squared-loss model. In this model, the Frobenius norm (sum of squares of the entries) of the residual matrix $(D - UV^T)$ is minimized. Here, D is an $n \times d$ matrix, but U and V are $n \times k$ and $d \times k$ matrices, respectively. Therefore, the loss function of matrix factorization is as follows:

$$\text{Minimize}_{U,V} J = \frac{1}{2} \|D - UV^T\|_F^2$$

One can use gradient-descent to minimize the above loss function. It can be shown that the derivatives of J with respect to U and V in matrix calculus notation are as follows:

$$\begin{aligned}\frac{\partial J}{\partial U} &= -(D - UV^T)V \\ \frac{\partial J}{\partial V} &= -(D - UV^T)^TU\end{aligned}$$

In gradient-descent, each of U and V are updated in the negative direction of the aforementioned gradients with learning rate $\alpha > 0$. Therefore, the updates are as follows:

$$\begin{aligned}U &\leftarrow U - \alpha \frac{\partial J}{\partial U} \\ V &\leftarrow V - \alpha \frac{\partial J}{\partial V}\end{aligned}$$

One can substitute the gradients in the above equation to obtain the following:

$$\begin{aligned}U &\leftarrow U + \alpha(D - UV^T)V \\ V &\leftarrow V + \alpha(D - UV^T)^TU\end{aligned}$$

After initializing the matrices U and V to random values, the aforementioned updates are repeated until convergence is achieved.

The above description presents the updates in matrix notation. It is also possible to present the updates in vector notation using the rows \vec{u}_i and \vec{v}_j of matrices U and V . Let x_{ij} be the (i, j) th entry of the matrix D . Then, $\hat{x}_{ij} = \vec{u}_i \cdot \vec{v}_j$ is the predicted value of the (i, j) th entry of D , and the corresponding residual error $e_{ij} = x_{ij} - \hat{x}_{ij}$ is the difference between the observed and predicted value of the entries. Note that e_{ij} is also the (i, j) th value of the residual matrix $(D - UV^T)$. Instead of the use of the Frobenius norm, the optimization problem may be expressed in scalar form as follows:

$$\begin{aligned}\text{Minimize}_{U, V} J &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^d e_{ij}^2 \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^d (x_{ij} - \vec{u}_i \cdot \vec{v}_j)^2\end{aligned}$$

The derivative of the above objective function yields updates that decompose the aforementioned matrix-wise updates into row-wise updates. A further simplification is to randomly sample a single entry e_{ij} and perform an update that reduces the error of that entry. One can express a simplified form of the aforementioned gradient-descent updates in vector notation as follows:

$$\begin{aligned}\vec{u}_i &\leftarrow \vec{u}_i + \alpha e_{ij} \vec{v}_j \\ \vec{v}_j &\leftarrow \vec{v}_j + \alpha e_{ij} \vec{u}_i\end{aligned}$$

The above form of the update is useful because it can be used for *stochastic* gradient descent in which a single entry of the matrix is used for the update rather than all the entries (as in gradient descent). In stochastic gradient descent, the entries e_{ij} are sampled from the data and the above updates are repeatedly made until convergence is reached. The process of random sampling reduces the complexity of each update without (proportionately) increasing the number of updates, thereby resulting in significantly faster convergence.

9.3.1.1 Probabilistic Interpretation of Squared Loss

The probabilistic interpretation of squared loss in matrix factorization is very similar to that of squared loss in linear regression. In the probabilistic perspective, the optimization formulation can be derived as a maximum-likelihood problem in which the parameters in matrices \vec{u}_i and \vec{v}_j are instantiations of random variables that need to be estimated in a data-driven manner. Each residual entry $e_{ij} = x_{ij} - \vec{u}_i \cdot \vec{v}_j$ is drawn from a normal distribution defined by random variable \mathcal{E} with zero mean and variance σ^2 . In other words, we have the following distribution for the errors:

$$f_{\mathcal{E} | \vec{U}_1 = \vec{u}_1, \dots, \vec{U}_n = \vec{u}_n, \vec{V}_1 = \vec{v}_1, \dots, \vec{V}_d = \vec{v}_d}(e_{ij}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{e_{ij}^2}{2\sigma^2}\right)$$

Here, \vec{U}_i and \vec{V}_j are notations for random vectors corresponding to the i th and j th rows in the two factor matrices, whereas \vec{u}_i and \vec{v}_j are their specific instantiations. Therefore, the likelihood fit $\mathcal{L}([e_{ij}], \vec{u}_1, \dots, \vec{u}_n, \vec{v}_1, \dots, \vec{v}_d, \sigma^2)$ of matrix factorization over all the entries of the matrix is the product of these density functions over all the entries, which yields the

following:

$$\begin{aligned}\mathcal{L}([e_{ij}], \vec{u}_1, \dots, \vec{u}_n, \vec{v}_1, \dots, \vec{v}_d, \sigma^2) &= \prod_{i=1}^n \prod_{j=1}^d \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{e_{ij}^2}{2\sigma^2}\right) \\ &= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{n \cdot d} \exp\left(-\sum_{i=1}^n \sum_{j=1}^d \frac{e_{ij}^2}{2\sigma^2}\right)\end{aligned}$$

The notation $[e_{ij}]$ in the argument (with a square bracket) refers to the fact that all $n \times d$ errors in the residual matrix are part of the argument of the likelihood fit function. On taking the negative logarithm of both sides, we obtain the negative log-likelihood fit as follows:

$$\begin{aligned}\mathcal{L}([e_{ij}], \vec{u}_1, \dots, \vec{u}_n, \vec{v}_1, \dots, \vec{v}_d, \sigma^2) &= \frac{n \cdot d}{2} \cdot \ln(2\pi) + n \cdot d \cdot \ln(\sigma) + \sum_{i=1}^n \sum_{j=1}^d \frac{e_{ij}^2}{2\sigma^2} \\ &= H(\sigma) + \frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{j=1}^d e_{ij}^2\end{aligned}$$

It is evident that no matter what the optimal value of σ might be, the objective function can be minimized only when the expression $\sum_{i=1}^n \sum_{j=1}^d e_{ij}^2$ in the second term is minimized. This objective function is exactly the same as the squared-loss function discussed in the previous section. Therefore, *the maximum-likelihood estimation of the parameters of matrix factorization with Gaussian errors is equivalent to finding parameter values that minimize the sum of squares of the residuals.*

Lemma 9.1 (Probabilistic Interpretation of Squared Loss in Matrix Factorization)

The problem of maximum-likelihood estimation of the parameters of matrix factorization with zero-centered Gaussian errors is equivalent to finding the parameter values that minimize the sum of squares of the residuals.

It remains to determine the maximum-likelihood estimate of the parameter σ . On differentiating the negative log-likelihood with respect to σ and setting it to zero, we obtain the following:

$$\frac{n \cdot d}{\sigma} - \sum_{i=1}^n \sum_{j=1}^d \frac{e_{ij}^2}{\sigma^3} = 0$$

This yields the following maximum-likelihood estimate for σ :

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^n \sum_{j=1}^d e_{ij}^2}{n \cdot d}$$

This is essentially the average value of the squared residual. It is noteworthy that the entire analysis of matrix factorization with squared residuals is very similar to that of squared-loss regression.

Problem 9.5 *Formulate the problem of matrix factorization with an L_1 -loss function instead of the L_2 -loss function, as discussed in this section. Show that the probabilistic model for the residual is the Laplace distribution (cf. page 312). In other words, you need to show that the errors are distributed using a zero-centered Laplace distribution. Derive the gradient-descent steps.*

9.3.1.2 Regularization

It is also possible to impose regularization on the optimization model in order to reduce overfitting. Consider the case where the fully specified $n \times d$ matrix D is factorized into $n \times k$ matrix U and $d \times k$ matrix V as $D \approx UV^T$. Then, the regularized formulation of matrix factorization is as follows:

$$\text{Minimize}_{U,V} J = \frac{1}{2} \|D - UV^T\|_F^2 + \frac{\lambda}{2} \|U\|_F^2 + \frac{\lambda}{2} \|V\|_F^2$$

Here, $\lambda > 0$ is the regularization parameter, which regulates the extent to which large values of the parameters are penalized. The updates for the regularized formulation are different from the unregularized formulation only in the sense that a shrinkage factor of $(1 - \alpha\lambda)$ is applied to the regularization parameters, where α is the learning rate. Therefore, the updates for the regularized variant of the problem are as follows:

$$\begin{aligned} U &\leftarrow U(1 - \alpha\lambda) + \alpha(D - UV^T)V \\ V &\leftarrow V(1 - \alpha\lambda) + \alpha(D - UV^T)^TU \end{aligned}$$

The regularized formulation of matrix factorization essentially imposes a Gaussian prior on the parameters in U and V for the purposes of optimization.

Example 9.6 Show that if a Gaussian prior is imposed on the parameters in U and V , the use of MAP estimation is equivalent to adding an L_2 -regularization term to the optimization model of MLE. Assume that regularization is used with L_2 -loss.

Solution: Since the prior distribution of the parameters is a Gaussian distribution, it may be expressed as follows:

$$f_{\vec{U}_1 \dots \vec{U}_n, \vec{V}_1 \dots \vec{V}_n}(\vec{u}_1 \dots \vec{u}_n, \vec{v}_1 \dots \vec{v}_n) \propto \exp\left(-\frac{\sum_{i=1}^n \|\vec{u}_i\|^2 + \sum_{j=1}^d \|\vec{v}_j\|^2}{2\sigma_0^2}\right)$$

Here, \vec{U}_i and \vec{V}_j are notations for random vectors corresponding to the i th and j th rows in the two factor matrices, whereas \vec{u}_i and \vec{v}_j are their specific instantiations. Furthermore, the hyper-parameter σ_0 is selected by the user, and it controls the degree of regularization. Small values of σ_0 will cause the magnitudes of U and V to be small because the Gaussian prior is more concentrated near the origin. Based on Equation 6.20 of Chapter 6, an MLE model can be converted to a MAP model by adding a term that is the negative logarithm of the prior (ignoring constants). The negative logarithm of the aforementioned Gaussian prior is as follows:

$$-\ln(f_{\vec{U}_1 \dots \vec{U}_n, \vec{V}_1 \dots \vec{V}_n}(\vec{u}_1 \dots \vec{u}_n, \vec{v}_1 \dots \vec{v}_n)) = \text{Constant} + \frac{\sum_{i=1}^n \|\vec{u}_i\|^2 + \sum_{j=1}^d \|\vec{v}_j\|^2}{2\sigma_0^2}$$

The constant (which accounts for the proportionality factor) can be ignored because it does not affect the optimization process. Furthermore, by expressing the regularization parameter in terms of using $\lambda \propto 1/\sigma_0^2$, one obtains an additive term that is identical to squared penalty-based regularization. ■

Problem 9.6 Suppose that you perform L_1 -regularization on the parameters. What type of prior on the parameters in U and V does L_1 -regularization correspond to? Show how to derive L_1 -regularization using MAP estimation.

9.3.1.3 Application to Incomplete Data: Recommender Systems

A useful application of matrix factorization is its application to filling in missing ratings in *collaborative filtering*. A collaborative filtering system is based on an $n \times d$ matrix D of ratings. The (i, j) th entry x_{ij} corresponds to the rating of user i for item j . The main challenge in the case of recommender systems is that the vast majority of the entries in the matrix D are unspecified. Therefore, the factorization $D \approx UV^T$ needs to be performed only with the observed entries, which is a slightly different setting from the model discussed in the previous section. Furthermore, once the matrices U and V have been learned, they can be used to reconstruct the entire matrix as UV^T .

The probabilistic model for the entries of the $n \times d$ matrix D is identical to the case in which all entries are specified. In other words, each error is assumed to be drawn from a Gaussian distribution with zero mean. The main difference is that only a subset of the entries is observed. It can be shown (cf. Problem 9.7) that the resulting maximum-likelihood estimation problem is equivalent to a similar loss function, except that it is defined only over the observed entries.

Consider the case in which the set O defines the indices of the observed entries:

$$O = \{(i, j) : x_{ij} \text{ is observed}\}$$

Let $e_{ij} = x_{ij} - \vec{u}_i \cdot \vec{v}_j$ be the error of the observed entries (where \vec{u}_i and \vec{v}_j represent the correspondingly indexed rows in U and V , respectively). Based on the set of observed entries in O , the optimization problem may be recast as follows:

$$\begin{aligned} \text{Minimize}_{U, V} J &= \frac{1}{2} \sum_{(i, j) \in O} e_{ij}^2 \\ &= \frac{1}{2} \sum_{(i, j) \in O} (x_{ij} - \vec{u}_i \cdot \vec{v}_j)^2 \end{aligned}$$

Note that this objective function is different from that of vanilla matrix factorization only in the sense that observed entries are used. Interestingly, the use of an incomplete matrix does not change the form of the updates to stochastic gradient descent, because the updates are only performed in entry-wise fashion. In other words, the updates remain the same, except that only the observed entries from O are sampled for updates:

$$\begin{aligned} \vec{u}_i &\leftarrow \vec{u}_i + \alpha e_{ij} \vec{v}_j \\ \vec{v}_j &\leftarrow \vec{v}_j + \alpha e_{ij} \vec{u}_i \end{aligned}$$

The entries $(i, j) \in O$ are sampled repeatedly and the aforementioned updates are performed until convergence is reached. It is also possible to add regularization with parameter $\lambda > 0$ to the updates as follows:

$$\begin{aligned} \vec{u}_i &\leftarrow \vec{u}_i (1 - \alpha \lambda) + \alpha e_{ij} \vec{v}_j \\ \vec{v}_j &\leftarrow \vec{v}_j (1 - \alpha \lambda) + \alpha e_{ij} \vec{u}_i \end{aligned}$$

Example 9.7 (Toy Example of Matrix Completion) Consider the following matrix $A = [a_{ij}]$ with significant cross-row correlations:

$$A = \begin{bmatrix} 6 & 3 & 9 \\ 2 & 1 & 4 \\ 2 & 8 & 6 \\ 2 & 4 & 3 \end{bmatrix}$$

Visually examine the matrix and complete it in a manner that is consistent with the patterns in the data. Now justify this completion by factorizing this matrix into 4×1 and 1×4 matrices.

Solution: In this case, it is evident that rows are multiples of one another (as are the columns). Using this fact, the completed matrix is as follows:

$$\begin{bmatrix} 6 & 3 & 12 & 9 \\ 2 & 1 & 4 & 3 \\ 4 & 2 & 8 & 6 \\ 2 & 1 & 4 & 3 \end{bmatrix}$$

Matrices in which rows/columns are multiples of one another can be factorized into the product of $d \times 1$ and $1 \times d$ matrices, where the entries contain the (proportional) values of the multiples. This can be verified by factorizing the matrix as follows:

$$\begin{bmatrix} 6 & 3 & 9 \\ 2 & 1 & 4 \\ 2 & 8 & 6 \\ 2 & 4 & 3 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 2 \\ 1 \end{bmatrix} [2 \quad 1 \quad 4 \quad 3] = \begin{bmatrix} 6 & 3 & 12 & 9 \\ 2 & 1 & 4 & 3 \\ 4 & 2 & 8 & 6 \\ 2 & 1 & 4 & 3 \end{bmatrix}$$

In this toy example, a large percentage of the entries are specified. For larger matrices, a much smaller percentage of entries need to be specified in order to complete it. ■

Problem 9.7 Show that the maximum-likelihood estimate of matrix factorization with incomplete entries and zero-centered Gaussian errors on the observed entries reduces to the problem of minimizing the squared errors on the residuals of the observed entries.

9.3.2 Probabilistic Latent Semantic Analysis

Probabilistic latent semantic analysis is a generative model for matrix factorization, and it is widely used in the text domain. It is assumed that the $n \times d$ matrix D contains nonnegative frequencies of words, where n is the number of documents and d is the number of words. It is assumed that the (i, j) th entry of D is x_{ij} . Throughout this exposition, the matrix D will be referred to as a document-term matrix.

Probabilistic latent semantic analysis creates a normalized three-way factorization of the document-term matrix of the following form:

$$D \propto U \Sigma V^T \tag{9.9}$$

Here, U is an $n \times k$ matrix, Σ is a $k \times k$ diagonal matrix, and V is a $d \times k$ matrix. Furthermore, each of the columns of U and V sum to 1, the entries in Σ sum to 1, and

the individual entries are interpreted as probabilities. The use of proportionality (instead of equality) in Equation 9.9 is necessitated by the strict probability-centric scaling of U , V , and Σ , although scaling down the entries of D to sum to 1 yields an equality relationship. The matrices U , V , and Σ define the parameters of a generative process that is used to create the observed matrix D . These parameters are learned in order to maximize the likelihood of the observed data for this generative process. This generative process uses a mixture model like clustering. However, in clustering, each step of the generative process creates a single row. In this case, each step of the generative process adds 1 to a single entry in the matrix D .

The basic idea is to assume that the frequencies in the document-term matrix are generated by sequentially incrementing entries of the document-term matrix, and the choice of the entry to increment is regulated by first selecting a mixture component from $\mathcal{G}_1 \dots \mathcal{G}_k$. These mixture components are *hidden variables*, also known as *latent variables*, because they are not observed in the data, but have an explanatory role in modeling the data. A mixture component is also referred to as an *aspect* or *topic*, which leads to it being considered a *topic modeling* method. Therefore, if a given mixture component is selected, it is likely to increment topic-relevant entries. As we will see later, the number of mixture components k defines the rank of the factorization. The basic generative process may be described in terms of repeatedly selecting a position from the document-term matrix and incrementing its frequency:

1. Select a mixture component (topic) \mathcal{G}_r with probability $\alpha_r = \Sigma_{rr}$, where $r \in \{1 \dots k\}$.
2. Select the index i of a document \vec{x}_i with probability $U_{ir} = P(\vec{x}_i | \mathcal{G}_r)$ and the index j of a term t_j with probability $V_{jr} = P(t_j | \mathcal{G}_r)$. It is assumed that the two selections are conditionally independent. Increment the (i, j) th entry x_{ij} of D by 1.

The generative process of incrementing matrix entries will need to be repeated as many times as the number of *tokens* in the corpus (including document-specific repetitions of term occurrences). A plate diagram of this generative process is shown in Figure 9.2(a).

One must formulate an optimization problem that minimizes the negative log-likelihood of the document-term matrix being generated by this model. In other words, the optimization problem for PLSA may be stated as follows:

Minimize _{(U, V, Σ)} – [Log likelihood of generating D using parameters in matrices (P, Q, Σ)]

$$\begin{aligned}
 &= -\log \left(\prod_{i,j} P(\text{Adding one occurrence of term } j \text{ in document } i)^{D_{ij}} \right) \\
 &= -\sum_{i=1}^n \sum_{j=1}^d D_{ij} \underbrace{\log(P(\vec{x}_i, t_j))}_{\text{Parametrized by } U, V, \Sigma}
 \end{aligned}$$

subject to:

$$P, Q, \Sigma \geq 0$$

Entries in each column of U sum to 1

Entries in each column of V sum to 1

Σ is a diagonal matrix that sums to 1

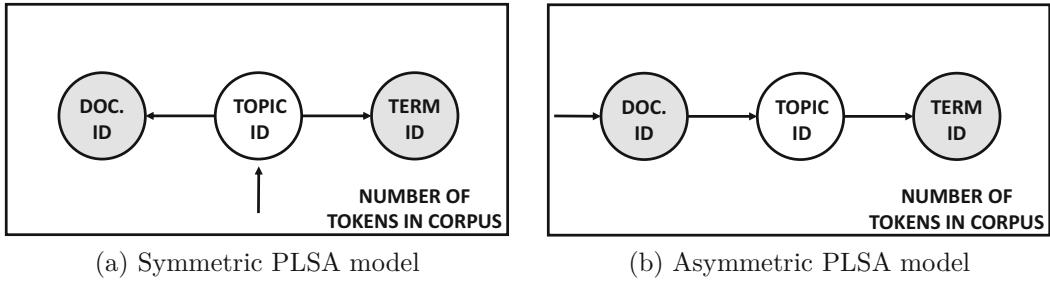


Figure 9.2: Examples of plate diagrams and two equivalent generative models for PLSA

A key point here is that the entries in U , V , and Σ are interpreted as probabilities and the generative process creates the observed matrix D on this basis. This is the reason for the normalization constraints on U , V , and Σ .

The conditional probability $P(\vec{x}_i, t_j | \mathcal{G}_r)$ of selecting a particular document-term pair (\vec{x}_i, t_j) in the generative process follows the conditional independence assumption:

$$P(\vec{x}_i, t_j | \mathcal{G}_r) = P(\vec{x}_i | \mathcal{G}_r) \cdot P(t_j | \mathcal{G}_r) \quad (9.10)$$

The main challenge in solving this optimization problem is that we do not know which mixture component generated which token. The problem would have been easy to solve, had there been only one mixture component (i.e., $k = 1$). As in the case of all expectation-maximization procedures, we need to *simultaneously* compute the mixture memberships and optimization parameters. Specifically, EM optimizes parameters and probabilistic assignments alternately in iterative fashion. The algorithm starts with random nonnegative parameters in U , Σ , and V , which are normalized² so that they can be interpreted as probabilities. In the E-step, we compute the posterior probability $P(\mathcal{G}_r | \vec{x}_i, t_j)$ that each observed document-term pair (\vec{x}_i, t_j) (i.e., token) was generated by a particular mixture component. Therefore, the E-step determines memberships in expectation. These probabilities are treated as “membership weights” of that token for the various mixture components. The M-step uses these membership weights to compute the maximum-likelihood values of all parameters in each mixture component. The M-step is really solving a simplified optimization problem in which the membership weights of the tokens for various mixture components have been fixed. The specific details of the E- and M-steps are as follows:

- 1. (E-step):** Estimate the posterior probabilities $P(\mathcal{G}_r | \vec{x}_i, t_j)$ for each document-term pair (\vec{x}_i, t_j) occurring in the corpus. The Bayes rule is used with the current state of the parameters:

$$P(\mathcal{G}_r | \vec{x}_i, t_j) = \frac{P(\mathcal{G}_r) \cdot P(\vec{x}_i | \mathcal{G}_r) \cdot P(t_j | \mathcal{G}_r)}{\sum_{s=1}^k P(\mathcal{G}_s) \cdot P(\vec{x}_i | \mathcal{G}_s) \cdot P(t_j | \mathcal{G}_s)} = \frac{(\Sigma_{rr}) \cdot (U_{ir}) \cdot (V_{jr})}{\sum_{s=1}^k (\Sigma_{ss}) \cdot (U_{is}) \cdot (V_{js})} \quad \forall i, j, r \quad (9.11)$$

- 2. (M-step):** Estimate the current parameters in U , V and Σ by using the conditional probabilities in the first step as weights for entries belonging to each generative com-

²In other words, the columns of U , the columns of V , and the diagonal of Σ each sum to 1.

ponent. This is achieved as follows:

$$\begin{aligned} U_{ir} &= P(\vec{x}_i | \mathcal{G}_r) = \frac{\sum_j P(\vec{x}_i, t_j) \cdot P(\mathcal{G}_r | \vec{x}_i, t_j)}{P(\mathcal{G}_r)} \propto \sum_j D_{ij} P(\mathcal{G}_r | \vec{x}_i, t_j) \quad \forall i, r \\ V_{jr} &= P(t_j | \mathcal{G}_r) = \frac{\sum_i P(\vec{x}_i, t_j) \cdot P(\mathcal{G}_r | \vec{x}_i, t_j)}{P(\mathcal{G}_r)} \propto \sum_i D_{ij} P(\mathcal{G}_r | \vec{x}_i, t_j) \quad \forall j, r \\ \Sigma_{rr} &= P(\mathcal{G}_r) = \sum_{i,j} P(\vec{x}_i, t_j) \cdot P(\mathcal{G}_r | \vec{x}_i, t_j) \propto \sum_{i,j} D_{ij} P(\mathcal{G}_r | \vec{x}_i, t_j) \quad \forall r \end{aligned}$$

The constants of proportionality are set by ensuring that the probabilities in the columns of U , V , and the diagonal of Σ each sum to 1.

As in all applications of the expectation-maximization algorithm, these steps are iterated to convergence. Convergence can be checked by computing the likelihood function at the end of each iteration, and checking if it has improved by a minimum amount over its average value in the last few iterations.

Why can we express the estimated parameters in the factorized form of $D \propto Q\Sigma P^T$? The reasoning for this follows directly from the probabilistic interpretation of the parameters:

$$\begin{aligned} D_{ij} &\propto P(\vec{x}_i, t_j) = \sum_{r=1}^k \underbrace{P(\mathcal{G}_r)}_{\text{Select } r} \cdot \underbrace{P(\vec{x}_i, t_j | \mathcal{G}_r)}_{\text{Select } \vec{x}_i, t_j} \quad [\text{Generative probability of incrementing } (i, j)] \\ &= \sum_{r=1}^k P(\mathcal{G}_r) \cdot P(\vec{x}_i | \mathcal{G}_r) \cdot P(t_j | \mathcal{G}_r) \quad [\text{Conditional independence}] \\ &= \sum_{r=1}^k P(\vec{x}_i | \mathcal{G}_r) \cdot P(\mathcal{G}_r) \cdot P(t_j | \mathcal{G}_r) \quad [\text{Rearranging product}] \\ &= \sum_{r=1}^k Q_{ir} \cdot \Sigma_{rr} \cdot P_{jr} = (U\Sigma V^T)_{ij} \quad [\text{The factorized form}] \end{aligned}$$

PLSA is a form of *nonnegative matrix factorization* in which we are optimizing a maximum likelihood model rather than the Frobenius norm. It is also common to perform nonnegative matrix factorization with squared loss using the Frobenius norm of the residual matrix. The resulting solution can be used to provide a similar (but not the same) factorization as PLSA.

Example 9.8 (Nonnegative Matrix Factorization) Formulate an optimization model of approximately factorizing D into UV^T , where U and V are constrained to be nonnegative. Propose a modified gradient descent procedure using (almost) the same updates as unconstrained matrix factorization. How would you convert this two-way factorization into a three-way factorization like PLSA?

Solution: The loss function minimizes the Frobenius norm $\|D - UV^T\|_F^2$ subject to the constraints $U, V \geq 0$. The updates are still the same as unconstrained matrix factorization, except that the negative entries of U and V are set to 0 after every update. Note that this is not the optimal approach for gradient descent in non-negative matrix factorization but it is the closest to the unconstrained update method

discussed earlier. In practice, a multiplicative update method based on Lagrangian relaxation is more common [6].

In order to convert the two-way factorization to three-way factorization $Q\Sigma P^T$, the i th column of U is scaled by the sum μ_i of its entries to convert them to probabilities and create Q . The i th column of V is scaled by the sum ν_i of its entries and create P . The i th diagonal entry of Σ is the product $\mu_i\nu_i$. In PLSA, the entries of D are scaled up front to sum to 1. If this type of scaling of D is done up front, it can be shown that the diagonal entries of Σ will sum to 1 (like probabilities). ■

9.3.2.1 Example of PLSA

Consider a toy example of a document-term matrix shown below and develop its PLSA-based factorization:

$$D = \begin{pmatrix} & \text{lion} & \text{tiger} & \text{cheetah} & \text{jaguar} & \text{porsche} & \text{ferrari} \\ \text{Document-1} & 2 & 2 & 1 & 2 & 0 & 0 \\ \text{Document-2} & 2 & 3 & 3 & 3 & 0 & 0 \\ \text{Document-3} & 1 & 1 & 1 & 1 & 0 & 0 \\ \text{Document-4} & 2 & 2 & 2 & 3 & 1 & 1 \\ \text{Document-5} & 0 & 0 & 0 & 1 & 1 & 1 \\ \text{Document-6} & 0 & 0 & 0 & 2 & 1 & 2 \end{pmatrix}$$

This matrix represents topics related to both cars and cats. The first three documents are primarily related to cats, the fourth is related to both, and the last two are primarily related to cars. The word “jaguar” is polysemous because it could correspond to either a car or a cat and is present in documents of both topics.

A possible factorization is shown in Figure 9.3. We have shown an approximate decomposition (with simple fractions) for simplicity, although the optimal solution would (almost always) be dominated by long floating point numbers in practice. It is clear that the first latent concept is related to cats and the second latent concept is related to cars. Furthermore, documents are represented by two non-negative coordinates indicating their affinity to the two topics. Correspondingly, the first three documents have strong positive coordinates for cats, the fourth has strong positive coordinates in both, and the last two belong only to cars. The matrix V tells us that the vocabularies of the various topics are as follows:

Cats: lion, tiger, cheetah, jaguar

Cars: jaguar, porsche, ferrari

It is noteworthy that the polysemous word “jaguar” is included in the vocabulary of both topics, and its usage is automatically inferred from its context (i.e., other words in document) during the factorization process.

The 3-way factorization of Figure 9.3 is defined in terms of a proportionality relationship because all matrices on the left-hand side and right-hand side are scaled to sum to 1. It is also possible to distribute the scaling factors in the diagonal entries to the corresponding columns in U and V in order to create a 2-way factorization. An example of such a 2-way factorization is shown in Figure 9.4. Note that this factorization is not unique within scaling factors — one can multiply the first column of U by 2 and divide the first column of V by 2 to obtain the same matrix product. The 2-way factorization is popularly referred to as nonnegative matrix factorization, although it popularly uses the Frobenius norm (which is

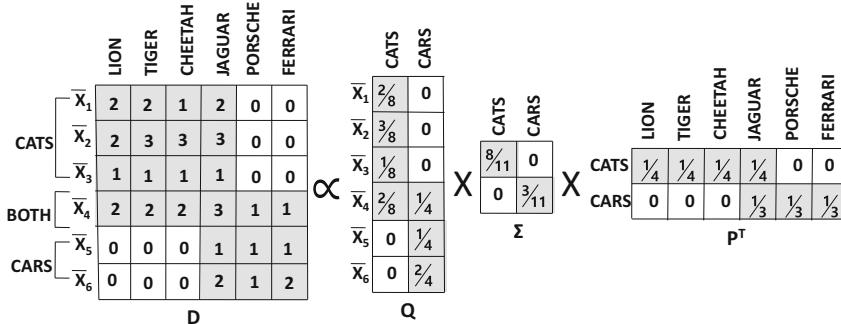


Figure 9.3: Example of PLSA (proportional 3-way factorization)

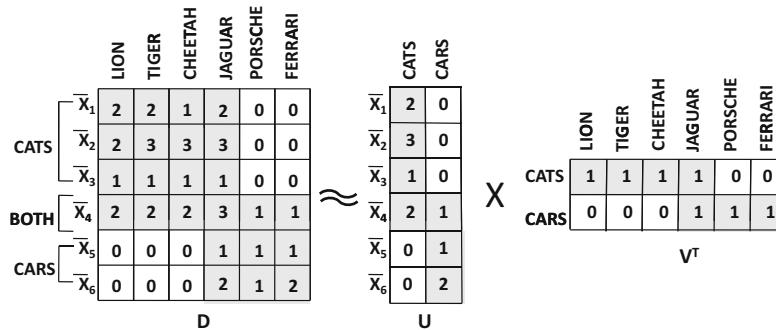


Figure 9.4: Equivalent 2-way factorization of PLSA in Figure 9.3

a different type of maximum-likelihood estimation than the one discussed in the previous section).

The three-way factorization also tells us a little bit more about the relative frequencies of the two topics. Since the diagonal entry in Σ is 32 for cats in comparison with 12 for cars, it indicates that the topic of cats is more dominant than cars. This is consistent with the observation that more documents and terms in the collection are associated with cats as compared to cars. The main advantage of PLSA and nonnegative matrix factorization is its high level of interpretability as compared to other factorizations.

9.3.2.2 Alternative Plate Diagram for PLSA

PLSA independently generates the different *tokens* of the document-term matrix rather than generating one *document* at a time (as is common with most mixture models in clustering). This ensures that topics (mixture components) are associated with matrix entries rather than individual documents. However, it is possible to perform the generative process of PLSA in a different order, so that all the matrix entries corresponding to a single document are updated contiguously even if these updates occur from different latent components. Using this approach has some benefits, as it can be used in order to control the proportional representation of the topics in a single document by adding some additional steps to the generative process. Indeed, PLSA has been extended in this way in order to perform *Latent Dirichlet Allocation* [12], although a detailed discussion of this work is beyond the scope of this book. Latent Dirichlet allocation solves this problem by deciding the composition of

topics in a document up front with the Dirichlet distribution, and then generating all the entries in a row of the document-term matrix in one shot. Therefore, a *prior* structure is imposed on each document with the Dirichlet distribution, and the learning process becomes one of maximum a posteriori estimation.

In order to generate a document contiguously, a slightly different *asymmetric* generative process of PLSA. The asymmetric generative process of PLSA is as follows:

1. Select the i th document, \vec{x}_i , with probability $P(\vec{x}_i) = \sum_s P(\mathcal{G}_s)P(\vec{x}_i|\mathcal{G}_s) = \sum_s (\Sigma_{ss})(U_{is})$.
2. Select the topic r with probability $P(\mathcal{G}_r|\vec{x}_i) = \frac{P(\mathcal{G}_r \cap \vec{x}_i)}{P(\vec{x}_i)} = \frac{(\Sigma_{rr})(U_{ir})}{\sum_s (\Sigma_{ss})(U_{is})}$.
3. Select the j th term, t_j , with probability $P(t_j|\mathcal{G}_r) = V_{jr}$.

Once the document-term pair has been selected, the corresponding entry in the document-term matrix is incremented by 1. The plate diagram for this asymmetric model is shown in Figure 9.2(b).

9.3.3 Logistic Matrix Factorization

Logistic matrix factorization uses a nonlinear prediction function that can be interpreted as a probabilistic model. Let U and V be the $n \times k$ and $d \times k$ factor matrices for the original $n \times d$ matrix $Q = [q_{ij}]$ of sparse frequency counts. In other words, most entries of Q have zero values (which is the sparsity criterion), and all values are nonnegative (since frequency counts are assumed to be nonnegative). Therefore, logistic matrix factorization is also intended for nonnegative matrices (like PLSA) but with the additional criterion of sparsity.

Logistic matrix factorization applies the logistic sigmoid function $F(x)$ to each entry of UV^T in order to generate a probability matrix P :

$$P = F(UV^T)$$

Here, the function $F(\cdot)$ is applied in an entry-wise fashion, and is defined as follows:

$$F(x) = \frac{1}{1 + \exp(-x)}$$

Each entry of the $n \times d$ matrix P is a probability value drawn from $(0, 1)$, and the goal is to maximize the log-likelihood (or minimize the negative log-likelihood) of the observed data matrix based on these probabilities. A binarized matrix $D = [x_{ij}]$ is created from the matrix Q as follows:

$$x_{ij} = \begin{cases} 1 & q_{ij} > 0 \\ 0 & q_{ij} = 0 \end{cases}$$

Since logistic matrix factorization is designed for sparse data, the zero entries are weighted differently from the nonzero entries. If this is not done, the errors on the zero entries will overwhelm the errors on the nonzero entries, and all entries in U and V will be too close to 0. The zero entries are also referred to as negative entries and the nonzero entries are also referred to as positive entries. We utilize a user-driven parameter m , which is the ratio of the *aggregate* weight of the zero entries to the aggregate weight of the positive entries.

This parameter therefore controls the relative importance of the positive entries and zero entries. Specifically, the weight w_{ij} of each entry of the matrix is defined as follows:

$$w_{ij} = \begin{cases} q_{ij} & q_{ij} > 0 \\ m(\sum_{s=1}^d q_{is})/d & q_{ij} = 0 \end{cases} \quad (9.12)$$

The value of m is set in a domain-specific way, and it is often a small integer such as 5. Implicitly, the negative entries are underweighted by this approach, because a sparse matrix will often contain negative entries that are hundreds of times the number of positive entries, whereas m is a small value such as 5.

In logistic matrix factorization, we would like the (learned) probability matrix $P = p_{ij}$ to have a large value of p_{ij} when x_{ij} is 1, and a small value of p_{ij} when x_{ij} is 0. This can be achieved with a likelihood objective function, and we will first define the likelihood objective function without the use of the weights w_{ij} . Then, we will define the heuristic approximation of this objective function with the use of weights. The likelihood objective function without the use of weights is defined as follows:

$$\text{Maximize}_{U,V} \mathcal{L} = \prod_{i=1}^n \prod_{j=1}^d p_{ij}^{x_{ij}} (1 - p_{ij})^{(1-x_{ij})}$$

It is noteworthy that the objective function above is a function of U and V , because p_{ij} is defined by applying the logistic function to the entries of UV^T . The above expression essentially computes the product of the probabilities that the entries (i, j) are zero or nonzero, respectively. The negative log-likelihood of the above expression yields the following objective function:

$$\text{Minimize}_{U,V} \mathcal{LL} = - \sum_{i=1}^n \sum_{j=1}^d [x_{ij} \log(p_{ij}) + (1 - x_{ij}) \log(1 - p_{ij})]$$

At this point, one can incorporate the heuristic weight w_{ij} in order to create an objective function in which positive and zero entries are weighted differently. Note that the weights w_{ij} are set by Equation 9.12, which uses different rules to set the weights of positive and zero entries. The incorporation of weights results in the following objective function:

$$\text{Minimize}_{U,V} J = - \sum_{i=1}^n \sum_{j=1}^d w_{ij} [x_{ij} \log(p_{ij}) + (1 - x_{ij}) \log(1 - p_{ij})]$$

It is evident that this Recall that each p_{ij} is the (i, j) th entry of $F(UV^T)$, and it is defined as follows:

$$p_{ij} = \frac{1}{1 + \exp(-\vec{u}_i \cdot \vec{v}_j)}$$

Here, \vec{u}_i is the i th row of the $n \times k$ matrix U , and \vec{v}_j is the j th row of the $d \times k$ matrix V . Therefore, one can substitute this value of p_{ij} in the objective function in order to obtain the following loss function for logistic matrix factorization:

$$J = - \sum_{i=1}^n \sum_{j=1}^d w_{ij} \left[x_{ij} \log \left(\frac{1}{1 + \exp(-\vec{u}_i \cdot \vec{v}_j)} \right) + (1 - x_{ij}) \log \left(\frac{1}{1 + \exp(\vec{u}_i \cdot \vec{v}_j)} \right) \right]$$

Now that we have set up the objective function of logistic matrix factorization, it remains to derive the gradient descent steps.

9.3.4 Gradient Descent Steps for Logistic Matrix Factorization

In order to perform the gradient-descent updates, we need to compute the gradient of the objective function with respect to the k -dimensional vectors \vec{u}_i and \vec{v}_j . This is best achieved using the chain rule in matrix calculus:

$$\begin{aligned}\frac{\partial J}{\partial \vec{u}_i} &= \sum_{j=1}^d \frac{\partial J}{\partial (\vec{u}_i \cdot \vec{v}_j)} \frac{\partial (\vec{u}_i \cdot \vec{v}_j)}{\partial \vec{u}_i} = \sum_{j=1}^d \frac{\partial J}{\partial (\vec{u}_i \cdot \vec{v}_j)} \vec{v}_j \\ \frac{\partial J}{\partial \vec{v}_j} &= \sum_{i=1}^n \frac{\partial J}{\partial (\vec{u}_i \cdot \vec{v}_j)} \frac{\partial (\vec{u}_i \cdot \vec{v}_j)}{\partial \vec{v}_j} = \sum_{i=1}^n \frac{\partial J}{\partial (\vec{u}_i \cdot \vec{v}_j)} \vec{u}_i\end{aligned}$$

Note that the partial derivative of $\vec{u}_i \cdot \vec{v}_j$ with respect to \vec{u}_i is \vec{v}_j , and that with respect to \vec{v}_j is \vec{u}_i . It is relatively easy to compute the partial derivative of J with respect to $\vec{u}_i \cdot \vec{v}_j$ because the objective function is defined as a function of this quantity. By computing this derivative and substituting in the above equations, we obtain the following:

$$\begin{aligned}\frac{\partial J}{\partial \vec{u}_i} &= - \sum_{j=1}^d \frac{w_{ij} x_{ij} \vec{v}_j}{1 + \exp(\vec{u}_i \cdot \vec{v}_j)} + \sum_{j=1}^d \frac{w_{ij}(1 - x_{ij}) \vec{v}_j}{1 + \exp(-\vec{u}_i \cdot \vec{v}_j)} \\ \frac{\partial J}{\partial \vec{v}_j} &= - \sum_{i=1}^n \frac{w_{ij} x_{ij} \vec{u}_i}{1 + \exp(\vec{u}_i \cdot \vec{v}_j)} + \sum_{i=1}^n \frac{w_{ij}(1 - x_{ij}) \vec{u}_i}{1 + \exp(-\vec{u}_i \cdot \vec{v}_j)}\end{aligned}$$

With these derivatives, a straightforward gradient-descent procedure can be applied at learning rate $\alpha > 0$:

$$\begin{aligned}\vec{u}_i &\leftarrow \vec{u}_i - \alpha \frac{\partial J}{\partial \vec{u}_i} \quad \forall i \\ \vec{v}_j &\leftarrow \vec{v}_j - \alpha \frac{\partial J}{\partial \vec{v}_j} \quad \forall j\end{aligned}$$

It is also possible to generalize the approach to stochastic gradient descent.

Problem 9.8 Derive the additional L_2 -regularization term for logistic matrix factorization based on MAP estimation principles. Derive the gradient-descent steps.

You may find it helpful to visit Example 9.6 for the above problem.

9.4 Outlier Detection

Outliers are naturally defined probabilistically based on Hawkins' definition of outliers:

“An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism.”

Note that the word “mechanism” can be naturally interpreted as a “generating mechanism,” such as a mixture model. Therefore, if the data set is modeled in the form of a mixture of Gaussians, an outlier is a point whose likelihood fit is low. Therefore, the likelihood fit (or its negative logarithm) provides an outlier score. If the negative logarithm is applied to the outlier score, then larger scores are indicative of a greater degree of outlierness. In the following, we will provide several algorithms that are based on this principle.

9.4.1 The Mahalanobis Method: A Probabilistic View of Whitening

The use of whitening for data preprocessing and outlier detection is discussed in section 2.4.2 of Chapter 2. This section will provide a second exposition of the Mahalanobis method, except that a probabilistic view will be provided. As we will see in this section, the whitening approach implicitly models the data to belong to a single Gaussian distribution.

Consider a data set with points denoted by $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$. The components of the vector \vec{x}_i are denoted by $[x_{i1} \dots x_{id}]$. The multivariate Gaussian distribution assumes that all points are drawn from the following Gaussian distribution (cf. Equation 4.3 of Chapter 4):

$$f_{\vec{X}}(\vec{x}) = \frac{1}{(2\pi)^{d/2}|C|^{1/2}} \exp\left(-\frac{[\vec{x} - \vec{\mu}]C^{-1}[\vec{x} - \vec{\mu}]^T}{2}\right) \quad (9.13)$$

Here, the vector $\vec{\mu} = [\mu_1, \dots, \mu_d]$ is the d -dimensional distribution mean and $C = [c_{ij}]$ is a $d \times d$ matrix containing the pairwise distribution covariances. As discussed in Chapter 6, the maximum-likelihood estimate of the distribution mean $\vec{\mu}$ is the sample mean:

$$\hat{\vec{\mu}} = \frac{\sum_{i=1}^n \vec{x}_i}{n}$$

Furthermore, the (i, j) th element c_{ij} of the covariance matrix C can be estimated as the sample covariance between the dimensions i and j :

$$\hat{c}_{ij}^2 = \frac{\sum_{k=1}^n (x_{ki} - \mu_i)(x_{kj} - \mu_j)}{n}$$

Note that the covariance matrix is estimated from samples without the Bessel correction, because it contains n in the denominator rather than $(n - 1)$. Therefore, the fit of the data point \vec{x}_i is obtained by substituting it in the density function for the Gaussian:

$$Fit(\vec{x}_i) = \frac{1}{(2\pi)^{d/2}|C|^{1/2}} \exp\left(-\frac{[\vec{x}_i - \vec{\mu}]C^{-1}[\vec{x}_i - \vec{\mu}]^T}{2}\right) \quad (9.14)$$

This fit value³ provides an outlier score for the data point, where lower values of the scores are indicative of greater outliers. Therefore, all points whose fits is below a particular threshold are designated as outliers. One problem with this approach is that the exponentiation function can cause the fit values to be small enough to cause numerical errors. However, since one only needs to know the *ordering* of the points in terms of their outlier tendency, it is possible to use the term in the exponent instead of the entire expression in order to define the score:

$$Score(\vec{x}_i) = [\vec{x}_i - \vec{\mu}]C^{-1}[\vec{x}_i - \vec{\mu}]^T$$

Note that this score inverts the order of the outliers because it is a *negative* log-likelihood value (with some constant offsets and multiplicative factors removed). Therefore, higher values of the scores are indicative of greater outlier tendency. Points with the score *above* a given threshold are designated as outliers. This outlier score is essentially the squared *Mahalanobis distance* of the data point from the centroid of the data.

As discussed in section 2.4.2, the Mahalanobis distance can be expressed in terms of the distance from the centroid of the data after whitening. In whitening, the data is rotated and reflected to a new orthogonal axis-system in which the covariances between different pairs

³The mean $\vec{\mu}$ and covariance matrix C in Equation 9.14 are actually the estimated values $\hat{\vec{\mu}}$ and \hat{C} (i.e., sample mean and covariance) although the circumflex has been omitted to avoid notational clutter.

of dimensions are 0. This axis-system is the same as one corresponding to the uncorrelated directions of the elliptical Gaussian distribution (cf. section 4.10.2 of Chapter 4). The data is then re-normalized along the new axis directions so that the variances along these directions are of unit value. As a result, the data is effectively re-normalized into a spherical Gaussian (instead of an elliptical Gaussian oriented along arbitrary axis directions). The Mahalanobis distance in the original space is equivalent to the Euclidean distance from the centroid in this re-normalized space.

Next, the above exposition will be formalized mathematically. The sample covariance matrix C can be diagonalized to find the estimated axis directions of zero covariance and corresponding variances along these directions:

$$C = P\Delta P^T$$

Here, P is a $d \times d$ matrix whose columns contain the orthonormal eigenvectors and Δ is a $d \times d$ diagonal matrix whose diagonal entries contain the nonnegative eigenvalues. As discussed in Chapter 2, the eigenvectors of covariance matrices are always orthonormal and the eigenvalues are nonnegative. The eigenvectors correspond to the independent axis directions of the Gaussian distribution and the eigenvalues reflect the variances along these directions. The eigenvector directions are the principal component directions in the data (Definition 2.14), and therefore the discovery of such directions is referred to as principal component analysis (PCA). Since the variances corresponding to eigenvalues are always nonnegative, a nonnegative matrix like Δ can be written as $\Delta = \Sigma^2$, where Σ is a $d \times d$ diagonal matrix in which the diagonal entries contain the standard deviations along the principal components. Therefore, one can write the diagonalization as follows:

$$C = P\Sigma^2P^T$$

One can then express C^{-1} as follows:

$$C^{-1} = P\Sigma^{-2}P^T = (P\Sigma^{-1})(P\Sigma^{-1})^T$$

Then, the (squared) Mahalanobis distance-based score may be defined as follows:

$$\begin{aligned} Score(\vec{x}_i) &= [\vec{x}_i - \vec{\mu}]C^{-1}[\vec{x}_i - \vec{\mu}]^T \\ &= [\vec{x}_i - \vec{\mu}][(P\Sigma^{-1})(P\Sigma^{-1})^T][\vec{x}_i - \vec{\mu}]^T \\ &= [(\vec{x}_i - \vec{\mu})(P\Sigma^{-1})][(\vec{x}_i - \vec{\mu})(P\Sigma^{-1})]^T \\ &= \|((\vec{x}_i - \vec{\mu})P)\Sigma^{-1}\|^2 \end{aligned}$$

Note that the vector $((\vec{x}_i - \vec{\mu})P)$ is essentially the projection of the vector connecting \vec{x}_i to the centroid $\vec{\mu}$ along the eigenvectors in P , and further post-multiplication with Σ^{-1} normalizes the distance along the i th eigenvector with the i th standard deviation. The squared norm of the normalized coordinates is used as the outlier score. Note that the distance to the centroid is computed in the whitened data representation (cf. section 2.4.2 of Chapter 2).

In the Mahalanobis scoring method, the data is assumed to be generated from an independent normal distribution along each principal component. By whitening, one can use a (simpler) *standard* normal distribution to represent the data. After whitening, the scatter plot of the data will roughly have a spherical shape, even if the original data is elliptically elongated. The whitening process discovers the uncorrelated directions in the data which are used to create a new axis system of representation. Normalizing the attribute values

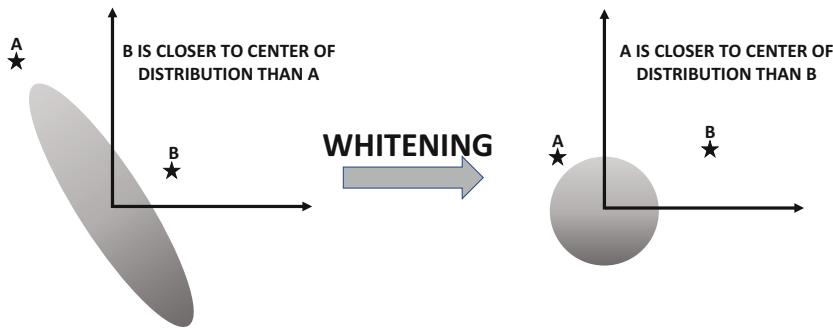


Figure 9.5: Revisiting Figure 2.15: Illustration of how whitening can expose outliers in the data

along each eigenvector direction with its standard deviation during whitening makes the outlier detection algorithm provide equal importance to the different features. In order to understand this point, we provide an example in Figure 9.5. Consider an outlier detection algorithm that reports the distance of a point from the center of the data as an outlier score. A larger distance from the center of the data is more indicative of outlierness. According to this approach, the point A is more likely to be considered an outlier than point B (based on the original data on the left of Figure 9.5). When whitening is used, the original data distribution now becomes spherical, and point B moves farther away from the center of the distribution. As a result, it is now more likely to be considered an outlier, which is the correct conclusion. This is because the data distribution is elongated along the direction of point A, and it is more likely that small variations from the original generating process can create point A as opposed to point B (which is more anomalous). In general, principal component analysis makes algorithms based on Euclidean distance much more sensitive to the aggregate distribution of the data. The whitening approach is able to perform the key normalizations that transform the elliptical data distribution into a spherical one. The Mahalanobis distance simply computes this adjusted Euclidean distance implicitly without making the whitening transformation in an explicit manner.

Since the squared Mahalanobis distance computes the sum of the squares of the normalized distances along the principal components, it is the sum of squares of d independent standard normal distributions. Therefore, the outlier is a χ^2 -distribution with d -degrees of freedom. One can use the cumulative χ^2 -distribution to calculate the probability that the score is at least this large. Larger scores cause smaller probabilities. Smaller probabilities are indicative of a greater degree of outlierness.

Example 9.9 Consider a mean-centered data set with the following points:

$$\{(2, 2), (2, 2), (1, 1), (1, 1), (-1, -1), (-1, -1), (-2, -2), (-2, -2), (1, -1), (-1, 1)\}$$

Note that some points are repeated. Calculate the squared Mahalanobis distance of each point. Which points have the highest outlier score? Calculate the χ^2 -probabilities that the scores are at least this large.

Solution: The covariance matrix of this data set can be shown to be the following:

$$C = \begin{bmatrix} 2.2 & 1.8 \\ 1.8 & 2.2 \end{bmatrix}$$

The inverse covariance matrix is as follows:

$$C^{-1} = \begin{bmatrix} 1.375 & -1.125 \\ -1.125 & 1.375 \end{bmatrix}$$

The two eigenvectors of this covariance matrix are $\vec{e}_1 = [1/\sqrt{2}, 1/\sqrt{2}]^T$ and $\vec{e}_2 = [1/\sqrt{2}, -1/\sqrt{2}]^T$. The corresponding eigenvalues are $\lambda_1 = 4$ and $\lambda_2 = 0.4$, which can be verified using the condition $C\vec{e}_i = \lambda_i\vec{e}_i$ for each i . The large value of λ_1 suggests that the points are primarily aligned along \vec{e}_1 , corresponding to a strong positive correlation. All means are zeros. The squared Mahalanobis distances of the points can be computed as $(\vec{x} - \vec{0})C^{-1}(\vec{x} - \vec{0})^T$ and their values are as follows (in the same order as the listed points):

$$2, 2, 0.5, 0.5, 0.5, 0.5, 2, 2, 5, 5$$

The squared Mahalanobis distances of the last two points are the largest, even though these points are among the closest to the centroid of the data in terms of the Euclidean distance. This is because these points are not aligned along the primary eigenvector $[1, 1]$ and they violate the covariance structure of the data. Even though the two dimensions of mean-centered data are positively correlated, the coordinates of these points have opposite signs. Therefore, these points can be considered the strongest outliers because of their violation of the covariance structure. Using the cumulative χ^2 -distribution with two degrees of freedom, the following probabilities of the score being at least this large are obtained:

$$0.37, 0.37, 0.78, 0.78, 0.78, 0.78, 0.37, 0.37, 0.082, 0.082$$

Note that the last two probabilities are the smallest because these points are outliers. ■

Example 9.10 (Probabilistic Modeling for Binary Data) Consider the data set $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$, in which the i th point \vec{x}_i has **binary** attributes $[x_{i1}, x_{i2}, \dots, x_{id}]$. Let the j th attribute have a fraction f_j of the points set to 1 and other points set to 0. Consider the following outlier score for the i th data point:

$$\text{Score}(\vec{x}_i) = \sum_{j=1}^d [x_{ij} \log(1/f_j) + (1 - x_{ij}) \log(1/(1 - f_j))]$$

What is the probabilistic interpretation of this outlier score?

Solution: If the data is modeled by a multivariate Bernoulli distribution with independent attributes and success parameters p_1, \dots, p_d , then the MLE estimates of $p_1 \dots p_d$ can be shown to be $f_1 \dots f_d$ (cf. section 6.3.2 of Chapter 6). Then, the prob-

ability of generation of the point \vec{x}_i is given by the following (using the Multivariate Bernoulli distribution):

$$p_{\vec{X}}(\vec{x}_i) = \prod_{j=1}^d f_j^{x_{ij}} (1 - f_j)^{(1-x_{ij})}$$

These values will be low for outliers because they have low probabilities of being generated by the model. In contrast, the negative logarithm of this value (i.e., negative log-likelihood fit) will be high for outliers. It can be shown that the above score is precisely this negative log-likelihood fit. ■

Example 9.11 *The Mahalanobis method uses the assumption of a general Gaussian. How does the score for $\vec{x}_i = [x_{i1}, \dots, x_{id}]$ simplify, if one were to assume that the individual attributes are independent? The mean along the j th dimension is μ_j and the variance is σ_j^2 .*

Solution: The negative logarithm of a Gaussian distribution is always proportional to the sum of the scores of the Z-values along independent directions (whether they are axis-parallel or not). Therefore, the score would simplify to the following:

$$\text{Score}(\vec{x}_i) = \sum_{j=1}^d \left(\frac{x_{ij} - \mu_j}{\sigma_j} \right)^2$$

Problem 9.9 Generalize the approach discussed in Example 9.10 for creating outlier scores to a data set containing categorical attributes.

9.4.2 Mixture Models in Outlier Detection

The mixture modeling approach for outlier detection is a generalization of the Mahalanobis method discussed in the previous section. In the Mahalanobis method, it is assumed that the data is generated from a single Gaussian cluster. Such an approach may not work very well for cases in which the data contains multiple clusters. A generalization of the idea underlying the Mahalanobis method is to assume that the data is generated from k Gaussian clusters. Therefore, the basic approach is as follows:

1. Use the mixture modeling approach discussed in section 9.2 in order to create a probabilistic model of data generation to find the parameters of the mixture components $\mathcal{G}_1 \dots \mathcal{G}_k$.
2. Compute the fit of each data point \vec{x}_i based on its generative probability (density). This is essentially an application of the total probability rule for distributions:

$$f_{\vec{X}}(\vec{x}_i) = \sum_{j=1}^k P(\mathcal{G}_j) f_{\vec{X}|\mathcal{G}_j}(\vec{x}_i)$$

3. Report all points \vec{x}_i for which the fit value $f_{\vec{X}}(\vec{x}_i)$ is below a particular threshold as outliers.

This approach is a generalization of the Mahalanobis method, which assumes that a single Gaussian mixture component exists in the data. Another advantage of the method discussed above is that it works for any type of mixture component and not just Gaussian components.

Example 9.12 Suppose that you model gorilla heights as a mixture of two 1-dimensional normal distributions under the assumption that males and females will naturally separate out into two components. Your EM algorithm finds two mixture components of means 4.0 feet and 6.9 feet, respectively. The corresponding standard deviations are 0.7 feet and 0.9 feet, respectively. The prior probabilities are estimated to be 0.6 and 0.4, respectively. You are given three gorillas, which are Mr./Ms. Tiny at 3.5 feet, Mr./Ms. Middling at 5.4 feet, and Mr./Ms. Giant at 7.5 feet. Which of these gorillas has the greatest possibility of being an outlier according to your probabilistic model?

Solution: Using the total probability rule in each case, and ignoring proportionality constants like $1/\sqrt{2\pi}$, we obtain the following:

$$\begin{aligned} f_X(\text{Tiny}) &\propto 0.6 \frac{1}{0.7} \exp\left(-\frac{(3.5 - 4.0)^2}{2 * 0.7^2}\right) + 0.4 \frac{1}{0.9} \exp\left(-\frac{(3.5 - 6.9)^2}{2 * 0.9^2}\right) \approx 0.668 \\ f_X(\text{Middling}) &\propto 0.6 \frac{1}{0.7} \exp\left(-\frac{(5.4 - 4.0)^2}{2 * 0.7^2}\right) + 0.4 \frac{1}{0.9} \exp\left(-\frac{(5.4 - 6.9)^2}{2 * 0.9^2}\right) \approx 0.227 \\ f_X(\text{Giant}) &\propto 0.6 \frac{1}{0.7} \exp\left(-\frac{(7.5 - 4.0)^2}{2 * 0.7^2}\right) + 0.4 \frac{1}{0.9} \exp\left(-\frac{(7.5 - 6.9)^2}{2 * 0.9^2}\right) \approx 0.356 \end{aligned}$$

In other words, Mr./Ms. Middling is the most likely outlier, since it has the lowest fit value. Intuitively, even though this gorilla does not have a very large or very small height, it fits neither of the two mixture components well. One possible conjecture is that the two mixture components correspond to male and female gorillas (who have a high degree of sexual dimorphism), and the middling height is not very natural to either group. Note that this type of outlier will not be found by the Mahalanobis method, which is biased towards extreme values because of its assumption of a single Gaussian component. Another advantage of the mixture model is that one can use other types of mixture components (e.g., Bernoulli or multinomial components) for data of other types. ■

Problem 9.10 Implement the algorithm discussed in this section in a programming language of your choice.

9.4.3 Matrix Factorization for Outlier Detection

Matrix factorization is an effective approach for outlier detection in the sense that it can find outliers at the level of data matrix *entries* as opposed to data matrix *rows*. Furthermore, the outlier scores on the entries can be used to find either outlier columns *or* outlier rows in the matrix. This type of approach provides better insights into the specific entries that cause a row or a column to be an outlier in terms of its inconsistency with respect to other

rows or columns. The greater granularity in insights also provides the ability to find outliers in incomplete data sets.

Consider an $n \times d$ data matrix in which the (i, j) th entry is x_{ij} . The matrix D can be factorized into two low-rank matrices U and V as follows:

$$D \approx UV^T$$

Here, U is an $n \times k$ matrix and V is a $d \times k$ matrix, where $k \ll \min\{n, d\}$. Therefore, U and V correspond to a compressed representation of the data, and the matrix can be approximately reconstructed as UV^T . Since compression relies on the aggregate correlations and patterns in the data matrix, the reconstruction is not very effective for those entries that are not consistent with these patterns. As a result, such entries will have large absolute values in the following *error* or *residual matrix* $E = [e_{ij}]$:

$$E = D - UV^T$$

Note that each entry e_{ij} may be positive or negative, although large absolute values are indicative of greater outlierness. Therefore, the outlier score of the (i, j) th entry is e_{ij}^2 and those entries with scores above a given threshold are designated as outliers.

It is also possible to create outlier scores for the i th row as follows:

$$\text{Score for } i\text{th row} = \sum_{j=1}^d e_{ij}^2 \quad \forall i \quad (9.15)$$

A similar approach may be used to create an outlier score for the j th column:

$$\text{Score for } j\text{th column} = \sum_{i=1}^n e_{ij}^2 \quad \forall j \quad (9.16)$$

All scores above a given threshold are designated as outliers. The outlier scores have a natural probabilistic interpretation when the Frobenius norm is used to determine outliers.

As discussed in the section on matrix factorization, each residual error e_{ij} is assumed to be drawn from a normal distribution with zero mean and some unknown standard deviation σ . Therefore, one can even use the residual to estimate the probability that a specific point is an outlier. The standard deviation of the normal distribution can be estimated as follows:

$$\sigma^2 = \frac{\sum_{i=1}^n \sum_{j=1}^d e_{ij}^2}{n \cdot d}$$

The Z -value of each residual can be computed as follows:

$$z_{ij} = e_{ij} / \sigma$$

All points whose Z -values are greater than 3 in absolute magnitude can be designated as outliers at a confidence level of 99.9%. One can use other probabilistic thresholds from the normal distribution in order to relax or tighten the criteria for points being designated as outliers.

Example 9.13 (Why Matrix Factorization Works) Consider the following matrix $A = [a_{ij}]$ with significant cross-row correlations:

$$A = \begin{bmatrix} 2 & 4 & 6 & 8 \\ 2 & 4 & 6 & 8 \\ 3 & 6 & 9 & 12 \\ 1 & 2 & 3 & 4.1 \end{bmatrix}$$

Visually examine the matrix and point out which entry is the outlier entry. Now justify this observation by factorizing this matrix into a 4×1 and 1×4 matrix. The purpose of this example is to illustrate how violations of patterns in matrix entries translate to errors in low rank factorizations.

Solution: In this case, it is evident that the entry $a_{44} = 4.1$ is not consistent with the patterns in other rows (which are all multiples of $[1, 2, 3, 4]$) and is therefore an outlier entry. This can be verified by approximately factorizing the matrix as follows:

$$\begin{bmatrix} 2 & 4 & 6 & 8 \\ 2 & 4 & 6 & 8 \\ 3 & 6 & 9 & 12 \\ 1 & 2 & 3 & 4.1 \end{bmatrix} \approx \begin{bmatrix} 2 \\ 2 \\ 3 \\ 1 \end{bmatrix} [1 \quad 2 \quad 3 \quad 4] = \begin{bmatrix} 2 & 4 & 6 & 8 \\ 2 & 4 & 6 & 8 \\ 3 & 6 & 9 & 12 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

In this case, it is evident that all errors are 0 except for the residual of a_{44} , which is 0.1. This is also the entry that violates the cross-row correlations. Although this example is slightly oversimplified in using a rank-1 factorization (where rows are multiples of a single row), the general principle is true for factorizations of higher ranks (where all matrix rows are expressed as linear combinations of a small number of *basis* rows). ■

9.4.3.1 Outlier Detection in Incomplete Matrices

One nice property of the matrix factorization approach is that it allows (observed) entries, rows, or columns in incomplete data to be identified as outliers. As discussed in section 9.3.1.3, one can use gradient descent in order to factorize an incomplete data matrix D into the form UV^T . One can then create an (incomplete) error matrix $E = D - UV^T$, where unobserved entries of D are correspondingly undefined in $E = [e_{ij}]$. All observed entries in E above a given threshold are designated as outliers.

One can also find outlier columns and rows of E by using a similar scoring approach as in the case of complete data matrices. The main difference is that one must be careful to normalize for the number of observed entries in each row (or column). Let O be the set of all observed entries in the matrix D :

$$O = \{(i, j) : x_{ij} \text{ is observed}\}$$

Let l_i be the number of observed entries in the i th row and m_j be the number of observed entries in the j th column. The outlier score for the i th row is defined as follows:

$$\text{Score for } i\text{th row} = \frac{\sum_{j:(i,j) \in O} e_{ij}^2}{l_i} \quad \forall i \quad (9.17)$$

Note the above score has l_i in the denominator to normalize for the number of entries in the i th row. A similar approach may be used to create an outlier score for the j th column:

$$\text{Score for } j\text{th column} = \frac{\sum_{i:(i,j) \in O} e_{ij}^2}{m_j} \quad \forall j \quad (9.18)$$

These types of outlier scores can be very useful in recommender systems, where x_{ij} is the rating of the i th user for the j th item. A high outlier score for the i th user means that they have often rated items in an “inconsistent” way in relation to the patterns seen by other users. For example, if most users who enjoy action movies do not like documentaries and a particular user likes both action movies and documentaries, they may be flagged as an outlier.

9.5 Summary

This chapter introduces unsupervised methods in machine learning from the probabilistic perspective. The clustering, matrix factorization, and outlier detection problems are introduced. In the case of clustering, a natural relationship to the classification problem is shown. For example, the same generative models are used for mixture-model clustering and naïve Bayes classification. The generative models for matrix factorization are similar to those of classification, except that the generation is done in an entry-wise fashion rather than row-wise fashion. In the case of matrix factorization, it is also possible to develop analogous models to regression and discriminative classifiers like logistic regression.

Both clustering and matrix factorization can be considered compression models. Outliers have a natural complementary relationship to this compression perspective. Those points that cannot be compressed easily have a low likelihood fit with respect to the data and are therefore considered outliers (because of their low probability of generation). As a result, outliers are often byproducts of such clustering and matrix factorization algorithms.

9.6 Further Reading

Various types of unsupervised clustering and matrix factorization models are introduced in [1, 3, 6, 10, 33]. A detailed discussion on data clustering may be found in [7]. A detailed discussion of matrix factorization and its application to recommender systems may be found in [2, 6]. Probabilistic latent semantic analysis was introduced in [36, 37] and its extension to Latent Dirichlet Allocation may be found in [12]. Logistic matrix factorization methods are discussed in [40]. The logistic matrix factorization algorithm is closely related to *word2vec*. An exposition of the *word2vec* algorithm along with its connection to logistic matrix factorization is provided in [4]. The problem of outlier analysis is discussed in [5].

9.7 Exercises

1. The exposition in the chapter designs a mixture model with axis-parallel Gaussians in which individual mixture components have zero covariances. Design a variant of the mixture modeling algorithm in which general Gaussians are used (with covariance parameters) for the mixture components.

2. Repeat Example 9.3 in order to develop a deterministic variant of the EM algorithm that uses general Gaussians (i.e., not axis-parallel Gaussians) for the mixture components.
3. Consider the case where you have count data along multiple dimensions, and you use an independent Poisson distribution along each dimension to model each mixture component in EM-clustering. Derive the E- and M-steps of the expectation-maximization algorithm.
4. In the case of matrix factorization with numeric data, consider the case where L_1 -regularization and L_2 -loss is used. What type of probabilistic assumption does this choice imply for the errors and the parameters?
5. Implement the PLSA algorithm in a programming language of your choice.
6. [Histogram-Based Outlier Detection] Extend the method for outlier detection discussed in Example 9.10 to numerical data with the use of discretization.
7. The logistic matrix factorization approach discussed in the chapter is designed for sparse numerical data. Design a simpler version for non-sparse binary data.
8. Discuss how logistic matrix factorization may be used to find outliers in sparse binary data.
9. Consider a situation in which the incomplete $n \times d$ matrix D is approximately factorized into an $n \times k$ matrix U and a $d \times k$ matrix V for prediction as follows:

$$D \approx UV^T$$

Suppose you add the constraint that all entries of the penultimate column of U and the final column of V are fixed to 1. Discuss the similarity of this model with the addition of bias to regression and classification models. How is gradient-descent modified?

10. In the scenario of Exercise 9, will the sum of squared errors on observed entries be better optimized with or without constraints on the final columns of U and V ? Why might it be desirable to add such a constraint during the estimation of missing entries?
11. Derive the gradient-descent updates of unconstrained matrix factorization with L_1 -regularization. You may assume that the regularization parameter is $\lambda > 0$.
12. Suppose that you have a data set consisting of a mixture of binary, categorical, and numeric attributes. One possible way of using the Bayes model is by preprocessing all attributes to the binary type. Discuss how you can directly create a Bayes model for mixed-attribute data without preprocessing.
13. Most matrix factorization methods use the squared error on the entries in order to define the objective function. As discussed in the chapter, such an objective function corresponds to a Gaussian model of the errors. Suppose you were to use the L_p -norm instead of the L_2 -norm to create the objective function. What type of probabilistic model on the errors does this objective function correspond to?
14. Suppose that you have a very large and dense matrix D of low rank that you cannot hold in memory, and you want to factorize it as $D \approx UV^T$. Propose a method for factorization that uses only sparse matrix multiplication. You might find ideas related to sampling helpful when combined with ideas used in recommender systems.

- 15.** Consider the EM clustering algorithm with the Bernoulli model. Can you think of a k -means-like algorithm that is a deterministic version of this model? What type of similarity function does it use?
- 16.** You create a 200×200 grid of values spanning the domain of a 2-dimensional data set. You calculate the kernel density at each grid-point. Discuss how you can use density thresholds to create clusters. You might find it helpful to examine some of the kernel density visualizations shown in Chapter 6. When should two adjacent grid points belong to the same cluster? When should a grid point not belong to *any* cluster? How can clusters on grid points be translated to clusters on data points?
- 17.** When you use a Gaussian distribution to model clusters, is there any type of bias you impose on the shapes of the clusters? How does the shape of the cluster vary with the type of Gaussian distribution used? What are the shapes of clusters in the case of the kernel density estimation technique of Exercise 16?
- 18.** Suppose that you are not given the original $n \times d$ data matrix D but an $n \times n$ similarity matrix S containing dot products between rows of D . Therefore, we have $S = DD^T$. Show how you can use the Mahalanobis method to identify outlier points starting from S rather than D . A key point of this exercise is to understand how one can extract a PCA-like representation of D from S . You will find the material on explicit feature engineering in Chapter 7 helpful.
- 19.** [Kernel Mahalanobis method:] Suppose that your similarity matrix S in Exercise 18 does not contain dot products like $\vec{x}_i \cdot \vec{x}_j$ but instead contains higher order similarities like $(2 + \vec{x}_i \cdot \vec{x}_j)^2$. How would you interpret the outliers found using the approach of Exercise 18?
- 20.** Use the ideas in Exercises 18 and 19 to propose how one might perform EM-clustering in a nonlinear engineered space.
- 21.** The Mahalanobis method makes the assumption that the data is distributed using the Gaussian distribution. Suppose you have the insight that the individual (nonnegative) attributes are distributed based on the exponential distribution (and one can assume that the different attributes are independent). Show that the MLE-based outlier score of the i th data point $\vec{x}_i = [x_{i1} \dots x_{id}]$ is given by the following:

$$\text{Score}(\vec{x}_i) = \sum_{j=1}^d (x_{ij}/\mu_j)$$

Here, μ_j is the mean of attribute j .

- 22.** A data set of *strictly* positive integer counts is modeled with the multinomial distribution. Show that the MLE-based outlier score of $\vec{x}_i = [x_{i1} \dots x_{id}]$ is the following:

$$\text{Score}(\vec{x}_i) = \sum_{j=1}^d [x_{ij} \ln(x_{ij}/p_j) + x_{ij} + 0.5 \ln(x_{ij})] - \ln \left[\left(\sum_i \sum_j x_{ij} \right)! \right]$$

Here, $p_j = \sum_i x_{ij} / (\sum_i \sum_j x_{ij})$. Use the approximation $m! \approx \sqrt{2\pi m}(m/e)^m$ for $m \geq 1$. In the general case of nonnegative counts, show the following for $x_{ij}^+ = \max\{x_{ij}, 1\}$:

$$\text{Score}(\vec{x}_i) = \sum_{j=1}^d [x_{ij} \ln(x_{ij}^+ / p_j) + x_{ij}^+ + 0.5 \ln(x_{ij}^+)] - \ln \left[\left(\sum_i \sum_j x_{ij} \right)! \right]$$



Chapter 10

Discrete State Markov Processes

“The events in our lives happen in a sequence in time, but in their significance to ourselves they find their own order the continuous thread of revelation.”—
Eudora Welty

10.1 Introduction

The probabilistic processes discussed thus far in this book for generating random variables (e.g., binomial or Poisson processes) are based on trials that are independent of one another. In other words, if multiple random variables were to be generated, the generation of each variable is an independent and identical process. This chapter discusses a different class of probabilistic processes in which the trials follow one another in time and the outcomes are dependent on one another. For example, tossing a coin ten times is an independent sequence of Bernoulli trials. On the other hand, consider a situation where a user selects links on Web pages at random and clicks on them to surf the Web. The outcomes (i.e., accessed Web pages) using random clicks are not independent of one another. This is because the outcome of the previous trial (i.e., current choice of Web page) will affect which Web pages are available to be clicked on in the next trial.

Most real-world events occur in a sequence in which the individual elements are dependent on one another by some probabilistic form of causality or correlation. This is because the outcome of a current event *probabilistically* affects how subsequent outcomes unfold, and these outcomes cannot be deterministically predicted in most cases. For example, if it is very hot today, it is more likely that it will be hot tomorrow (because weather patterns show a certain level of temporal contiguity). Many real-world events show a high level of temporal dependency, because of which it is important to have probabilistic mechanisms to model such dependencies across time. A particularly important type of probabilistic process that models such dependencies is the *discrete state Markov process*, which is also referred to as a *Markov chain*.

A Markov chain is a model representing a sequence of trials in which the outcome of a trial depends only on the outcome of the *immediately* preceding trial (and not the

earlier ones). The property of an outcome depending on only the immediately preceding one is referred to as the *Markov property*. Although this assumption might sound simplistic, it turns out to be very powerful for modeling real-world applications. For example, the temperature at a given moment depends heavily on the temperature in the previous clock tick to an extent that one can ignore all other (preceding) ticks. Although the idea that outcomes are dependent only on immediately preceding trials is a simplifying assumption, it retains a significant level of generality in modeling sequential dependencies. For example, it is possible to collapse a set of k consecutive trials into a single “compound” trial in order to transform dependencies of length k into dependencies of length 1.

It is evident that Markov processes need a formal mathematical representation of the “memory” of outcomes that have occurred in the past, which can be used to model the probabilities of future events. This type of memory is modeled with the notion of a *state*, which captures the current configuration of the Markov process and determines the distribution of outcomes for the next trial. The states can be discrete (e.g., current Web page being surfed), or they can be continuous (e.g., current temperature). This chapter will focus on discrete state Markov processes. The fundamental model that represents the sequence of outcomes of such a process is referred to as a *Markov chain*.

A Markov chain contains a set of states, and the chain is always in one of these states, corresponding to the current configuration of the experiment (including the most recent trial). The trials performed at a state have a categorical PMF that depends on that state — the outcome of this trial decides the state to which the Markov chain moves in order to execute the next trial. Therefore, each trial of the Markov chain results in a change in state, which is also referred to as a *transition*. The probability distribution of the state to which a Markov chain moves is itself dependent on the state that it currently is in. Therefore, each state has its conditional probability distribution of transitions, which is defined by a categorical distribution with as many outcomes as the number of states. The use of the categorical distribution is a natural consequence of the discrete nature of the processes studied in this chapter. The Markov chain has a starting state, which defines the starting configuration. Many Markov chains also have a *steady-state* (i.e., long-term) probability distribution that is independent of the starting state. The steady-state distribution is also referred to as the *stationary distribution*.

This chapter will also introduce applications of Markov chains to generative processes in machine learning. The states of the Markov chain play the same role as mixture components, and each state is associated with a generating distribution. Just as independent draws from the generating distributions of mixture components can be used to generate data in mixture models, Markov models can be used to generate sequence data by drawing symbols from a categorical distribution associated with the current state during each transition. In many cases, one may only have access to the observed data sequence, and it is assumed that the data is generated by some Markov model behind the scenes. Such Markov models are also referred to as *hidden Markov models*, and are used for modeling many types of sequential data. Thus, *hidden Markov models are the sequential analog of mixture models*.

10.1.1 Chapter Organization

This chapter is organized as follows. The next section introduces Markov chains. Applications of Markov chains are introduced in section 10.3. The general principles behind the use of Markovian models for generation of data are discussed in section 10.4. The use of these principles to construct hidden Markov models is introduced in section 10.5. Applications of hidden Markov models are discussed in section 10.6. A summary is given in section 10.7.

10.2 Markov Chains

A Markov chain is defined over a set of n states denoted by $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$, and the Markov process is always in one of these states (based on the outcome of the most recent trial). Since Markov processes use sequential transitions, it is assumed that these transitions occur at integer time stamps $1, 2, \dots, t, \dots, \infty$. At time stamp t , the Markov chain is in state s_{it} , and it transitions to one of the states s_{it+1} drawn from \mathcal{S} . It is possible for the state s_t to be the same as s_{t+1} , which corresponds to a self-transition. The probability distribution of the outcomes of the trial at state s_{it} is specific to the current state s_{it} , and is defined by an n -way categorical distribution — the possible outcomes of this categorical distribution are $s_1 \dots s_n$. This distribution is technically a conditional distribution, since it depends on the state s_{it} . Rather than using the heavy notations of conditional distributions, it is easier to define the distributions of these trials in terms of *transition probabilities*. Specifically, the probability that a transition occurs from state s_i to state s_j (given that the Markov process is currently in state s_i) is denoted¹. Since one always moves from the current state s_i to *some* state in each transition (including the possibility of a self-transition), the sum of the probabilities of transitions from state s_i to all other states is given by 1:

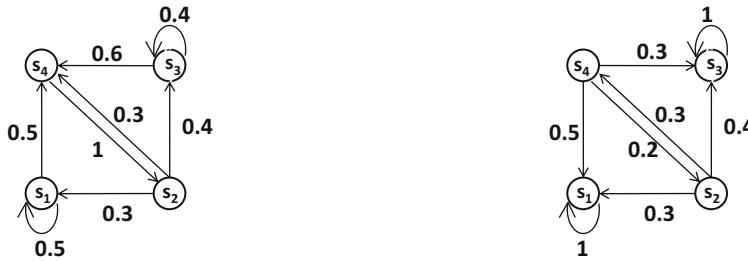
$$\sum_{j=1}^n p_{ij} = 1$$

This condition is the same as the requirement that the PMF of a categorical distribution must sum to 1 over all outcomes. Since such a vector of probabilities exists for each state s_i , one can create a *transition matrix* $P = [p_{ij}]$. Another notational simplification is that we will interchangeably denote s_i by its index i in the following discussion. Each row of the matrix P has probability values that sum to 1, and the i th row contains the probabilities of transitions from state i to other states. Representing a Markov chain with a transition matrix has the advantage that one can draw on a number of properties of matrices from linear algebra to derive useful results.

The transition matrix P can also be used to create a graphical representation of the Markov chain. This graphical representation denotes states with vertices and transitions with edges. A vertex is created for each state s_i , resulting in a graphical structure with $|\mathcal{S}|$ vertices. For each positive transition probability p_{ij} , a directed edge is used to indicate the transition probability from node s_i to node s_j . The graph may not be complete, as the transitions corresponding to $p_{ij} = 0$ can be omitted from the graph. The graph may contain self-loops, when $p_{ii} \neq 0$, and it corresponds to the case when a transition occurs from a state to itself. Although we have used the notation $\{s_1, \dots, s_n\}$ to indicate states, we will occasionally use the integers $\{1, \dots, n\}$ to indicate states in such transition diagrams for greater simplicity. In both cases, the transition probability from the i th state (or s_i) to the j th state (or s_j) is denoted by p_{ij} . These alternative notations will be used interchangeably in this chapter. Examples of Markov chains are illustrated in Figure 10.1. It is evident that not all pairs of states have edges between them and missing edges correspond to transition probabilities of 0. The set of edges in the Markov chain is denoted by A . Furthermore, the sum of the probabilities outgoing from a node is 1, which mirrors the fact that the sum of the probabilities of outgoing transitions from a node is 1.

¹The probability p_{ij} is actually a convenient and matrix-friendly abbreviation for the conditional PMF of the outcome at a given state S_t at time t , when S_t is known to be s_i :

$$p_{S_{t+1}|[S_t=s_i]}(s_j) = p_{ij}$$



(a) Steady-state probabilities independent of starting state (b) Steady-state probabilities not independent of starting state

Figure 10.1: Examples of Markov chains

Example 10.1 (Snakes and Ladders) Explain how the classical game of snakes and adders can be modeled as a Markov chain. Assume that a single player rolls the die with a position marker starting from square (state) 1, until it reaches square 100. If a die roll with the marker at square number 95 or higher results in overshooting square 100, then the position of the marker does not change.

Solution: The classical game of snakes and ladders is shown in Figure 10.2 and can be represented as a Markov chain. Each square on the board (where the position marker can sit) is a state, and it typically has six outgoing edges from each state other than the “finish” state (corresponding to square 100). These six edges correspond to the six outcomes of a die roll. Therefore, each such transition has probability $1/6$, leading to some other state depending on the outcome of a die roll. However, some states in which the position marker is at square 95 or higher might have self-loops corresponding to those die rolls in which the position marker would overshoot square 100 if it were to be moved from its current position based on die outcome (thereby resulting in an illegal move). The probabilities of these self-loops is $k/6$ where k is the number of outcomes leading to an illegal move. The destination state of the transition might incorporate the effect of sliding on a snake or climbing a ladder. The “finish” state contains a self-loop with probability 1. ■

Example 10.2 (Web Browsing) You are a business owner and keep track of how people navigate your Website with the use of Web logs. You are able to track the path of individual visitors in your site. Discuss how you can model your Web traffic as a Markov chain.

Solution: Each page in the Web site is a state. For each page, the probability of moving to any other state during the session is calculated as the fraction of times that the user has moved to that page (based on log data). This calculation is an MLE estimate of a categorical probability. Note that when the user does not move to any other state (because it is the end of a session), it is still considered a transition to itself and therefore, the probability out of a page will always be equal to 1. ■

The trials of the Markov chain use a starting state $f \in \mathcal{S}$, and make repeated transitions from state to state. Therefore, at any given time t , there must be some probability

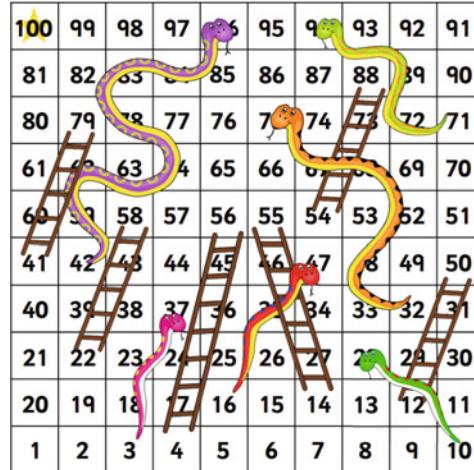


Figure 10.2: An example of a snakes-and-ladders game

distribution $P(S_t = s_i)$ for each i . Note that this probability distribution depends on time t , especially in the initial phases of the Markov process. The probabilities of the different states begin to stabilize after a long period of time for many important types of Markov chains. The stabilized probabilities of states s_1, s_2, \dots, s_n are denoted by $\pi_1 \dots \pi_n$, and are referred to as *steady-state probabilities* or *stationary probabilities*. These stationary probabilities depend on the topology of the chain and (in some cases) the starting state. This process of convergence of the Markov chain is referred to as *mixing*. The n -dimensional row vector of steady-state probabilities is denoted by $\vec{\pi} = [\pi_1 \dots \pi_n]$.

A natural question arises as to what conditions enable steady-state probabilities to exist. When they do exist, are they independent of the initial state? In order to understand the effect of the initial state f , consider the Markov chain illustrated in Figure 10.1(b). In this case, starting at state $f = s_1$ results in a steady-state probability of 1 for state s_1 and 0 for all other states. This is because the transitions get “trapped” in state s_1 , which has a self-loop with probability 1. Similarly, starting at state $f = s_3$ results in a steady-state probability of 1 for state s_3 and 0 for all other states. Therefore, the steady-state probability in this case is not independent of the starting state. On the other hand, it can be shown that the steady-state probability of Figure 10.1(a) is independent of the starting state. What is the difference between Figure 10.1(a) and Figure 10.1(b) that causes one of the Markov chains to have a steady-state distribution that is independent of starting state and the other to have a steady-state distribution that is dependent on the starting state? This is a point that we will revisit later in section 10.2.4. For now, we simply assume that we have a Markov chain with a steady-state probability distribution that is independent of starting state.

10.2.1 Steady-State Behavior of Markov Chains

Consider a Markov chain in which a unique steady-state probability distribution that is independent of the starting state does exist. How can these steady-state probabilities be determined? The key property of steady-state is that *the probability of a state s_i is also equal to the probability of transition into the state s_i* . After all, the Markov chain is currently in state s_i because a transition must have occurred from some state s_j into it in the most

recent transition. Note that the probability that a transition occurred from state s_j and that it transitioned to state s_i is given by the product of the probabilities of these two events, which is $\pi_j p_{ji}$. This leads to the important steady-state condition for a Markov chain for each state i by summing up the transition probabilities from all possible states from which the transition occurred to state i :

$$\pi_i = \sum_{j=1}^n \pi_j p_{ji} \quad \forall i \quad (10.1)$$

By summing up the mutually exclusive probability values $\pi_j p_{ji}$ of transitions from different states s_j , one obtains the total probability of transition to state s_i . This value is equal to the probability π_i of state s_i . Note that the above condition must be true for each state s_i for steady-state to hold. Therefore, one can write the above set of n conditions in vector and matrix form by defining them in terms of the row vector $\vec{\pi} = [\pi_1, \dots, \pi_n]$ and transition matrix P :

$$\vec{\pi} = \vec{\pi}P$$

In addition, one must have the following normalization condition on the steady-state probabilities:

$$\sum_{i=1}^n \pi_i = \|\vec{\pi}\|_1 = 1$$

The normalization condition reflects the fact that the sum of the probabilities over all states is 1. Furthermore, since each π_i is a probability, the constraint $\pi_i \in (0, 1)$ must hold for each i . A condition such as $\vec{\pi} = \vec{\pi}P$ is a left *eigenvector*² condition, in which $\vec{\pi}$ is an eigenvector of P with eigenvalue 1. One can also solve the system of equations numerically in order to determine the steady-state behavior of a Markov chain. A simple way to solve the system numerically would be to initialize $\vec{\pi}^{(0)}$ to a vector in which each probability is $1/n$ and updating it iteratively as follows:

$$\vec{\pi}^{(t+1)} = \vec{\pi}^{(t)}P$$

It is noteworthy that the elements of $\vec{\pi}^{(t+1)}$ represent the probabilities of the states after $(t+1)$ transitions of the Markov chain. The property that the elements of $\vec{\pi}^{(t)}$ sum to 1 will be maintained³ in $\vec{\pi}^{(t+1)}$ by this update. However, to guard against the effect of successive numerical errors, one can explicitly scale $\vec{\pi}^{(t+1)}$ after the update so that its elements sum to 1. When t becomes very large, $\vec{\pi}^{(t+1)}$ will converge to the steady-state probability vector if it does exist.

In order to understand steady-state behavior, we will use an example as discussed below.

Example 10.3 A food-truck serves a dish of a single type on a given day out of four possible types. If fish is served today, then either beef or chicken will be served tomorrow with probabilities 0.4 and 0.6, respectively. If chicken is served today, then either pork or fish will be served tomorrow with probabilities 0.3 and 0.7, respectively. If either pork or beef is served today, then either fish or chicken will be served tomorrow.

²A left eigenvector of a square matrix A is a row vector \vec{x} so that $\vec{x}A = \lambda\vec{x}$ for some scalar λ , referred to as its eigenvalue. In other words, the linear operator A simply scales the vector [6]. On the other hand, a right eigenvector is the column vector \vec{x} so that $A\vec{x} = \lambda\vec{x}$. Although the word “eigenvector” refers to a right eigenvector by default, this chapter will deviate from this practice and assume that “eigenvectors” are all left eigenvectors unless otherwise specified.

³The maintenance of this property can be shown by using the fact that the rows of P sum to 1.

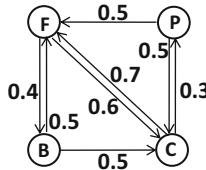


Figure 10.3: An example of a Markov chain modeling food choice on successive days

row with probabilities 0.5 and 0.5. Construct a Markov chain showing this sequential process. Over a long period of time, what fraction of the days is each type of food eaten?

Solution: In this case, the Markov chain can be constructed in which the states $\{F, C, B, P\}$ are denoted by the first letter of specific type of dish constructed on a given day. Note that the state on a current day completely defines the probabilities of states on the next day. The corresponding Markov chain is illustrated in Figure 10.3. Since the question asks about the fraction of time *over a long period of time* that each food is cooked, one is really looking for the steady-state probabilities of the different states. The 4-dimensional steady-state vector is as follows:

$$\vec{\pi} = [\pi_f, \pi_c, \pi_b, \pi_p]$$

In this case, the subscript alphabets f, c, b, p correspond to the different types of foods in that order. The 4×4 transition matrix P of the Markov chain also arranges the rows and columns in that order. Correspondingly, the matrix P is as follows:

$$P = \begin{bmatrix} & f & c & b & p \\ f & 0.0 & 0.6 & 0.4 & 0.0 \\ c & 0.7 & 0.0 & 0.0 & 0.3 \\ b & 0.5 & 0.5 & 0.0 & 0.0 \\ p & 0.5 & 0.5 & 0.0 & 0.0 \end{bmatrix}$$

On solving the equation $\vec{\pi} = \vec{\pi}P$ numerically, one obtains the following vector of probabilities:

$$\vec{\pi} = [0.3812, 0.3587, 0.1525, 0.1076]$$

Therefore, fish is served 38.12% of the time, whereas chicken is served 35.87% of the time. Beef is served 15.25% of the time, and pork is served 10.76% of the time. ■

Problem 10.1 Days are either rainy, sunny, or cloudy. A crude weather model says that if it is sunny today, then it will be sunny tomorrow with probability 60%, cloudy with probability 25%, and rainy, otherwise. If it is cloudy today, then it is equally likely to be cloudy, rainy, or sunny tomorrow. If it is rainy today, it will be rainy tomorrow with probability 70%, cloudy with probability 20%, and sunny with probability 10%. Based on the crude weather model, what percentage of the days are rainy, cloudy, and sunny?

Problem 10.2 Mary is cheerful, normal, or gloomy. If she is cheerful today, she will be cheerful tomorrow with probability 0.5, normal with probability 0.4, and gloomy with proba-

bility 0.1. If she is normal today, she will be normal tomorrow with probability 0.7, cheerful with probability 0.2, and gloomy with probability 0.1. If she is gloomy today, she will be gloomy tomorrow with probability 0.3, normal with probability 0.6, and cheerful with probability 0.1. Construct a Markov chain that models Mary's emotional state. What is the percentage of days on which Mary is expected to be cheerful?

10.2.2 Transient Behavior of Markov Chains

When a Markov process starts, it typically begins in a particular state or in a distribution of states (for initial conditions of a probabilistic nature). The transient behavior examines the probabilities of the states of the Markov chain in the first k steps for relatively small values of k . Let $\vec{\pi}^{(0)}$ be the probability vector corresponding to the initial state configuration. In most cases, one starts from a single state, and therefore the vector $\vec{\pi}^{(0)}$ contains a single value of 1 and remaining 0s. Let $\vec{\pi}^{(k)}$ be the probability vector corresponding to state probabilities after k steps. Using the same argument as that for justifying the steady-state condition $\vec{\pi} = \vec{\pi}P$, one can create a similar condition relating $\vec{\pi}^{(k)}$ and $\vec{\pi}^{(k+1)}$:

$$\vec{\pi}^{(k+1)} = \vec{\pi}^{(k)}P$$

For smaller values of k , the value of $\vec{\pi}^{(k)}$ is heavily dependent on the initial state $\vec{\pi}^{(0)}$. However, as k increases, the value of $\vec{\pi}^{(k)}$ converges to the steady-state value, as long as the Markov chain is *ergodic* (which corresponds to strong connectivity and aperiodicity of its graph structure according to section 10.2.4). In fact, the update $\vec{\pi}^{(k+1)} = \vec{\pi}^{(k)}P$ corresponds to the numerical approach used to solve the steady-state equations. One can also write the aforementioned transient update equation between $\vec{\pi}^{(k)}$ and $\vec{\pi}^{(t+1)}$ directly in terms of $\vec{\pi}^{(0)}$:

$$\vec{\pi}^{(k)} = \vec{\pi}^{(0)}P^k$$

The matrix P^k represents the k -step transition matrix of the Markov chain, which is defined as follows:

Definition 10.1 (k -Step Transition Matrix) *The k -step probability of a transition from state i to state j is the probability that the Markov chain makes a transition from state i to state j in exactly k steps. The $n \times n$ matrix in which the (i, j) th entry contains this k -step transition probability is given by the k th power P^k of the transition matrix P .*

Another interesting measure used in Markov chains is the *expected number of hits in m steps* starting from an initial state-vector $\vec{\pi}^{(0)}$. The expected number of hits is an n -dimensional vector just like the state probability vector, and its elements contain the expected number of times that a state is visited in the first m steps (excluding the initial state).

Definition 10.2 (Expected Number of Hits) *The expected number of hits of state i for starting state vector $\vec{\pi}^{(0)}$ is the expected number of times that state i is visited in the first m steps (not including the initial state of the Markov process). The expected number of hits for the n nodes is represented by the n -dimensional row vector \vec{h} .*

The expected number of hits can be computed by observing that the contribution of the k th step to the expected number of hits to each of the n nodes is contained in the vector $\vec{\pi}^{(k)}$.

Therefore, by summing up the contributions of various steps, one obtains the following:

$$\begin{aligned}\vec{h} &= \sum_{k=1}^m \vec{\pi}^{(k)} \\ &= \vec{\pi}^{(0)} \sum_{k=1}^m P^k\end{aligned}$$

The matrix $\sum_{k=1}^m P^k$ in the above expression is an $n \times n$ matrix in which the (i, j) th entry contains the expected number of times one would visit state j starting at state i . This is a useful matrix because it is a measure of the “visit dependence” between pairs of nodes.

Definition 10.3 (m-Step Expected Hit Matrix) *The m-step expected hit matrix is an $n \times n$ matrix in which the (i, j) th entry contains the expected number of hits on a node j in m steps, when starting the Markov process in step i . The initial state i is not included in the expected count of the (i, i) th entry. The m-step expected hit matrix is expressed in terms of the transition matrix as $\sum_{k=1}^m P^k$.*

When a node (state) i in a Markov chain is highly connected to node j through many directed paths, the (i, j) th entry of the m -path expected hit matrix will be high. The value of m is a free parameter, and it should be set to values that are typically somewhere between 1 and n . Larger values of m allow connectivity through longer paths, but also tend to favor nodes that are highly connected on a global basis (and not just a local basis). Very small values of m allow connectivity only through a very small number of steps, and therefore the effect of the starting state will be very significant.

This type of approach is often used in machine learning for finding related nodes to a given starting node in social networks based on a Markov chain. For example, consider a probabilistic process in which one performs a random walk over an undirected social network, where one can hop from a given node to any adjacently connected node (friend) with equal probability. The expected number of hits at various nodes in m steps (starting at node i) is used to find related nodes. In such a case, using a larger value of m would result in highly connected social actors to be heavily favored, irrespective of their distances to node i . On the other hand, using a small value of m will favor nodes in the immediate friendship locality of node i .

Example 10.4 Consider the Markov chain of Example 10.3 in which beef is served on a particular day (May 1). Compute the expected number of servings of all dishes over the next three days (starting May 2).

Solution: The transition matrix P of the Markov chain in Figure 10.3 is as follows:

$$P = \begin{bmatrix} & f & c & b & p \\ f & 0.0 & 0.6 & 0.4 & 0.0 \\ c & 0.7 & 0.0 & 0.0 & 0.3 \\ b & 0.5 & 0.5 & 0.0 & 0.0 \\ p & 0.5 & 0.5 & 0.0 & 0.0 \end{bmatrix}$$

The initial starting state is given by $\vec{\pi}^{(0)} = [0, 0, 1, 0]$. One can then successively compute the expected probability vector over the first three steps as follows:

$$\begin{aligned}\vec{\pi}^{(1)} &= \vec{\pi}^{(0)}P = [0.5, 0.5, 0, 0] \\ \vec{\pi}^{(2)} &= \vec{\pi}^{(1)}P = [0.35, 0.3, 0.2, 0.15] \\ \vec{\pi}^{(3)} &= \vec{\pi}^{(2)}P = [0.385, 0.385, 0.14, 0.09]\end{aligned}$$

Note that $\vec{\pi}^{(3)}$ is already beginning to come closer to the steady-state vector (as computed in Example 10.3). The expected number of hits \vec{h} is given by the sum of the vectors $\vec{\pi}^{(1)}, \vec{\pi}^{(2)},$ and $\vec{\pi}^{(3)}$ which is as follows:

$$\begin{aligned}\vec{h} &= \vec{\pi}^{(1)} + \vec{\pi}^{(2)} + \vec{\pi}^{(3)} \\ &= [1.235, 1.185, 0.34, 0.24]\end{aligned}$$

Note that the sum of the elements of \vec{h} is 3, because one is computing the expected number of hits over the next three servings. ■

10.2.2.1 Transitory Behavior with Probabilistic Termination

One problem with the use of a parameter like the number of steps, m , to control the trade-off between local connectivity and global connectivity is the fact that the approach is completely myopic about longer paths when small values of m are used. For example, using $m = 1$ will ignore even paths of length 2 in the computation of expected hit rates. On the other hand, using larger values of m deemphasizes local structure and assigns higher values to entries (i, j) in which node j has larger steady-state probabilities because of better global connectivity; in other words the starting node i will become less important at large values of m , where all rows of the expected hit matrix start becoming similar. Ideally, small values of m are needed, but one would somehow continue to incorporate the effect of longer paths in a limited way. Clearly, a different way of using path-length is required, so that a combination of small and long path lengths is required.

Instead of using the maximum number of steps as a free parameter, another approach is to use a continuation probability parameter $\alpha < 1$. The basic principle is that before each step, the Markov process is terminated with probability $1 - \alpha$. Otherwise, the Markov process is continued with probability α . Small values of α result in quick termination that favors local connectivity. Note that the number of steps (plus one for the termination step that is not actually counted) follows a geometric distribution with parameter $(1 - \alpha)$. Therefore, the number of steps M is a random variable here, and we have $E[M] + 1 = 1/(1 - \alpha)$ using results from the geometric distribution. In other words, using α as a parameter here is equivalent to using $\alpha/(1 - \alpha)$ as the equivalent value of m in the previous case. For example, using $\alpha = 0.5$ corresponds to $m = 1$; however, in this case, one is no longer myopic to longer paths, since they are accounted for during the processes where one gets lucky enough not to terminate too quickly. For example, at $\alpha = 0.5$ around 3% of the paths will have length , which is large enough for most real-world settings. In this case, one can show that the $n \times n$ matrix containing the number of hits H contains random variables as entries. The expected values of the entries of H is contained in the following matrix:

$$E[H] = \sum_{k=1}^{\infty} (\alpha P)^k$$

The main difference between this matrix and the m -step expected hit matrix is the fact that the summation is infinite here, because the decisions made during each step with parameter α control termination. Furthermore, the matrix P is multiplied with α because the expected values get scaled down by the termination probability in each step.

This summation can be shown to converge because the entries of $(\alpha P)^k$ go to zero rapidly for $\alpha < 1$ as k becomes large. Furthermore $(I - \alpha P)$ can be shown to be invertible for $\alpha < 1$. One trick that can be used to show convergence is to multiply the above matrix with the identity matrix written in the form $[(I - \alpha P)(I - \alpha P)^{-1}]$ to show the following:

$$\begin{aligned} E[H] &= \sum_{k=1}^{\infty} (\alpha P)^k = \left[\sum_{k=0}^{\infty} (\alpha P)^k \right] - I \\ &= \left[\sum_{k=0}^{\infty} (\alpha P)^k \right] [(I - \alpha P)(I - \alpha P)^{-1}] - I \\ &= \underbrace{\left[\sum_{k=0}^{\infty} (\alpha P)^k (I - \alpha P) \right]}_{I - (\alpha P)^{\infty} = I} [(I - \alpha P)^{-1}] - I \\ &= (I - \alpha P)^{-1} - I \end{aligned}$$

Interestingly, the matrix $(I - \alpha P)^{-1} - I$ contains the pairwise *Katz measures* of a Markov chain.

Definition 10.4 (Katz Measure) *The Katz measure for node pair $[i, j]$ and decay parameter α is the expected number of hits to state j when starting at state i and terminating with probability $(1 - \alpha)$ before each step. The $n \times n$ matrix K of Katz measures is given by the following:*

$$K = (I - \alpha P)^{-1} - I$$

The Katz measure is defined for any *adjacency matrix* representing connections in a network (e.g., social or Web network) and not just a transition matrix of a Markov chain. When applied to transition matrices, it becomes very similar to another measure referred to as *personalized PageRank* (cf. section 10.3).

The Katz measure is used frequently in machine learning to measure structural similarity among nodes in large networks (e.g., social networks), while accounting for the degree of local connectivity of the destination node with random-walk models. For example, the Katz measure between pairs of nodes in a tightly knit community of friends in a social network like Facebook will be very high (after converting the social network into a Markov chain with a random walk model). The notion of random-walk models for Web and social networks is discussed in greater detail in section 10.3. In the case of directed networks like Twitter, the Katz measure from i to j will generally be different than that between node j to i . This is particularly true when one of the nodes has a much larger number of followers than the other. For example, the Katz measure from a fan to a famous movie star will often be higher than that from the movie star to the fan. When lots of directed paths of small length exist from node i to node j , it will cause an increase in the Katz measure between nodes i and j . In such cases, the Katz measure between i and j is a combination of the degree of local connectivity of i and j , as well as a measure of the local *prestige* of node j .

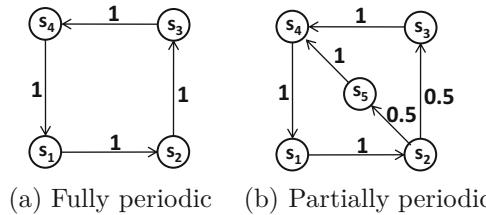


Figure 10.4: Periodic Markov chains

10.2.3 Periodic Markov Chains

The discussion thus far simply assumes that the underlying Markov chain has an existing steady-state probability distribution that is independent of starting state. There are cases in which a steady-state solution to the Markov chain may not even exist for one or more starting states. This can occur when the Markov chain exhibits *periodicity*. In a periodic Markov chain, the problem is that P^∞ does not exist, even though a vector $\vec{\pi}$ for which $\vec{\pi} = \vec{\pi}P$ is guaranteed by the Perron-Frobenius theorem [6]. Such a vector cannot be viewed as a steady-state probability vector and should be thought of in terms of the fraction of time spent in each state over the long term. We will elucidate this point with the help of an example.

Consider a 4-state Markov chain in which there is a cycle of four states as shown in Figure 10.4(a). In this case, a steady-state probability does not exist if a specific state is chosen as the starting state. This is because the current state of the Markov chain cycles among the four states and there is no *convergence* to a steady-state probability. Therefore, even though a solution corresponding to the probability vector $\vec{\pi} = [1/4, 1/4, 1/4, 1/4]$ does exist, it cannot be considered a steady-state probability given that convergence does not occur *for arbitrary starting states*. The only way to obtain steady-state is to set the starting distribution carefully to equal values for each state. This vector can also be thought of in terms of the fraction of time spent in each state over the long term, without actually reaching steady-state at any particular transition point. In fact, if one starts at state s_1 , the Markov chain will deterministically be in state s_1 after $4i$ transitions for any integer i . Similarly, it will periodically visit states s_2, s_3 , and s_4 without reaching convergence.

Beyond the obvious cases of periodicity (such as Figure 10.4(a)), there are other versions of periodicity that are more subtle. The Markov chain in Figure 10.4(b) is a case of partial periodicity. In this case, one can partition the states into four sets $S_1 = \{s_1\}$, $S_2 = \{s_2\}$, $S_3 = \{s_3, s_5\}$ and $S_4 = \{s_4\}$. The Markov chain continually cycles through these four sets without reaching steady state. As a result, P^k will oscillate with a period of 4.

Example 10.5 What is the transition matrix of a Markov chain of two states deterministically transitioning into one another with edges of probability 1? Show that P^k oscillates with increasing k and therefore does not converge as $k \rightarrow \infty$. Is this Markov chain ergodic?

Solution: In this case, the transition matrix P is as follows:

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

In this case, it can be shown that P^k flips between P and the identity matrix I , as k is increased. In other words, P^k is equal to P for odd k and it is equal to the identity matrix for even k . Therefore, P^k does not converge with increasing k and the underlying Markov chain is not ergodic because of periodicity in the transitions. ■

Example 10.6 Write the transition matrix for the Markov chain of Figure 10.4(b). What is the unique solution to $\vec{\pi} = \vec{\pi}P$? How would you interpret this solution, considering the fact that periodic Markov chains do not have a steady-state in general?

Solution: The transition matrix for state order $[s_1, s_2, s_3, s_4, s_5]$ is as follows:

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Note that one can solve the eigenvector equation in order to find the solution. However, one can also find this vector $\vec{\pi}$ by calculating the fraction of time spent in each state. The Markov chain spends 0.25 fraction of the time in sets S_1 , S_2 , S_3 and S_4 , where $S_1 = \{s_1\}$, $S_2 = \{s_2\}$, $S_3 = \{s_3, s_5\}$ and $S_4 = \{s_4\}$, since these sets create the cycle. Furthermore, the time in set S_3 is split equally into s_3 and s_5 because of symmetry. Therefore, the vector $\vec{\pi}$ is $[0.25, 0.25, 0.125, 0.25, 0.125]$. This vector cannot be considered a steady-state probability, because the probability of a state depends on the number of steps since the beginning of the process. This vector should be considered the fraction of time spent in each state over the long term. Furthermore, if the Markov chain is started with this probability distribution, it will maintain this probability distribution. ■

Problem 10.3 Write the transition matrix for the Markov chain in Figure 10.4(a). Compute P^2 , P^3 , P^4 , P^5 , and confirm that P^k oscillates between four different matrices for increasing k .

Markov chains that do not have any type of periodicity are said to be *aperiodic*. It is a necessary condition for a Markov chain to be aperiodic in order for it to have a steady-state probability.

10.2.4 Ergodicity

For a Markov chain to be ergodic, a steady-state probability distribution must exist (i.e., the chain is aperiodic), and it must be independent of starting state. The Perron-Frobenius theorem [6] states that at least one solution exists to $\vec{\pi} = \vec{\pi}P$ for a transition matrix P . This does not mean that the steady-state probability vector is unique (irrespective of starting state) or that a steady-state probability vector exists for every starting configuration of probabilities. Multiple probability vectors $\vec{\pi}$ satisfying $\vec{\pi} = \vec{\pi}P$ can occur in cases like Figure 10.1(b) in which both the solutions $[1, 0, 0, 0]$ and $[0, 0, 1, 0]$ satisfy the aforementioned condition. The solution to the above system of equations is unique if and only if the above matrix P has a single nonnegative left eigenvector with eigenvalue 1.

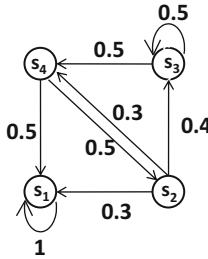


Figure 10.5: A Markov chain with a unique steady-state probability that is not ergodic.

The notion of *ergodicity* is defined by the nature of the steady-state probabilities in Markov chains:

Definition 10.5 (Ergodic Markov Chain) *An ergodic Markov chain is one in which (a) a steady-state probability distribution of states exists in the Markov chain that is independent of starting state, and (b) no state has zero steady-state probability.*

The second condition ensures that no state is *transient*. Formally, a state is referred to as transient, if a Markov chain starting in that state will eventually never return to that state after an initial period in which it reaches a stationary distribution. Consider a snakes-and-ladders game (see Example 10.1 and Figure 10.2) in which a single player is rolling a six-sided die to move between one of a hundred numbered squares (with correspondingly numbered states). The underlying Markov chain always reaches the self-loop of the “finish” state (square 100) over a long period of time, even if one starts the game from an arbitrary point. Therefore, a steady-state probability vector exists and is unique — it contains 99 zeros and a single entry of one for the finishing state. However, it is not considered an ergodic Markov chain because every state other than the “finish” state has a steady-state probability of 0. Ergodic Markov chains are not allowed to have *transient* states.

It turns out that the necessary and sufficient conditions for a Markov chain to be ergodic are for it to have two-way reachability between every state-pair and for it to be aperiodic:

Lemma 10.1 (Conditions for Ergodicity) *A Markov chain is ergodic, if and only if (i) it is possible to reach any state from any other state using at least one directed path of positive probabilities, and (ii) the states are not partitioned into k cyclically connected sets. In other words, no partition of the states into sets $S_1 \dots S_k$ exists so that each state transition must occur cyclically from a state in the i th set S_i to a state in the $([i \bmod k] + 1)$ th set $S_{([i \bmod k] + 1)}$.*

The underlying graph structures that satisfy the property of mutual reachability between every pair of nodes are referred to as *strongly connected*. An example of an ergodic Markov chain is shown in Figure 10.1(a). In this Markov chain, every pair of nodes exhibits mutual reachability and there are no periodic states. On the other hand, it is not possible to reach from either state s_1 or state s_3 to any other state in Figure 10.1(b). Therefore, the Markov chain in Figure 10.1(b) is not strongly connected or ergodic. The lack of strong connectivity precludes ergodicity even when a unique steady-state vector exists (because it causes transient states). An example of a Markov chain that has a unique steady-state solution (irrespective of starting state) but is not ergodic is shown in Figure 10.5. The Markov chain is not ergodic because the structural condition of strong connectivity is not satisfied in Figure 10.5 — it is not possible to reach any state from state s_1 , and all other

states are transient with zero steady-state probabilities. On the other hand, the unique steady-state vector that is reached irrespective of starting state is $[1, 0, 0, 0]$. The Markov chain for the one-player version of snakes and ladders also has a unique steady-state vector. However, it is not ergodic because of the presence of transient states.

A state that is not transient is referred to as *recurrent*. Formally, a state s_i is referred to as recurrent, if the Markov chain will return to s_i infinitely often, given that it starts in s_i . States such as s_1 in Figure 10.5 are recurrent, because they repeat again and again over the long-term as long as the Markov chain starts in that state. In general, a recurrent state will have non-zero steady-state probability for at least one starting state for aperiodic Markov chains. In the case of Figure 10.1(a), all states are recurrent. On the other hand, states s_1 and s_3 are recurrent in Figure 10.1(b), whereas states s_2 and s_4 are transient. States with self-loops of probability 1 are referred to as *absorbing states*.

Example 10.7 For each of the Markov chains in Figure 10.4 and Figure 10.5, which states are recurrent, transient, and absorbing?

Solution: In Figure 10.4(a), all states are recurrent and none are transient. There are no absorbing states.

In Figure 10.4(b), all states are recurrent. There are no absorbing or transient states. In Figure 10.5, the state s_1 is both recurrent and absorbing. All other states are transient. ■

The aforementioned discussion assumes that strongly connected Markov chains do not have transient states, although this result was not formally proven. For completeness, we provide a proof sketch of this result:

Lemma 10.2 If every state is reachable from every other state by a directed path in a Markov chain (and the transition matrix is aperiodic), then all steady-state probabilities must be non-zero.

Proof: Consider a Markov chain that is ergodic with steady-state probability vector $\vec{\pi}$, but there is a subset of states Q that are such that $\pi_i = 0$ for all $s_i \in Q$ and $\pi \neq 0$ for $s_i \notin Q$. Consider the following steady state condition for each $s_i \in Q$:

$$\pi_i = \sum_{j=1}^n \pi_j p_{ji} \quad \forall s_i \in Q$$

Since each π_i for $s_i \in Q$ is 0, one can simplify the above expression as follows:

$$0 = \sum_{j:s_j \notin Q} \pi_j p_{ji} \quad \forall s_i \in Q$$

Summing up this condition for each $s_i \in Q$, one obtains the following:

$$\sum_{i,j: s_i \in Q, s_j \notin Q} \pi_j p_{ji} = 0 \tag{10.2}$$

However, this is a contradiction because at least one of the terms on the left-hand side is a positive quantity in a strongly connected graph. In a strongly connected graph, paths of nonzero probability edges will exist from all states outside Q to states inside Q ; traversing

any such path will also lead to the traversal of at least one edge from $s_j \notin Q$ to $s_i \in Q$. Corresponding to that edge (j, i) , the value of $\pi_j p_{ji}$ will be positive, causing the left-hand side of Equation 10.2 to become positive. However, the right-hand side of the above equation is 0. The only way in which the above expression can hold true is if set Q of zero probability states is empty. In other words, every state must have non-zero steady-state probability.

■

10.2.4.1 Alternative Characterization of Ergodicity

For a Markov chain with transition probability matrix P to have a unique steady-state solution that is independent of starting state, P^k must converge with increasing k to a matrix in which every row is the same and has positive elements. As discussed earlier in this chapter, raising the transition matrix P to the k th power results in a transition matrix of a Markov chain in which k transitions of the original Markov chain with transition matrix P correspond to a single transition of the new Markov chain corresponding to the transition matrix P^k . Therefore, P^k corresponds to the k -step transition matrix. What happens in the case when $k = \infty$? In the case where the Markov chain is ergodic, $\lim_{k \rightarrow \infty} P^k$ must exist and converge to a matrix P^∞ in which (i) every entry is positive, and (ii) every row is equal to the unique steady-state probability vector $\vec{\pi}^*$. For such a matrix, it can be shown that for any starting state vector $\vec{\pi}^{(0)}$, the following holds:

$$\vec{\pi}^{(0)} P^\infty = \vec{\pi}^*$$

The above result holds because multiplying $\vec{\pi}^{(0)}$ with P^∞ creates a convex combination of the rows of P^∞ , all of which are $\vec{\pi}^*$. This probability vector $\vec{\pi}^*$ is the steady-state vector of the Markov chain that is independent of the starting state. One can use the aforementioned observations to propose the following alternative characterization of an ergodic Markov chain:

Lemma 10.3 (Ergodic Markov Chain: Alternative Conditions) *A Markov chain is ergodic if and only if every element of P^k is positive for some k . Furthermore, P^r will always be positive for each $r > k$.*

10.2.5 Different Cases of Ergodicity and Non-Ergodicity

Ergodicity can be quickly identified using the graph structure of the Markov chain. Based on the discussion thus far, one can summarize the behavior of transition matrices as follows:

1. If P is a transition matrix of an ergodic Markov chain (i.e., aperiodic and strongly connected Markov chain), then P^∞ exists (i.e., P^k exhibits convergence with increasing k) and will have identical rows with all positive elements. The i th row contains the steady-state vector starting from state i , which implies that the steady-state vector is independent of starting state.
2. If P is a transition matrix of a non-ergodic Markov chain that is aperiodic but not strongly connected, then P^∞ exists but some of its elements will be 0 (signifying the existence of transient states). Examples are shown in Figures 10.1(b) and 10.5. It is also possible for the rows of P^∞ to be non-identical (signifying the existence of non-unique steady-state vectors). The Markov chain of Figure 10.1(b) has a non-unique steady-state vector in addition to having transient states. On the other hand, the Markov chain of Figure 10.5 has transient states s_2 , s_3 , and s_4 but its steady-state probability vector $[1, 0, 0, 0]$ is unique.

3. If P is a transition matrix of a non-ergodic Markov chain that is periodic, then P^∞ does not exist irrespective of whether the Markov chain is strongly connected. In other words, P^k does not converge with increasing k and oscillates between different possibilities. Examples of such Markov chains are shown in Figures 10.4(a) and 10.4(b).

The second case above with non-unique steady-state vectors is particularly interesting because many decision-making applications require one to determine the probabilities of final outcomes, starting from an initial configuration. This is useful, because if certain outcomes are desirable, one can choose to start from initial configurations that probabilistically favor such outcomes. The next section will discuss the structure of such Markov chains in detail.

Example 10.8 Write the transition matrix for the non-ergodic Markov chain in Figure 10.1(b). Using a matrix power calculator on the internet, estimate P^∞ by calculating P^{1000} . Comment on the rows of this matrix.

Solution: The transition matrix for the Markov chain in Figure 10.1(b) is as follows:

$$P = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.3 & 0.0 & 0.4 & 0.3 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.5 & 0.2 & 0.3 & 0.0 \end{bmatrix}$$

The steady-state transition matrix P^∞ is estimated using an online matrix power calculator:

$$P^\infty \approx P^{1000} \approx \begin{bmatrix} 1.000 & 0.000 & 0.000 & 0.000 \\ 0.479 & 0.000 & 0.521 & 0.000 \\ 0.000 & 0.000 & 1.000 & 0.000 \\ 0.596 & 0.000 & 0.404 & 0.000 \end{bmatrix}$$

It is evident that the two absorbing states corresponding to s_1 and s_3 have rows in P^∞ indicating that the Markov chain ends in those states when starting in those states. The more interesting cases correspond to the two transient states s_2 and s_4 . Starting in these states always results in one of the absorbing states s_1 and s_3 , as is evident from the second and fourth rows of P^∞ . ■

Problem 10.4 Construct a non-ergodic Markov chain in which the steady-state probability depends on the starting state, and no state has a steady-state probability of 1, irrespective of the starting state.

10.2.6 Properties and Applications of Non-Ergodic Markov Chains

Several machine learning applications like *vertex classification on graphs* work with non-ergodic Markov chains (that are aperiodic but not strongly connected). Therefore, it is useful to examine some properties of such aperiodic Markov chains lacking strong connectivity. Such Markov chains satisfy the following properties:

1. States are either transient (with zero steady-state probabilities) or recurrent (with nonzero steady-state probabilities). It is impossible for a transition to occur from a recurrent state to a transient state. Otherwise, a mathematical contradiction can be reached by showing a nonzero steady-state probability of the transient state by using the steady-state equations.

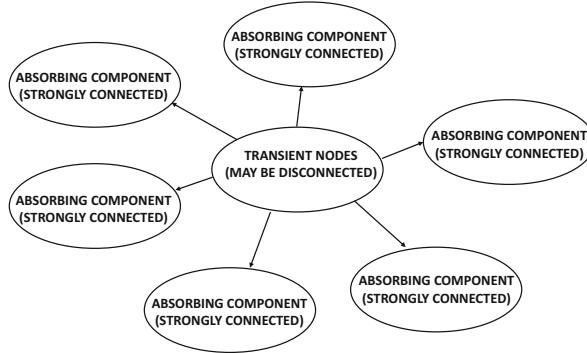


Figure 10.6: The structure of a non-ergodic Markov chain lacking strong connectivity. The transient set need not be connected even in a connected Markov chain. A non-ergodic Markov chain with only one absorbing component has a unique steady-state vector.

2. If recurrent state i is reachable from state j , then recurrent state j must be reachable from i (or else a contradiction can be reached by showing that state j does not meet the definition of recurrence by failing to return to itself with probability 1). Therefore, pairs of recurrent states are either strongly connected or mutually unreachable. In other words, recurrent states can be partitioned into strongly connected components, and there are no transitions among these different partitions. These components are referred to as *absorbing components*.

The two aforementioned observations imply that the states of a Markov chain that is not strongly connected can be partitioned into a transient set and multiple recurrent sets. Transitions across partitions only occur from transient sets to recurrent sets. Such a Markov chain always appears in the form of Figure 10.6. Note that the transient vertices may or may not be strongly connected (or even connected) *among themselves*, based on *only* the subgraph obtained by dropping all absorbing components and their incident edges. However, the entire Markov chain will still be connected (but not strongly connected) through connections to absorbing components. For example, the transient vertices in Figure 10.7(b) (which are labeled by ‘T’) are not connected to one another, when considered as a subgraph. Each of the satellite components in Figure 10.6 is an absorbing component. If a random walk is performed on the full graph, and the random walk happens to enter an absorbing component, the walk will never exit the component. It is possible for a non-ergodic Markov chain to contain only one absorbing component, as long as some of the vertices are transient. If a Markov chain does not have any transient nodes, it is also strongly connected.

The structure of connections among the transient set is irrelevant and the only key property they satisfy is that they are unreachable from vertices outside this component. Therefore, once a random walk exists the transient set, it cannot be entered again. For example, in the case of Figure 10.7(a), vertex 9 is the only transient vertex. On the other hand, in Figure 10.7(b), the vertices 1, 4, 5, 6, and 7 are transient vertices. The transient vertices are labeled by ‘T’ in these figures. The reader should take a moment to verify that a random walk starting at any of these vertices will eventually reach a vertex outside this set so that it becomes impossible to ever visit any of these vertices again. Furthermore, these vertices do not even form a connected graph if the non-transient nodes are dropped.

Each node in an absorbing component has a non-zero probability to be visited in steady-state from a random walk that starts at a node within the component. However, if the walk

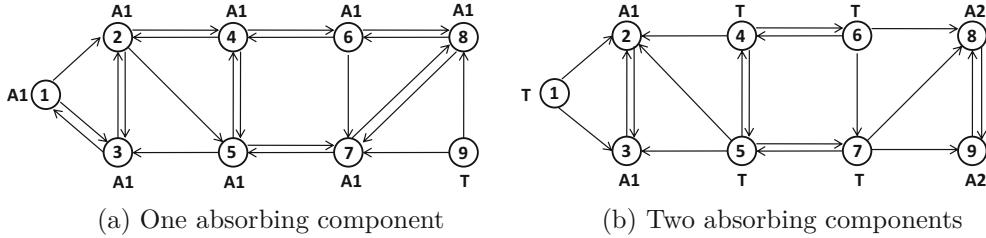


Figure 10.7: Examples of directed graphs that are not strongly connected. Transient vertices are labeled ‘T’ and vertices belonging to absorbing components are labeled ‘A1’ and ‘A2’.

starts at a transient node, the steady-state probability of a node in an absorbing component could be either zero or nonzero, depending on the vertex at which the random walk starts. In Figure 10.7(a), there is a single absorbing component containing all vertices except vertex 9. On the other hand, Figure 10.7(b) contains two absorbing components, which are labeled by ‘A1’ and ‘A2’. Note that starting the random walk at vertex 1 will always reach absorbing component A1, but will never reach absorbing component A2. Starting the walk at vertex 4 will allow both A1 and A2 to be reached.

Next, we examine the structure of the transition matrix for Markov chains of the form shown in Figure 10.6. Without loss of generality, we assume that the vertices of such a graph are ordered as follows. The first block of this graph contains all the transient vertices. For each transient vertex i in this set, the *outgoing* transition probability p_{ij} can be non-zero for all j from 1 to n . All other l absorbing components are arranged in block diagonal form. For example, a graph with a transient set and $l = 3$ absorbing components will have the following block structure of the transition matrix on appropriate reordering the vertex indices (to put the transient vertices first and the vertices of the absorbing components contiguously in succession):

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ 0 & P_{22} & 0 & 0 \\ 0 & 0 & P_{33} & 0 \\ 0 & 0 & 0 & P_{44} \end{bmatrix} \quad (10.3)$$

It is important to note that the square blocks P_{22} , P_{33} , and P_{44} correspond to the edges within absorbing components, and are complete probabilistic transition matrices whose rows sum to 1. These submatrices correspond to ergodic Markov chains and the steady-state probabilities obtained by using an initial state in one of the absorbing components can be computed independently of other components. Specifically, one can compute the principal left eigenvectors of each of the blocks P_{22} , P_{33} , and P_{44} (corresponding to the absorbing components) separately. Each such eigenvector of the absorbing submatrix defines one of the non-unique steady-state probability vectors of the entire Markov chain (in which the probabilities of states in other components are 0).

In general, consider a graph of the form shown in Figure 10.6 with one transient component and three strongly connected components. In such a case, the transition matrix has the block form shown in Equation 10.3. One can easily generalize the basic structure of this transition matrix to any number of strongly connected components (rather than three components). The Markov chain cannot be ergodic as long as there is at least one transient state. However, as long as P is aperiodic, the matrix P^∞ exists and can be shown to be the

following:

$$P^\infty = \begin{bmatrix} 0 & (I - P_{11})^{-1}P_{12}P_{22}^\infty & (I - P_{11})^{-1}P_{13}P_{33}^\infty & (I - P_{11})^{-1}P_{14}P_{44}^\infty \\ 0 & P_{22}^\infty & 0 & 0 \\ 0 & 0 & P_{33}^\infty & 0 \\ 0 & 0 & 0 & P_{44}^\infty \end{bmatrix} \quad (10.4)$$

It is noteworthy that the top-left corner block of the above matrix contains the zero matrix because P_{11}^∞ reduces to the zero matrix. The informal justification for this claim is that after a long period of time, any Markov process starting from this transient vertex set moves to one of the strongly connected absorbing components and therefore the probability of remaining in the transient set is 0. The matrix P_{11}^∞ contains the pair-wise probabilities of remaining in a transient node after starting in a transient node.

Lemma 10.4 *Let P_{11} be the submatrix of the transition matrix containing only the transient nodes. Then, the matrix P_{11}^∞ is the zero matrix.*

Furthermore, the expression $(I - P_{11})^{-1}P_{1m}P_{mm}^\infty$ for component m denotes the matrix of long-term probabilities of moving from states in the transient set to states in the m th absorbing component. This probability matrix is obtained by summing up (over all k) the probability of k transitions within the transient set followed by a transition out of it into the m th absorbing component and subsequent steady-state transitions within the m th component. This summation is replicated below with a corresponding simplification:

$$\sum_{k=0}^{\infty} P_{11}^k P_{1m} P_{mm}^\infty = (I - P_{11})^{-1} P_{1m} P_{mm}^\infty$$

The correctness of the above relationship can be shown by premultiplying both sides with $(I - P_{11})$ and using the fact that $(I - P_{11}^\infty) = I$, since P_{11}^∞ yields the zero matrix. The submatrix P_{11} is very useful because it regulates the transient behavior of the Markov chain. It will subsequently be referred to as P_{tran} .

Non-ergodic Markov chains are very useful for modeling various aspects of a Markov chain, such as the probability of hitting a particular state for the first time within k steps (starting from an initial state) and the time taken to reach a particular state for the first time. All these quantities can be evaluated using non-ergodic extensions of the transient characteristics of Markov chains discussed in section 10.2.2. A particularly interesting application of these results is the *expected transience time* in Markov chains:

Definition 10.6 (Expected Transience Time) *The expected transience time for starting node s_i is the expected number of steps that an aperiodic and non-ergodic Markov chain spends in a transient state before reaching an absorbing component. When starting in a transient state, its initial state is counted in the time spent.*

The expected transience time is at least 1 when starting in a transient state because the initial state is counted in the transience time. It makes sense to compute the expected transience time only for starting states that are transient. Otherwise, the transience time starting from an absorbing component is exactly 0. One can show the following for the computation of the expected transience time:

Lemma 10.5 (Computing Expected Transience Time) *Let P_{tran} be the submatrix of the stochastic transition matrix containing only transient states. Then, starting at transient state s_i , the expected transience time is the sum of the elements in the row corresponding to s_i in $(I - P_{tran})^{-1}$.*

Proof: The probability that one moves from transient state s_i to transient state s_j in exactly k steps is the (i, j) th entry is P^k . Therefore, the expected time spent in state s_j starting from state s_i (over all possible number of steps) is given by the (i, j) th entry in $\sum_k P_{tran}^k$. The summation can be simplified⁴ because P_{tran}^∞ can be shown to be the zero matrix:

$$\sum_{k=1}^{\infty} P_{tran}^k = \underbrace{\left[\sum_{k=1}^{\infty} P_{tran}^k (I - P_{tran}) \right]}_I (I - P_{tran})^{-1} = (I - P_{tran})^{-1}$$

Therefore, summing up the i th row of the above matrix yields the expected transience time. ■

One can use non-ergodic Markov chains to compute various other useful probabilities and expected quantities. The following examples provide a flavor of some of these applications:

Example 10.9 Consider the Markov chain of Figure 10.1(b) on page 438. Calculate the expected transience time starting at each of the two transient states s_2 and s_4 .

Solution: The 2×2 matrix P_{tran} with rows corresponding to s_2 and s_4 may be composed as follows:

$$P_{tran} = \begin{bmatrix} 0 & 0.3 \\ 0.2 & 0 \end{bmatrix} \quad (I - P_{tran}) = \begin{bmatrix} 1 & -0.3 \\ -0.2 & 1 \end{bmatrix}$$

The inverse of the 2×2 matrix $(I - P_{tran})$ may be computed by swapping the diagonal entries, negating the off-diagonal entries, and dividing each entry with the determinant $1 \times 1 - (-0.3)(-0.2) = 0.94$. Therefore, the inverse is as follows:

$$(I - P_{tran})^{-1} = \frac{1}{0.94} \begin{bmatrix} 1 & 0.3 \\ 0.2 & 1 \end{bmatrix}$$

Therefore, the expected transience time starting in state s_2 is $(1 + 0.3)/0.94 \approx 1.38$, and the expected transience time starting in state s_4 is $(1 + 0.2)/0.94 \approx 1.28$. ■

Example 10.10 Consider the Markov chain of Figure 10.5 on page 448. Calculate the expected transience time starting at each of the two transient states s_2 , s_3 , and s_4 .

Solution: The 3×3 matrix P_{tran} with rows corresponding to s_2 , s_3 , and s_4 (in that order) may be composed as follows:

$$P_{tran} = \begin{bmatrix} 0 & 0.4 & 0.3 \\ 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0 \end{bmatrix} \quad (I - P_{tran}) = \begin{bmatrix} 1 & -0.4 & -0.3 \\ 0 & 0.5 & -0.5 \\ -0.5 & 0 & 1 \end{bmatrix}$$

One can compute the inverse of this matrix using any online matrix calculator:

$$(I - P_{tran})^{-1} = \begin{bmatrix} 1.538 & 1.231 & 1.077 \\ 0.769 & 2.615 & 1.538 \\ 0.769 & 0.615 & 1.538 \end{bmatrix}$$

⁴This identity is the generalization of the scalar identity $(1 - x)^{-1} = \sum_{i=0}^{\infty} x^i$ for $|x| < 1$.

Summing up the rows of the above matrix, the expected transience time starting in state s_2 is $1.538 + 1.231 + 1.077 \approx 3.85$, and the expected transience time starting in state s_3 is $0.769 + 2.615 + 1.538 \approx 4.92$. The expected transience time starting in state s_4 is $0.769 + 0.615 + 1.538 \approx 2.92$. ■

Example 10.11 (Expected Hitting Time) Suppose you have an ergodic Markov chain with n states, and you want to calculate the expected number of transitions it takes to first hit state s_j starting from state s_i . Outline the steps you would use to calculate the expected hitting time.

Solution: The Markov chain should first be converted to a non-ergodic chain by deleting outgoing edges from s_j and adding a self-loop with probability 1. The $(n - 1) \times (n - 1)$ transient matrix P_{tran} can be computed that excludes state s_j . The expected hitting time to state s_j from state s_i is exactly equal to the expected transience time starting at state s_i . Therefore, the row for state s_i in $(I - P_{tran})^{-1}$ is summed and reported as the expected hitting time. ■

Example 10.12 (Termination Probability of Snakes-and-Ladders)

Consider the one-player version of snakes-and-ladders discussed in Example 10.1. Compute the probability that the game ends within 75 steps beginning from the starting state.

Solution: A 100-state Markov chain is constructed as discussed in Example 10.1. Note that the finishing state is absorbing and therefore the Markov chain is not ergodic. The entry $(1, 100)$ in the 75-step transition matrix P^{75} provides the probability of termination in 75 moves or less. A more efficient approach would be to compute $\vec{e}_1 P^{75}$ using repeated row-to-matrix multiplication. Here, \vec{e}_1 (starting state vector) is a 100-dimensional row vector with a single 1 in the first entry and 0s in all other entries. The 100th entry of the resulting row vector (i.e., $\vec{e}_1 P^{75}$) will provide the desired probability. It is noteworthy that the presence of a self-loop at the finishing state ensures that if the Markov chain reaches that state in less than 75 moves, it will continue to stay at that state in the remaining transitions to 75 moves. ■

Example 10.13 (Expected Termination Time of Snakes-and-Ladders)

Consider the one-player version of snakes-and-ladders discussed in Example 10.1. Compute the expected time the game lasts beginning from the starting state.

Solution: A 100-state Markov chain is constructed as discussed in Example 10.1. This Markov chain has a single absorbing state and therefore its expected transience time from the starting state is equal to the expected time the game lasts. According to Lemma 10.5, this transience time is equal to the sum of the elements of the row of $(I - P_{tran})^{-1}$ corresponding to the starting state. Note that P_{tran} is a 99×99 matrix including all states of snakes and ladders except for the ending state. ■

Example 10.14 Consider a Markov chain of the type of Figure 10.6 in which the transient component contains a single node and you are given (i) the transition probability of each edge from the single transient node to all other nodes and the identity of the strongly connected component to which each of those recurrent nodes belong, and (ii) the steady-state probability vector $\vec{\pi}_j$ if one were to start the Markov process in component j . You are not given any other information about the structure of each strongly connected component. How will you calculate the steady-state probability of each node when the starting node is the transient node?

Solution: Let $c_1 \dots c_k$ be identities of the different strongly connected components. Let $P(i|C = c_j)$ be the steady state probability of node i if the first node to which the walk reaches belongs to component c_j . Note that this value is simply the i th component of $\vec{\pi}_j$. Let $P(C = c_j)$ be the probability that the Markov chain eventually enters component j when one starts at the single transient node. Then, using the total probability rule, the steady state probability of node i is given by the following:

$$\pi(i) = \sum_{j=1}^k P(C = c_j)P(i|C = c_j) \quad \forall i \in \{1 \dots n\}$$

One can also express the above in vector notation:

$$\vec{\pi} = \sum_{j=1}^k P(C = c_j)\vec{\pi}_j$$

The value of $P(C = c_j)$ can be evaluated as the sum of the probabilities of the edges from the transient node to a node in component c_j . ■

Example 10.15 Suppose that you are given the $n \times n$ transition matrix P of a non-ergodic Markov chain defined over states $\{s_1, \dots, s_n\}$. The Markov chain is aperiodic but not strongly connected. Therefore, for a given starting state, it does have a steady-state probability distribution that is specific to the starting state. The n starting states are sampled with probabilities $q_1 \dots q_n$. After a long period of transitions from the sampled starting state, the current state of the chain is sampled and determined to be the m th state s_m . Given the known identity of the final state s_m , outline a procedure to determine the conditional probability that the starting state was s_i for each i .

Solution: It is assumed that the Markov chain reaches steady state after a long period. The first step is to estimate P^∞ by squaring P and repeating the process for about 20 times (which results in P^{20}). Let S be the random variable representing the starting state and E be the random variable representing the state that was sampled at the end. The (i, m) th entry of P^∞ yields $P(E = s_m | S = s_i)$, whereas the desired probability to be computed is $P(S = s_i | E = s_m)$. Therefore, using the Bayes rule, one obtains the following:

$$P(S = s_i | E = s_m) = \frac{P(S = s_i)P(E = s_m | S = s_i)}{\sum_{j=1}^n P(S = s_j)P(E = s_m | S = s_j)} = \frac{q_i P(E = s_m | S = s_i)}{\sum_{j=1}^n q_j P(E = s_m | S = s_j)}$$

The reader is also encouraged to work out Exercises 8 and 11, which are similar to the above problem. ■

Problem 10.5 Can you simplify your answer to Problem 10.15, if the Markov chain were both aperiodic and strongly connected (and therefore ergodic)?

A hint for solving the above problem is ask yourself about the conditions in which the probability vectors for the starting state and stationary distribution are dependent. Does any simplification happen under the independence assumption?

10.2.7 Probabilities of Absorbing Outcomes

A particularly interesting case arises, where each absorbing component in Figure 10.6 is a single state with a self loop, and all states without self-loops are transient. One can view each such absorbing state as an outcome from the discrete state Markov process. Furthermore, the transient states correspond to various initial and intermediate configurations. Depending on the initial configuration that one chooses, different outcomes may be more or less likely. This type of model can be very useful for (i) deciding which initial configuration of a decision-making scenario (i.e., state) is likely to lead to a desired outcome, (ii) comparing the probability of each possible termination outcome based on current state, or (iii) comparing two Markov chains with the same states but different topology in terms of the probability of reaching a desired outcome from the same initial state. At the end of this section, a specific example in the context of the snakes-and-ladders game is provided (see Example 10.17).

Consider a non-ergodic and aperiodic Markov chain with n states s_1, s_2, \dots, s_n of which the first $r < n$ states s_1, s_2, \dots, s_r are transient states and all other states s_{r+1}, \dots, s_n are absorbing (outcome) states. How can one determine the probabilities of various outcomes, given a particular starting state s_p for $p < r$? It turns out that the $r \times (n - r)$ submatrix of P^∞ for all r rows corresponding to transient states and $(n - r)$ columns corresponding to absorbing (outcome) states contains all the information that we need. Specifically, each entry of this matrix contains the probability that one reaches a particular absorbing state when starting from a particular transient state. It is also noteworthy that the matrix P^∞ in this case is a special case of the structure of the matrix in Equation 10.4. In the following we compute a simplified expression for this case.

Let \vec{e}_i be an r -dimensional row vector containing a value of 1 at the i th position (corresponding to starting state) and 0 in all other positions. Let P_{tran} be the $r \times r$ submatrix of the transition matrix containing only the transient states. Let \vec{p}_m be the r -dimensional column vector containing the transition probabilities from the r transient states to state m . One can compute the steady-state probability of reaching the m th state (for $m > r$) for starting state $i \leq r$ as follows:

$$\begin{aligned} P(\text{Termination at state } m) &= \sum_{k=0}^{\infty} P(k \text{ transitions in transient set before termination at } m) \\ &= \sum_{k=0}^{\infty} \vec{e}_i P_{tran}^k \vec{p}_m \\ &= \vec{e}_i (I - P_{tran})^{-1} \vec{p}_m \end{aligned}$$

Since the row vector \vec{e}_i contains all zeros except for a single 1 in the i th position, it pulls out the i th row from the vector $(I - P_{tran})^{-1} \vec{p}_m$. Instead of pulling out a specific row, one

can write the entire r -dimensional vector of probabilities \vec{q}_m for probability of termination at state m (starting at each of the r transient states) as follows:

$$\vec{q}_m = (I - P_{\text{tran}})^{-1} \vec{p}_m$$

The reader is encouraged to verify that the above equation is in the same form as the entries in the first row of the matrix in Equation 10.4; the main difference is that the matrix P_{1m} in Equation 10.4 is replaced with the vector \vec{p}_m and each occurrence of a steady-state transition matrix P_{mm}^∞ for a self-loop is a 1×1 matrix containing the scalar 1.

Interestingly, the above non-ergodic Markov chain model can also be used to obtain the probability of *first hit* in an ergodic Markov chain with a subset of competing states. Several examples of the utility of the first-hit paradigm are discussed in section 10.3 for vertex classification problems in graph machine learning.

Example 10.16 (First Hit Probability in Ergodic Markov Chain) Consider an ergodic Markov chain in which the goal is to compute the probability of a first hit among t competing states when starting at state s_i . Therefore, we would like to find a t -dimensional probability vector that sums to 1. The i th component of the vector provides the probability that the i th state will be hit first in the Markov process.

Solution: A key observation is that transitions out of competing states are irrelevant, since one is only interested in the behavior of the Markov chain until the first hit. Therefore, the ergodic Markov chain can be transformed into a related non-ergodic one containing absorbing states. The outgoing transitions from competing states are removed and replaced with self-loops of probability 1. Let P be the transition matrix of this Markov chain and P_{tran} be its submatrix of non-absorbing states. The vector \vec{p}_m contains the probabilities of transitions from each non-absorbing state to the m th absorbing state. Then, the probability of a first hit at the m th absorbing state (starting at the i th state s_i) is given by $\vec{e}_i(I - P_{\text{tran}})^{-1} \vec{p}_m$. ■

In order to provide another example of the usefulness of this type of outcome modeling, we provide an example that computes the probability of winning from a particular intermediate position in the snakes-and-ladders game:

Example 10.17 (Probability of Winning in Snakes and Ladders) Consider the snakes-and-ladders game shown in Figure 10.2 with two players A and B alternately throwing six-sided dice. The game has 99 intermediate positions (including the starting state) on the board along with a final position corresponding to square 100, reaching which causes one of the players to win. Assume that the board layout in terms of the locations of the snakes and ladders is known. Player A is currently at square 38 and player B is currently at square 34. Player B will move next. Outline the details of a computational procedure that uses a Markov chain to compute the probability that player A wins. You will have to describe the Markov chain, its states, and whether the states are transient or recurrent. Then, describe the specific computational procedure to determine the probability that each player wins from that position in the game.

Solution: In this case, a Markov chain can be created with $99 \times 99 \times 2$ intermediate states corresponding to the combinations of square numbers in which the two players

find themselves as well as whose turn it is to currently play. Any transition from a state where player A is to move next results in a state where player B is to move next (and vice versa). For example, the state $(38, 34, B)$ refers to the fact that player A is on square 38, player B is on square 34, and that player B is to move next. In addition, there are two special recurrent states called *A-wins* and *B-wins*, representing the victory of one of the two players. These are the only non-transient states and are modeled as absorbing states with self-loops. Therefore, this Markov chain has a total of 19,604 states of which 2 states are recurrent/absorbing and the remaining 19,602 states are transient states.

Each transient state typically has six outgoing edges with probability $1/6$ representing the six outcomes of a throw of the die. A transition from one state to another not only includes the effect of the squares moved that are equivalent to the face value of the die roll but also the effect of the snake or ladder on the landing square. Therefore, transitions may sometimes occur between states where square numbers are more than six apart. For some states near the finish state large die outcomes might overshoot and are treated equivalently with a face value of 0. Such states may have self-loops and fewer than six outgoing edges. Furthermore, landing on the finishing square results in either the *A-wins* or *B-wins* terminal state. Finally, it is noteworthy that some of the 19602 states that have the mouths of snakes or the tails of ladders in them do not have an incoming edge since it is impossible to land on them. However, an outgoing edge with probability 1 is added from that state to the state to which the corresponding snake or ladder leads.

The 19602×19602 transient matrix P_{tran} for this Markov chain is constructed. Luckily, this matrix is extremely sparse because each row has at most six nonzero entries with value $1/6$ (and might have five, if a transition lands on the finishing state from that state). The 19602-dimensional vector \vec{p}_A contains a value of 0 or $1/6$. The value of $1/6$ occurs at the i th entry if a transition from the i th state in \vec{p}_A could reach the finishing state *A-wins* (i.e., player A is to move next and their position on the board is within a distance of 6 from square 100). The vector \vec{p}_B is defined similarly. Then, the 19602-dimensional probability vectors of A and B winning are $(I - P_{tran})^{-1}\vec{p}_A$ and $(I - P_{tran})^{-1}\vec{p}_B$, respectively. Note that these vectors will always add to a vector of 1s because either A or B must win in steady-state. The specific probability of A winning can be obtained as the entry in the 19602-dimensional vector $(I - P_{tran})^{-1}\vec{p}_A$, which belongs to the state $(38, 34, B)$. This vector contains the probability of A winning from every possible of the 19602 positions on the board. ■

10.2.8 The View from Matrix Algebra (*)

This section requires significant background in linear algebra and can be omitted without loss in continuity by uninitiated readers. The following discussion will briefly go over the interpretations of eigenvectors of transition matrices and how they can be used for efficient computation in Markov chains. Readers are referred to [6] for an introduction to concepts such as Jordan normal form and *generalized* eigenvectors, which will be used below assuming that the reader has prior knowledge of these concepts.

The previous sections use several algebraic polynomials of the matrix P to compute different properties of the Markov chain. At first glance, it might seem onerous to compute quantities like P^k , P^∞ , and $(I - \alpha P)^{-1}$, if the underlying transition matrices are large.

However, all of these quantities can be computed very rapidly using the Jordan normal form of the $n \times n$ transition matrix P :

$$P = V\Delta V^{-1}$$

All matrices in the above equation are $n \times n$ matrices. The matrix V contains the generalized right eigenvectors of P in its columns, and V^{-1} contains the generalized left eigenvectors of P in its rows. The matrix Δ is an upper triangular $n \times n$ matrix, but it can be considered “almost diagonal,” because it typically contains far fewer non-zero off-diagonal entries than the number of diagonal elements. Multiplying such a matrix with itself, computing its k th power, or inverting it is a computationally simple matter. With this in mind, the following tricks can be used to compute various functions of P :

$$\begin{aligned} P^k &= V\Delta^k V^{-1} \\ P^\infty &= V\Delta^\infty V^{-1} \\ (I - \alpha P)^{-1} &= V(I - \alpha\Delta)^{-1}V^{-1} \end{aligned}$$

It is much easier to perform these operations on upper-triangular matrices, particularly “almost diagonal” matrices. The computation of Δ^∞ is enabled by some special properties of P . All real or complex diagonal entries of Δ can be shown to be at most 1 in *magnitude* using the Perron-Frobenius theorem [6], and there is at least one eigenvalue with value 1. If any diagonal entry is a negative or a complex root of unity, then P^∞ does not exist because of *periodicity* in the Markov chain. Therefore, periodic Markov chains are recognized by transition matrices whose eigenvalues include negative or complex roots of unity.

Observation 10.1 *The process of mixing in the Markov chain requires the entries of Δ^k to stabilize with increasing k . The presence of negative or complex roots of unity on the diagonal of Δ prevents this stabilization.*

For aperiodic matrices (in which P^∞ exists), diagonal entries are either exactly 1 in *value* or they are less than 1 in *magnitude* (even if they are complex or negative eigenvalues). The number of diagonal entries with a value of exactly 1 is equal to the number of absorbing components⁵ in the Markov chain. Furthermore, it can be shown that there are no off-diagonal entries for repeated eigenvalues of 1 on the diagonal of Δ because each such eigenvalue has a distinct eigenvector with an interpretation relating to a particular absorbing component in the Markov chain; the distinct left eigenvector for a particular absorbing component is the steady-state probability vector when starting the Markov process in that component. The distinct right eigenvector of an absorbing component contains the n probabilities⁶ of ending up in that component when starting in each state. For example, ergodic Markov chains always have a right eigenvector of 1s. The interpretation is that there is only one component in that Markov chain, and one will therefore always end up in that component when starting in any state.

The aforementioned properties can be used to construct P^∞ as follows. If the i th column of Δ contains a single 1 on the diagonal entry and is therefore equal to the unit column vector \vec{e}_i containing 0s and a single 1, then the i th column of Δ^∞ can be set to \vec{e}_i as well. If the i th column contains an eigenvalue of magnitude less than 1, then the i th column of Δ^∞

⁵For an ergodic Markov chain, the number of absorbing components is assumed to be 1.

⁶These interpretations apply to a carefully chosen basis of the eigenspace with eigenvalue 1 but not to an arbitrary basis of the eigenspace. For example, the average/difference of two eigenvectors with the same eigenvalue is also an eigenvector but it may not be interpretable.

can be set to the zero vector. The resulting matrix $V\Delta^\infty V^{-1}$ yields P^∞ . One can make this computation even simpler by recognizing that Δ^∞ only contains k ones along its diagonal. In particular, if V_k is the $n \times k$ submatrix of V containing all the k right eigenvectors of P with eigenvalue 1 in its columns and $[V^{-1}]_k$ is the $k \times n$ submatrix of V^{-1} containing all the left eigenvectors with eigenvalue 1 in its rows, then P^∞ is defined as follows:

$$P^\infty = V\Delta^\infty V^{-1} = V_k[V^{-1}]_k$$

The matrix P^∞ will be of rank- k (where k is the number of absorbing components), and the value of k will be 1 for ergodic Markov chains. As a result, P^∞ can be constructed very efficiently using the Jordan normal form. Furthermore, P^∞ can be stored efficiently in space $O(n \cdot k)$ by separately storing V_k and $[V^{-1}]_k$, which contain probabilistic properties of the k absorbing components. We omit detailed proofs of the above results and refer the reader to [6].

Example 10.18 Consider the Markov chain in Figure 10.1(b). Without constructing the transition matrix or doing any linear algebra, answer the following based on the properties discussed in this section relating eigenspaces to the graphical topology of the Markov chain structure: (a) How many absorbing components does this Markov chain have? (b) Discuss why the transition matrix has two eigenvectors with eigenvalue 1. (c) Provide a probabilistic interpretation of the two left eigenvectors $[1, 0, 0, 0]$ and $[0, 0, 1, 0]$ with eigenvalue 1 in terms of steady-state probabilities when starting in particular absorbing components. (d) Provide a probabilistic interpretation of the two right eigenvectors $[1, 0.4787, 0, 0.5958]^T$ and $[0, 0.5213, 1, 0.4043]^T$ with eigenvalue 1 in terms of probabilities of ending up in particular absorbing components when starting in various states.

Solution: (a) The Markov chain has two absorbing components corresponding to states s_1 and s_3 .

(b) The matrix has two eigenvectors with eigenvalue 1 because there are two absorbing components in the matrix.

(c) The left eigenvector $[1, 0, 0, 0]$ corresponds to the steady-state probability when starting at any node in this absorbing component (which happens to be s_1 in this case). A similar argument applies to the left eigenvector $[0, 0, 1, 0]$ corresponding to the absorbing state containing only s_3 .

(d) The right eigenvector $[1, 0.4787, 0, 0.5958]^T$ contains the probability of ending up in the first absorbing component (containing only s_1) when starting off in the four states. For example, the value of 0.4787 is the probability of ending up in the first absorbing component when starting in state s_2 . A similar interpretation applies to the other right eigenvector in the context of the other absorbing component (containing only state s_3). The reader is also encouraged to compare these right eigenvectors with the columns of P^∞ worked out in Example 10.8. ■

Problem 10.6 The eigenvalues of the periodic Markov chain in Figure 10.4(a) are $1, -1, \sqrt{-1}, -\sqrt{-1}$ and the matrix Δ of the Jordan normal form is diagonal. Compute Δ^2 , Δ^3 , Δ^4 , and Δ^5 . How does Δ^5 relate to Δ ? Show that Δ^k oscillates between four possible matrices with increasing k . Use this fact to discuss why P^k oscillates between four matrices with increasing k . You should not have to need to explicitly compute the matrix V in the Jordan normal form.

10.3 Machine Learning Applications of Markov Chains

In this section, we will discuss a number of machine learning applications of graphs. Perhaps the most celebrated application of Markov chains occurs in the *PageRank* problem, which is used by search engines to rank documents on the Web. This application uses ergodic Markov chains. However, some applications use non-ergodic Markov chains as well. A specific application of non-ergodic Markov chains occurs in the case of node classification in graphs. Both these applications will be discussed in this section.

10.3.1 PageRank

The PageRank problem arises in the context of Web search, and it is used to find reputable Web pages. In keyword-based search, one is often looking not only for relevant Web pages but also for reputable Web pages. Reputable Web pages are generally more useful to the person performing the search, and it therefore makes sense to rank them higher in the search results. One property of reputable Web pages is that they are often pointed to by other Web pages. Furthermore, the Web pages pointing to them are themselves likely to be reputable. This measure of reputation is referred to as the *PageRank*.

The *PageRank* problem is naturally modeled using a Markov chain. The basic idea is to assume that a random surfer clicks on pages (among the links on the Web page currently being surfed) at random in order to traverse the Web. Therefore, each Web page can be treated as a state in the Markov chain, and each traversal of a page can be treated as a probability transition. Then, the *PageRank* is treated as the steady-state probability of the state corresponding to a Web page in this transition process. In other words, The *PageRank* represents the fraction of time that a random surfer would spend on each Web page. It is clear that a random surfer would spend more time on reputable Web pages, because they are typically pointed to by more pages, and a substantial number of pages pointing to them are also reputable.

Let π_i be the steady-state probability of vertex i under the random walk model. Then, the probability π_i of visiting a particular vertex i is given by the sum of the probabilities of transitioning into that vertex from each of its incoming vertices. The probability of transitioning into vertex i from vertex j is given by $\pi_j p_{ji}$. We can write this relationship mathematically for each vertex $i \in \{1, \dots, n\}$ as follows:

$$\pi_i = \sum_{j=1}^n \pi_j p_{ji} \quad \forall i \in \{1, \dots, n\}$$

Note that we have n equations in n variables π_1, \dots, π_n . We can denote the vector of n variables by $\vec{\pi} = [\pi_1, \pi_2, \dots, \pi_n]$. The above system of equations can then be written in vector form as follows:

$$\vec{\pi} = \vec{\pi}P$$

This is exactly a left-eigenvector equation at an eigenvalue of 1. A unique eigenvector with eigenvalue 1 exists for the stochastic transition matrices associated with strongly connected graphs. This is one of the consequences of an important theorem in spectral graph theory, referred to as the *Perron-Frobenius theorem*. The discussion of this theorem is beyond the scope of this book and interested readers are referred to [6].

The main problem is that the Web graph may not be strongly connected, which also causes challenges for the ergodicity of the Markov chain. Some Web pages may have no outgoing links, which may result in the random surfer getting trapped at specific nodes. In

fact, a probabilistic transition is not even meaningfully defined at such a node. Such nodes are referred to as *dead ends*. An example of a dead-end node is illustrated in Figure 10.8(a). Clearly, dead ends are undesirable because the transition process for *PageRank* computation cannot be defined at that node. To address this issue, two modifications are incorporated in the random surfer model. The first modification is to add an edge from each dead-end node to all other nodes. Each such edge has transition probability $1/n$. This does not fully solve the problem, because the dead ends can also be defined on *groups of nodes*. In these cases, there are no outgoing links from a *group of nodes* to the remaining nodes in the graph. This is referred to as a *dead-end component*, or *absorbing component*. An example of a dead-end component is illustrated in Figure 10.8(b).

Dead-end components are common in the Web graph because the Web is not strongly connected. In such cases, the transitions at individual nodes can be meaningfully defined, but the steady-state transitions will stay trapped in these dead-end components. All the steady-state probabilities will be concentrated in dead-end components because there can be no transition out of a dead-end component after a transition occurs into it. Therefore, as long as even a minuscule probability of transition into a dead-end component exists, *all* the steady-state probability becomes concentrated in such components. This situation is not desirable from the perspective of *PageRank* computation in a large Web graph, where dead-end components are not necessarily an indicator of popularity. Furthermore, in such cases, the final probability distribution of nodes in various dead-end components is not unique and it is dependent on the starting state. This is easy to verify by observing that random walks starting in different dead-end components will have their respective steady-state distributions concentrated within the corresponding components.

While the addition of edges (exiting dead-end nodes) solves the problem for dead-end nodes, an additional step is required to address the more complex issue of dead-end components. Therefore, aside from the addition of these edges, a *teleportation*, or *restart step* is used within the random surfer model. This step is defined as follows. At each transition, the random surfer may either jump to an arbitrary page with probability α , or it may follow one of the links on the page with probability $(1 - \alpha)$. A typical value of α used is 0.1. Because of the use of teleportation, the steady state probability becomes unique and independent of the starting state. The value of α may also be viewed as a *smoothing* or *damping probability*. Large values of α typically result in the steady-state probability of different pages to become more even. For example, if the value of α is chosen to be 1, then all pages will have the same steady-state probability of visits. Note that both teleportation and the addition of self-loops can be addressed by making modifications to the transition matrix P .

Let P_o be the transition matrix obtained by adding self-loops to the Web graph. The matrix P_o assumes that it is equally likely to exit any node. The final transition matrix does not include the effect of teleportation. The transition matrix P is obtained from this transition matrix P_o (corresponding to the Web graph with self-loops) as follows:

$$P \Leftarrow (1 - \alpha)P_o + \alpha M/n$$

Here, M is an $n \times n$ matrix containing 1s. The matrix M/n is a transition matrix in which one can move from any vertex to another with probability $1/n$. Therefore, this is a strongly-connected restart matrix. The final transition matrix P is a weighted combination of the original transition matrix and the restart matrix. This transition matrix is used for the *PageRank* computation.

One can use the power method [6] to solve for $\vec{\pi}$ since it is the *principal* eigenvector of the stochastic transition matrix. The approach works by initializing $\vec{\pi}$ to a vector of random positive values between 0 and 1, and then scaling all values to sum to 1. Subsequently, the

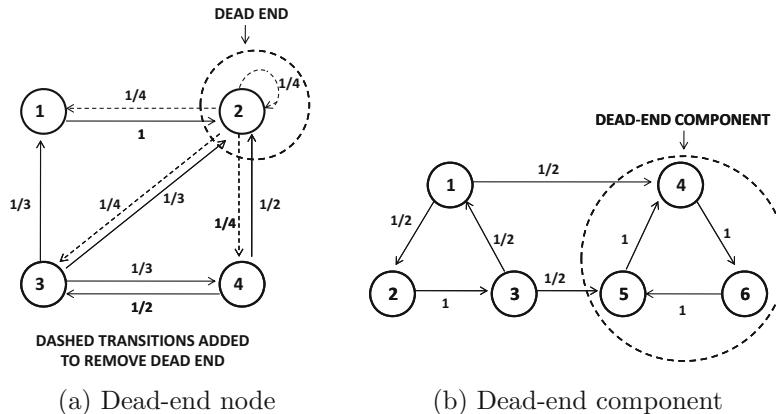


Figure 10.8: Transition probabilities for *PageRank* computation with dead end nodes

following iterative process is repeated:

1. $\vec{\pi} \Leftarrow \vec{\pi}P$
 2. Normalize $\vec{\pi}$ so that its elements sum to 1.

This approach is applied to convergence.

beyond its use on the Web, *PageRank* is useful for finding prominent actors in directed social networks like Twitter. Users that have a lot of followers are generally more important than those who do not have many followers. In addition, users with reputable followers are likely to be reputable themselves. This is precisely the type of application-centric scenario in which the *PageRank* algorithm is very effective.

Example 10.19 (Efficient Computation) One problem with the use of restart is that it makes the transition matrix P dense even when the underlying graph is sparse. Discuss how you can treat the restart component of P more carefully, so as to be able to compute $\bar{\pi}$ using only sparse matrix operations.

Solution: The final transition matrix can be represented in terms of the original transition matrix P_o as $(1 - \alpha)P_o + \alpha\vec{1}_n\vec{1}_n^T/n$. Here, $\vec{1}_n$ is an n -dimensional column vector of 1s.

The matrix P_o is sparse. Instead of applying the $\vec{\pi} = \vec{\pi}P$ iteration, one can implement it as follows:

$$\vec{\pi} \leftarrow (1 - \alpha)\vec{\pi}P_o + \alpha[\vec{\pi} \vec{1}_n] \vec{1}_n^T/n$$

Note the order of operations in the last term. The quantity $[\vec{\pi} \vec{1}_n]$ is a scalar, which is computed first. As a result, expensive matrices do not need to be maintained as intermediates. ■

Example 10.20 Discuss how the modification of the transition probabilities to incorporate restart can be considered a special case of Laplacian smoothing.

Solution: One can view the transition probability from a Web graph as an estimated value from a single instance of the Web graph. Smoothing incorporates a prior in

which the probability of transitioning from each Web page to each adjacent page is the same. The restart probability is similar to the Laplacian smoothing parameter λ . When the restart probability is 1, every node has the same steady-state probability. ■

10.3.1.1 Application to Undirected Networks

Although the Web is a directed network, many real-world networks like social and bibliographic networks are undirected. For example, Facebook is different from Twitter in these sense that there is no direction to the friendship links in the former (unlike follower-follower links). Even in the case of undirected networks, actors with many reputable links are reputable themselves. In such cases, it is possible to find reputable actors by converting the undirected networks to directed ones. Specifically, an undirected network is converted to a directed one by replacing an undirected edge with two directed ones. The resulting network is already strongly connected (and therefore no modifications are required to handle dead-end nodes). However, it might still have the problem of periodicity (see Exercise 3). This does not turn out to be a problem, when the *PageRank* algorithm is applied with teleportation. This is because teleportation implicitly adds edges between each pair of nodes in the graph (which naturally breaks any underlying periodic structure).

10.3.1.2 Personalized PageRank

The teleportation procedure provides a way to personalize the *PageRank* to a particular node (or set of nodes) in the network. The approach can be applied to either directed graphs or undirected graphs. In the event that the procedure is applied to directed graphs, dead-end nodes need to be addressed by adding edges from each node to all other nodes in the network (with equal transition probabilities). In the even that the graph is undirected, each undirected edge needs to be replaced with two directed edges.

Consider a social network where it is desirable to find all reputable actors that are similar to particular actor of interest X. Therefore, the *PageRank* computation needs to be personalized to actor X, and there is a trade-off between finding actors that are reputable and those that are similar to X. A modification to the teleportation procedure of the general *PageRank* method provides a good way to achieve a proper trade-off between achieving good similarity and reputation. Specifically, instead of teleporting to any of the n nodes in the network with equal probability α/n , one only transitions to the special node X with probability α . Making these modification increases the *PageRank* of nodes that are close to X. The choice of $\alpha \in (0, 1)$ regulates the trade-off between reputation and personalization. A small value of α reduces the relative importance of personalization, whereas a large value of α increases the relative importance of personalization. Therefore, personalized *PageRank* is a form of reputation-centric similarity computation. Personalized *PageRank* is very similar to the Katz measure.

10.3.2 Application to Vertex Classification

Reducible matrices can be used for applications to vertex classification in graphs. The problem of vertex classification is also referred to as *collective classification*. The problem of vertex classification is suited to undirected graphs with a symmetric adjacency matrix A . The adjacency matrix of a graph is a matrix A in which the entries a_{ij} contain the

nonnegative weights of the edges in the matrix. For example, in a social network, a_{ij} might represent the number of messages exchanged between actors i and j . Note that the adjacency matrix can be converted into a transition matrix by normalizing each row to sum to 1. In other words, one can set p_{ij} as follows:

$$p_{ij} = \frac{a_{ij}}{\sum_{j=1}^n a_{ij}}$$

In collective classification, a subset of the vertices are labeled with one of k labels, denoted by $\{1, \dots, k\}$. Some of the vertices may not be labeled. The goal is to classify the unlabeled vertices based on the structure of the graph and the known labels. This problem often arises in social networks, where one attempts to find actors with particular properties (e.g., interest in a particular product), based on other (known) actors with these properties. The known actors can, therefore, be labeled with their properties. An example of an undirected graph with a subset of vertices that are labeled either ‘A’ or ‘B’ is shown in Figure 10.9(a). Therefore, this particular example corresponds to the binary label setting with $k = 2$. For ease in discussion, assume that the edge weights of the adjacency matrix A are binary. Many vertices are not labeled in Figure 10.9(a), and the goal is to classify precisely these vertices.

The basic principle for solving this problem is to use the principle of *homophily* in social networks. The idea is that vertices tend to be connected with vertices that have similar properties. Therefore, a random walk starting at an unlabeled vertex is more likely to first reach a labeled vertex, whose label value matches its own. Therefore, a probabilistic approach for solving this problem is as follows:

Given an unlabeled vertex, perform a random walk by using the stochastic transition matrix of the adjacency matrix until a labeled vertex is reached. Output the observed label of the destination (labeled) vertex as the predicted class label of the source (unlabeled) vertex from which the random walk begins.

For better robustness, one can compute the *probability* that a vertex of each class is reached. *This setting is very similar to that of computing outcome probabilities in a Markov chain according to the approach discussed in section 10.2.7.* The intuition for this approach is that the walk is more likely to terminate at labeled vertices in the proximity of the starting vertex i . Therefore, when many vertices of a particular class are located in its proximity, then the vertex i is more likely to be labeled with that class. In the particular case of Figure 10.9(a), any random walk starting from test vertex X will always reach label ‘A’ first rather than label ‘B’ because of the topology of the graph. However, it is also possible to select test vertices, where there is a non-zero probability of reaching either the label ‘A’ or the label ‘B’. For example, if one starts the random walk at test vertex Y, then a vertex corresponding to either label ‘A’ or label ‘B’ could be reached first.

An important assumption is that the graph needs to be *label connected*. In other words, every unlabeled vertex needs to be able to reach a labeled vertex in the random walk. For undirected graphs, this means that every connected component of the graph needs to contain at least one labeled vertex. In the following discussion, it will be assumed that the entire undirected graph is connected; any undirected graph with only one connected component will always lead to a modified transition matrix that is label connected.

Since the approach is based on random walks, the first step is to create the (directed) stochastic transition matrix P from the undirected $n \times n$ adjacency matrix A . As discussed earlier in this section, the adjacency matrix is converted into the stochastic transition matrix

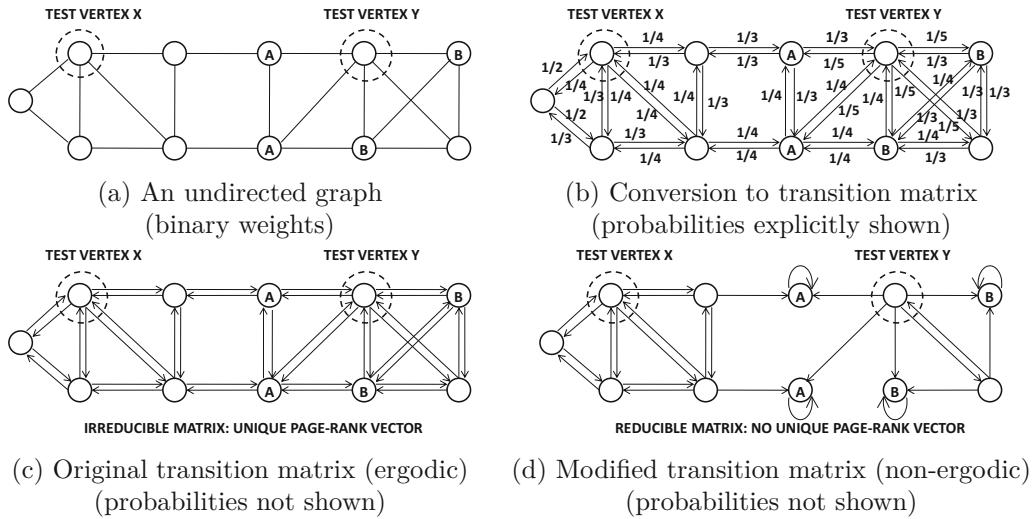


Figure 10.9: Creating directed transition graphs from undirected graph

by using the following normalization:

$$p_{ij} = \frac{a_{ij}}{\sum_{j=1}^n a_{ij}} \quad (10.5)$$

Since this transition matrix is derived from a connected and undirected graph, it will always be strongly connected (as each undirected edge is replaced with two edges in opposite directions). The corresponding strongly connected graph is illustrated in Figure 10.9(b), and the probabilities on the various edges are explicitly shown. The same illustration is shown in Figure 10.9(c), but without the probabilities on the edges (to avoid clutter).

Although we can use this transition matrix to model the random walks in the graph, such an approach will not provide the *first stopping point* of the walk. In other words, we need to model the random walks in such a way that they always terminate at their *first arrival* at labeled vertices. This can be achieved by removing outgoing edges from labeled vertices and replacing them with self-loops. This results in a singleton absorbing component containing only one vertex, which is essentially an absorbing vertex. Such vertices are referred to as absorbing vertices because they trap the random walk after an incoming transition. The goal of creating such absorbing vertices is to ensure that a random walk is trapped by the first labeled node it reaches.

The stochastic transition matrix P needs to be modified to account for the effect of absorbing vertices. For each absorbing vertex i , the i th row of P is replaced with the i th row of the identity matrix. Henceforth, we will assume that the matrix denoted by notation P incorporates this modification (and is therefore not exactly equal to $\Delta^{-1}A$ according to Equation 10.5). An example of the final transition graph is illustrated in Figure 10.9(d). The resulting matrix is no longer strongly connected; no other vertex can be reached from an absorbing vertex. Note that this graph has exactly the structure of Figure 10.6 because each of the absorbing components is a singleton (labeled) vertex, and all unlabeled vertices are transient. The corresponding transition matrix does not have a unique eigenvector with eigenvalue 1. Rather, it has as many principal eigenvectors with eigenvalue 1 as the number of absorbing vertices.

For any given starting vertex i , the steady-state probability distribution has positive values only at labeled vertices. This is because a random walk will eventually reach an absorbing vertex in a label-connected graph, and it will never emerge from that vertex. Therefore, if one can estimate the steady-state probability distribution of labeled nodes for a starting unlabeled vertex i , then the probability values of the labeled vertices in each class can be aggregated. The class with the highest probability is reported as the relevant label of the unlabeled vertex i .

Note that the (i, j) th entry of P^r yields the probability of a random walk of length r starting at any vertex i to terminate at any vertex j . Because of the self-loops at absorbing vertices, all walks of length less than r are automatically included in the probability (as the remaining steps can be completed inside the self-loop). Therefore, P^∞ is the steady-state matrix of probabilities, in which the (i, j) th entry of P^∞ provides the probability that a walk starting a vertex i terminates at vertex j . For each row, we would like to aggregate the probabilities of the classes belonging to the different labels. As we will see below, this can be achieved with a simple matrix multiplication.

let Y be an $n \times k$ matrix in which the (i, c) th entry is 1, if the i th vertex is labeled and it belongs to the class $c \in \{1, \dots, k\}$. Then, the aggregation of the probabilities of labeled vertices in each row of P^∞ is given by the matrix $P^\infty Y$. In other words, we can obtain an $n \times k$ matrix Z of probabilities of the classes for the various vertices as follows:

$$Z = P^\infty Y \quad (10.6)$$

The class with the maximum probability in Z for unlabeled vertex (row) i may be reported as its class label. This approach is also referred to as the *rendezvous approach* to label propagation [8]. How is P^∞ computed? One possibility is keep multiplying P with itself in order to compute P^∞ . However, doing so can be extremely expensive. One can use eigendecomposition tricks to speed up the process. However, we do not want to work with large matrices (like P^∞) of size $n \times n$. Therefore, a more efficient (but equivalent) approach is *iterative label propagation* [65].

In iterative label propagation, we initialize $Z^{(0)} = Y$ and then repeatedly use the following update for increasing value of iteration index t :

$$Z^{(t+1)} = PZ^{(t)} \quad (10.7)$$

It is easy to see that $Z^{(\infty)}$ is the same as the value of Z in Equation 10.6. Furthermore, each column of Z is a principal right eigenvector of P at convergence.

Example 10.21 Suppose you have a directed follower network with only four actors $s_1 \dots s_4$, and edges weighted according to the strength of the followee brand influence. Two of the nodes are labeled as Coca Cola for s_1 and Pepsi for s_3 as stated brand loyalties. The other two actors have no known brand loyalty but are presumed to have been influenced into liking one of them by their followees. You convert labeled states s_1 and s_3 into absorbing states using the approach discussed in this section, resulting in the Markov chain of Figure 10.1(b). Calculate the probabilities that the strength of follower influence will cause s_2 and s_4 to like Coca Cola and Pepsi, respectively. You may use P^∞ from Example 10.8.

Solution: We first create the 4×2 matrix Y of brand loyalty labels, in which the first column is *Coca Cola* and the second column is *Pepsi*:

$$Y = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

As discussed in this section, the probabilities for the labels of all nodes can be calculated as $Z = P^\infty Y$:

$$Z = P^\infty Y \approx \begin{bmatrix} 1.000 & 0.000 & 0.000 & 0.000 \\ 0.479 & 0.000 & 0.521 & 0.000 \\ 0.000 & 0.000 & 1.000 & 0.000 \\ 0.596 & 0.000 & 0.404 & 0.000 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 \\ 0.479 & 0.521 \\ 0.000 & 1.000 \\ 0.596 & 0.404 \end{bmatrix}$$

Based on the above calculation, actor s_2 will prefer *Coca Cola* with probability 0.479, whereas actor s_4 will prefer *Coca Cola* with probability 0.496. The probabilities for *Pepsi* are the complements of these values. ■

10.4 Markov Chains to Generative Models

A natural question arises as to how one can convert the Markov chains discussed in previous sections into generative models. It is easier to understand this by drawing an analogy with the generative models discussed in previous chapters. In the generative models discussed in previous chapters, each data item is generated by an independent trial. In other words, the generative methodologies of the previous chapters use the following broad approach:

1. Select a mixture component via a random process that is independent of any random processes that have occurred thus far.
2. Generate a data item from a distribution that depends on the choice of this mixture component.

Markov chains can be used to create generative models in which the generated data items are not independent of one another. The main difference is that the mixture components are now selected as the states of a Markov chain. Then, a data item is generated that is specific to the state of that probability distribution. Therefore, the generative process associated with a Markov chain is as follows:

1. Select a mixture component that is defined by the current state in the Markov chain.
2. Generate a data item from a distribution that depends on the choice of this mixture component.

Since the selection of the mixture components is defined by the states in the Markov chain, it is evident that the data items that are generated are also not independent of one another.

In most cases, the data items that are generated from each mixture component (state) in a Markov model are symbols generated from a categorical distribution. As a result, the outputs of the generative models defined by Markov chains are generally sequences.

Since successive symbols of the sequence are generated from successive states of the Markov chain, it is evident that the successively generated symbols are stochastically related to one another. This behavior reflects real-world settings in which the adjacent symbols in real-world sequences (e.g., amino acid sequences) are closely related to one another. As a result, such generative models are used in a variety of applications involving sequences, including bioinformatics, Web traversal patterns, and user buying behavior.

Although it is possible to generalize Markov models to generate continuous time series, this will not be the focus of this chapter. This chapter will focus on a specific type of Markov model that only generates discrete sequences. It is assumed that one has a database of observed sequences, which are generated by the transitions of a Markov model behind the scenes. Since the states of such a model are not directly observed, they are referred to as *Hidden Markov Models*. These models will be discussed in the next section.

10.5 Hidden Markov Models

Hidden Markov models (HMM) are probabilistic models that generate sequences through a sequence of transitions between states in a Markov chain. Since hidden Markov models are analogs of mixture models in multidimensional data, it is not difficult to see that hidden Markov models can enable the same types of applications as mixture models do in multi-dimensional data. Hidden Markov models are used for clustering, classification, and outlier detection. Therefore, the applicability of these models is very broad in sequence analysis. In section 10.6, other applications of hidden Markov models will be used to facilitate further understanding.

A hidden Markov model is built on top of a Markov chain by generating symbols on each transition of the Markov chain. In a hidden Markov model, the states of the system are *hidden* and not directly visible to the user. Only a sequence of (typically) discrete observations is visible to the user that is generated by symbol emissions from the states after each transition. This is not different from a traditional mixture model such as that used in EM-clustering, where the identities of the mixture components generating the points are not visible to the user. The generated sequence of symbols corresponds to the application-specific sequence data. In many cases, the states may be defined (during the modeling process) on the basis of an *understanding* of how the underlying system behaves, though the precise sequence of transitions may not be known to the analyst. This is why such models are referred to as “*hidden*.” The quality of the modeling, therefore, depends on the level of insight used by the end-user in creating an appropriate model.

Each state in an HMM is associated with a *set of emission probabilities* over the symbol Σ . In other words, a visit to the state s_j leads to an emission of one of the symbols $\sigma_i \in \Sigma$ with probability $\theta^j(\sigma_i)$. Correspondingly, a sequence of transitions in an HMM corresponds to an *observed data sequence*. Hidden Markov models are, therefore, mixture models in which different components of the mixture are not generated using independent trials, but are related through sequential transitions. Thus, each state is analogous to a component in the multidimensional mixture model. Each symbol generated by this model is analogous to a data point generated by the multidimensional mixture model. Furthermore, unlike multidimensional mixture models, the successive generation of individual data items (sequence symbols) are also not independent of one another. This is a natural consequence of the fact that the successive states emitting the data items are dependent on one another with the use of probabilistic transitions. Unlike multidimensional mixture models, hidden Markov models are designed for sequential data that exhibits temporal correlations.

The symbol set Σ often uses special (dummy) START and END symbols, when it is desired to model sequences of limited length. often the case in many real-world applications. Correspondingly, the Markov model contains START and END (dummy) states, which emit this symbol deterministically. No other state emits these symbols. The transition probabilities from the START state to each state regulate the initial state probabilities of the non-dummy states, which (in turn) control the probabilities of the first non-dummy symbols in the sequence. It is also possible for the symbol set to include the null symbol. Including the null symbol in Σ allows the cases where no symbol is generated in a particular state.

To better explain hidden Markov models, an illustrative example will be used for the specific problem of using HMMs for anomaly detection. Consider the scenario where a set of students register for a course and generate a sequence corresponding to the grades received in each of their weekly assignments. This grade is drawn from the symbol set $\Sigma = \{A, B\}$. *The model created by the analyst is that* the class contains students who, at any given time, are either *doers* or *slackers* with different grade-generation probabilities. A student in a *doer* state may sometimes transition to a *slacker* state and vice versa. These represent the two states in the system. Weekly home assignments are handed out to each student and are graded with one of the symbols from Σ . This results in a *sequence* of grades for each student, and it represents the only *observable* output for the analyst. The state of a student represents only a *model* created by the analyst to *explain* the grade sequences and is, therefore, not observable in of itself. It is important to understand that if this model is a poor reflection of the true generative process, then it will impact the quality of the learning process.

Assume that a student in a *doer* state is likely to receive an *A* grade in a weekly assignment with 80% probability and a *B* with 20% probability. For *slackers*, these probability values are reversed. Although these probabilities are explicitly specified here for illustrative purposes, they need to be *learned* or *estimated* from the observed grade sequences for the different students and are not known *a priori*. The precise status (state) of any student in a given week is not known to the analyst at any given time. These grade sequences are, in fact, the only *observable* outputs for the analyst. Therefore, from the perspective of the analyst, this is a *Hidden Markov Model*, which generates the sequences of grades from an *unknown* sequence of states, representing the state transitions of the students. The precise sequence of transitions between the states can be only *estimated* for a particular observed sequence.

The two-state hidden Markov model for the aforementioned example is illustrated in Figure 10.10. This model contains two states, denoted by *doer* and *slacker*, that represent the state of a student in a particular week. It is possible for a student to transition from one state to another each week, though the likelihood of this is rather low. It is assumed that set of initial state probabilities governs the *a priori* distribution of *doers* and *slackers*. This distribution represents the *a priori* understanding about the students when they join the course. Some examples of *typical sequences* generated from this model, along with their rarity level, are illustrated in Figure 10.10. For example, the sequence **AAABAAAABAAAA** is most likely generated by a student who is consistently in a *doer* state, and the sequence **BBBBBABBBA** is most likely generated by a student who is consistently in *slacker* state. The second sequence is typically rarer than the first because the population mostly contains *doers*. The sequence **AAABAAABBABBB** corresponds to a *doer* who eventually transitions into a *slacker*. This case is even rarer because it requires a transition from the *doer* state to a *slacker* state, which has a very low probability. The sequence **ABABABABABABA** is *extremely anomalous* because it does not represent temporally consistent *doer* or *slacker* behavior that

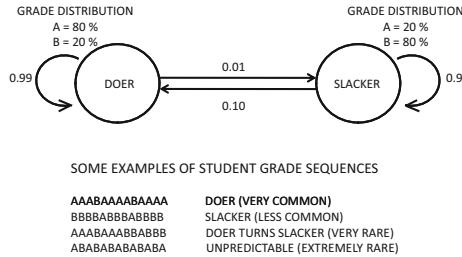


Figure 10.10: Generating grade sequences from a hidden Markov model

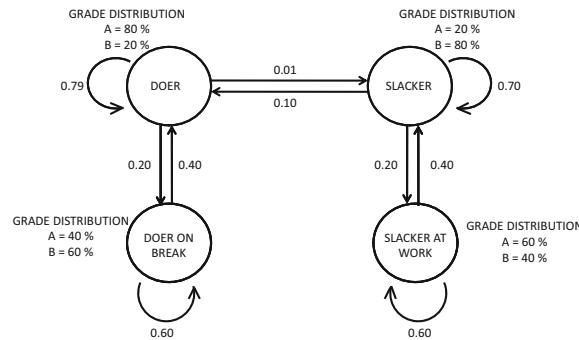


Figure 10.11: Increasing the expressive power of the model in Figure 10.10

is implied by the model. Correspondingly, such a sequence has very low probability of fitting the model. One can already see that if appropriate methods are available for estimating the likelihood fits of sequences, they can be converted into outlier scores.

A larger number of states in the Markov Model can be used to encode more complex scenarios. It is possible to encode domain knowledge with the use of states that describe different generating scenarios. In the example discussed earlier, consider the case that *doers* sometimes slacks off for short periods and then return to their usual state. Alternatively, *slackers* may sometimes become temporarily inspired to be *doers*, but may eventually return to what they are best at. Such episodes will result in local portions of the sequence that are distinctive from the remaining sequence. These scenarios can be captured with the 4-state Markov model illustrated in Figure 10.11. The larger the number of states are, the more complex the scenarios that can be captured. Of course, more training data is required to learn the (larger number of) parameters of such a model, or this may result in overfitting. For smaller data sets, the transition probabilities and symbol-generation probabilities are not estimated accurately.

Example 10.22 Consider the hidden Markov mode of Figure 10.10. What fraction of the grades generated by the model in steady-state will be A? What fraction of the A grades generated by the steady-state model will be generated by the slacker state?

Solution: The first step is to compute the steady-state probabilities of the underlying 2-state Markov chain $\{s, d\}$ (corresponding to slacker and doer states) for which the

transition matrix is as follows:

$$\begin{bmatrix} & s & d \\ s & 0.90 & 0.10 \\ d & 0.01 & 0.99 \end{bmatrix}$$

Using a Web calculator, it can be shown that the principal left eigenvector of this matrix is [1/11, 10/11], which also provides the steady-state probability for the two states. Using the total probability rule, the fraction of A grades generated is $(1/11) * 0.2 + (10/11) * 0.8 = 8.2/11$. This computation evaluates to 0.745.

One can use the Bayes rule to compute the probability $P(s|A)$ of an A grade being a slacker as follows:

$$P(s|A) = \frac{P(A|s)P(s)}{P(A)} = \frac{0.2 * (1/11)}{8.2/11} = \frac{1}{41}$$

Therefore, only 1/41 of the A grades will be generated by slackers. ■

Problem 10.7 If John goes jogging on a particular morning, the probability that he goes jogging the next morning is 0.9. On the other hand, if John does not go jogging on a particular morning, the probability that he goes jogging the next morning is 0.6. If John goes jogging in the morning, his probability of taking a shower is 0.9. Otherwise, his probability of taking a shower in the morning is 0.5. Construct a hidden Markov model that generates a sequence of 0-1 bits, indicating whether or not John takes a shower each morning. In the long term, what percentage of mornings does John take a shower?

Problem 10.8 Consider the crude weather model introduced in Problem 10.1. Tom goes to work every day. He takes his umbrella with him 100% of the time on rainy days, 75% of the time on cloudy days, and 50% of the time on sunny days. Construct a hidden Markov model that generates a sequence of bits indicating whether or not Tom takes his umbrella with him each day. In the long term, what percentage of the time does Tom carry his umbrella?

10.5.1 Formal Definition and Techniques for HMMs

In this section, hidden Markov models will be formally introduced along with the associated training methods. It is assumed that a hidden Markov model contains n states denoted by $\mathcal{S} = \{s_1 \dots s_n\}$. Simply speaking, \mathcal{S} represents the states from the underlying Markov chain. The symbol set from which the observations are generated is denoted by $\Sigma = \{\sigma_1 \dots \sigma_{|\Sigma|}\}$. The symbols are generated from the model by a sequence of transitions from one state to the other. Each visit to a state (including self-transitions) generates a symbol drawn from a categorical probability distribution on Σ . The symbol emission distribution is specific to each state. The probability $P(\sigma_i|s_j)$ that the symbol σ_i is generated from state s_j is denoted by $\theta^j(\sigma_i)$. The probability of a transition from state s_i to s_j is denoted by p_{ij} (which is essentially a transition probability of the underlying Markov chain). The initial state probabilities are denoted by $\pi_1 \dots \pi_n$ for the n different states. The topology of the model can be expressed as a network $G = (\mathcal{S}, A)$, in which \mathcal{S} is the set of states $\{s_1 \dots s_n\}$. The set A represents the possible transitions between the states for which the transition probabilities p_{ij} are positive. In the most common scenario, where the architecture of the model is constructed with a domain-specific understanding, the set A is not the complete

network of edges between all pair of states. In cases where domain-specific knowledge is not available, the set A may correspond to the complete network, including self-transitions. *The goal of training the HMM model is to learn the initial state probabilities, transition probabilities, and the symbol emission probabilities from the training database $\{T_1 \dots T_N\}$.* Each T_i is a sequence of symbols generated by the hidden Markov model, and there are a total of N sequences. Three methodologies are commonly leveraged in creating and using a hidden Markov model:

- *Training:* Given a set of training sequences $T_1 \dots T_N$, estimate the model parameters, such as the initial probabilities, transition probabilities, and symbol emission probabilities with an Expectation-Maximization algorithm. The Baum-Welch algorithm is used for this purpose.
- *Evaluation:* Given a test sequence V (or comparison unit U_i), determine the probability that it fits the HMM. This is used to determine the anomaly scores. A recursive forward algorithm is used to compute this.
- *Explanation:* Given a test sequence V , determine the most likely sequence of states that generated this test sequence. This is helpful for providing an understanding of why a sequence should be considered an anomaly (in outlier detection) or belong to a specific class (in data classification). The idea is that the states correspond to an intuitive understanding of the underlying system. In the example of Figure 10.10, it would be useful to know that an observed sequence is an anomaly because of the unusual oscillation of a student between *doer* and *slacker* states. This can provide the *intensional knowledge* for understanding the state of a system. This most likely sequence of states is computed with the Viterbi algorithm.

Since the description of the training procedure relies on technical ideas developed for the evaluation method, we will deviate from the natural order of presentation and present the training algorithms last. The evaluation and explanation techniques will assume that the model parameters, such as the transition probabilities, are already available from the training phase.

10.5.2 Evaluation: Computing the Fit Probability for Observed Sequence

One approach for determining the fit probability of a sequence $V = a_1 \dots a_m$ would be to compute all the n^m possible sequences of states (paths) in the HMM, and compute the probability of each, based on the observed sequence, symbol-generation probabilities, and transition probabilities. The sum of these values can be reported as the fit probability. Obviously, such an approach is not practical because it requires the enumeration of an exponential number of possibilities.

This computation can be greatly reduced by recognizing that the fit probability of the first r symbols (and a fixed value of the r th state) can be recursively computed in terms of the corresponding fit probability of first $(r - 1)$ observable symbols (and a fixed $(r - 1)$ th state). Specifically, let $\alpha_r(V, s_j)$ be the probability that the first r symbols in V are generated by the model, and the last state in the sequence is s_j . Then, the recursive computation is as follows:

$$\alpha_r(V, s_j) = \sum_{i=1}^n \alpha_{r-1}(V, s_i) \cdot p_{ij} \cdot \theta^j(a_r)$$

This approach recursively sums up the probabilities for all the n different paths for different penultimate nodes. The aforementioned relationship is iteratively applied for $r = 1 \dots m$. The probability of the first symbol is computed as $\alpha_1(V, s_j) = \pi_j \cdot \theta^j(a_1)$ for initializing the recursion. This approach requires $O(n^2 \cdot m)$ time. Then, the overall probability is computed by summing up the values of $\alpha_m(V, s_j)$ over all possible states s_j . Therefore, the final fit $F(V)$ is computed as follows:

$$F(V) = \sum_{j=1}^n \alpha_m(V, s_j)$$

This algorithm is also known as the *Forward Algorithm*. Note that the fit probability has a direct application to many problems, such as classification and anomaly detection, depending upon whether the HMM is constructed in supervised or unsupervised fashion. By constructing separate HMMs for each class, it is possible to test the better-fitting class for a test sequence. The fit probability is useful in problems such as data clustering, classification and outlier detection. In data clustering and classification, the fit probability can be used to model the probability of a sequence belonging to a cluster or class, by creating a group-specific HMM. In outlier detection, it is possible to determine poorly fitting sequences with respect to a global HMM and report them as anomalies.

10.5.3 Explanation: Determining the Most Likely State Sequence for Observed Sequence

One of the goals in many data mining problems is to provide an explanation for why a sequence fits part (e.g. class or cluster) of the data, or does not fit the whole data set (e.g. outlier). Since the sequence of (hidden) generating states often provides an intuitive explanation for the observed sequence, it is sometimes desirable to determine the *most likely sequence of states* for the observed sequence. The Viterbi algorithm provides an efficient way to determine the most likely state sequence.

One approach for determining the most likely state path of the test sequence $V = a_1 \dots a_m$ would be to compute all the n^m possible sequences of states (paths) in the HMM, and compute the probability of each of them, based on the observed sequence, symbol-generation probabilities, and transition probabilities. The maximum of these values can be reported as the most likely path. Note that this is a similar problem to the fit probability except that it is needed to determine the *maximum* fit probability, rather than the *sum* of fit probabilities, over all possible paths. Correspondingly, it is also possible to use a similar recursive approach as the previous case to determine the most likely state sequence.

Any sub-path of an optimal state path must also be optimal for generating the corresponding subsequence of symbols. This property, in the context of an optimization problem of sequence selection, normally enables dynamic programming methods. The best possible state path for generating the first r symbols (with the r th state fixed to j) can be recursively computed in terms of the corresponding best paths for the first $(r - 1)$ observable symbols and different penultimate states. Specifically, let $\delta_r(V, s_j)$ be the probability of the best state sequence for generating the first r symbols in V and also ending at state s_j . Then, the recursive computation is as follows:

$$\delta_r(V, s_j) = \text{MAX}_{i=1}^n \delta_{r-1}(V, s_i) \cdot p_{ij} \cdot \theta^j(a_r)$$

This approach recursively computes the maximum of the probabilities of all the n different paths for different penultimate nodes. The approach is iteratively applied for $r = 1 \dots m$.

The first probability is determined as $\delta_1(V, s_j) = \pi_j \cdot \theta^j(a_1)$ for initializing the recursion. This approach requires $O(n^2 \cdot m)$ time. Then, the final best path is computed by using the maximum value of $\delta_m(V, s_j)$ over all possible states s_j . This approach is, essentially, a dynamic programming algorithm. In the anomaly example of student grades, an oscillation between *doer* and *slacker* states will be discovered by the Viterbi algorithm as the causality for outlier behavior. In a clustering application, a consistent presence in the *doer* state will explain the cluster of diligent students.

10.5.4 Training: Baum-Welch Algorithm

The problem of learning the parameters of an HMM is a very difficult one, and no known algorithm is guaranteed to determine the global optimum. However, options are available to determine a reasonably effective solution in most scenarios. The Baum-Welch algorithm is one such method. It is also known as the *Forward-backward* algorithm, and it is an application of the EM approach to the generative hidden Markov model. First, a description of training with the use of a single sequence $T = a_1 \dots a_m$ will be provided. Then, a straightforward generalization to N sequences $T_1 \dots T_N$ will be discussed.

Let $\alpha_r(T, s_j)$ be the *forward* probability that the first r symbols in a sequence T of length m are generated by the model, and the last symbol in the sequence is s_j . Let $\beta_r(T, s_j)$ be the *backward* probability that the portion of the sequence after *and not including the rth position* is generated by the model, *conditional on the fact that* the state for the r th position is s_j . Thus, the forward and backward probability definitions are not symmetric. The forward and backward probabilities can be computed from model probabilities in a way similar to the evaluation procedure discussed above in section 10.5.2. The major difference for the backward probabilities is that the computations start from the end of the sequence in the backward direction. Furthermore, the probability value $\beta_{|T|}(T, s_j)$ is initialized to 1 at the bottom of the recursion to account for the difference in the two definitions. Two additional probabilistic quantities need to be defined to describe the EM algorithm:

- $\psi_r(T, s_i, s_j)$: Probability that the r th position in sequence T corresponds to state s_i , the $(r+1)$ th position corresponds to s_j .
- $\gamma_r(T, s_i)$: Probability that the r th position in sequence T corresponds to state s_i .

The EM procedure starts with a random initialization of the model parameters and then iteratively estimates $(\alpha(\cdot), \beta(\cdot), \psi(\cdot), \gamma(\cdot))$ from the model parameters, and vice versa. Specifically, the iteratively executed steps of the EM procedure are as follows:

- (E-step) Estimate $(\alpha(\cdot), \beta(\cdot), \psi(\cdot), \gamma(\cdot))$ from currently estimated values of the model parameters $(\pi(\cdot), \theta(\cdot), p_{..})$.
- (M-step) Estimate model parameters $(\pi(\cdot), \theta(\cdot), p_{..})$ from currently estimated values of $(\alpha(\cdot), \beta(\cdot), \psi(\cdot), \gamma(\cdot))$.

It now remains to explain how each of the above estimations is performed. The values of $\alpha(\cdot)$ and $\beta(\cdot)$ can be estimated using the forward and backward procedures, respectively. The forward procedure is already described in the evaluation section, and the backward procedure is analogous to the forward procedure, except that it works backward from the end of the sequence. The value of $\psi_r(T, s_i, s_j)$ is equal to $\alpha_r(T, s_i) \cdot p_{ij} \cdot \theta^j(a_{r+1}) \cdot \beta_{r+1}(T, s_j)$ because the sequence-generation procedure can be divided into three portions corresponding

to that up to position r , the generation of the $(r+1)$ th symbol, and the portion after the $(r+1)$ th symbol. The estimated values of $\psi_r(T, s_i, s_j)$ are normalized to a probability vector by ensuring that the sum over different pairs $[i, j]$ is 1. The value of $\gamma_r(T, s_i)$ is estimated by summing up the values of $\psi_r(T, s_i, s_j)$ over fixed i and varying j . This completes the description of the E-step.

The re-estimation formulas for the model parameters in the M-Step are relatively straightforward. Let $I(a_r, \sigma_k)$ be a binary indicator function, which takes on the value of 1 when the two symbols are the same, and 0 otherwise. Then the estimations can be performed as follows:

$$\begin{aligned}\pi(j) &= \gamma_1(T, s_j), \quad p_{ij} = \frac{\sum_{r=1}^{m-1} \psi_r(T, s_i, s_j)}{\sum_{r=1}^{m-1} \gamma_r(T, s_i)} \\ \theta^i(\sigma_k) &= \frac{\sum_{r=1}^m I(a_r, \sigma_k) \cdot \gamma_r(T, s_i)}{\sum_{r=1}^m \gamma_r(T, s_i)}\end{aligned}$$

The precise derivations of these estimations, on the basis of expectation-maximization principles, may be found in [43]. This completes the description of the M-step.

As in all EM methods, the procedure is applied iteratively to convergence. The approach can be generalized easily to N sequences by applying the steps to each of the sequences, and averaging the corresponding model parameters in each step. These steps are repeated to convergence.

Example 10.23 (Generative Language Models) *Discuss how you can create a hidden Markov model that generates a sentence in a particular language using each word as a state. Next discuss how you can generalize this model so that the Markov model effectively uses the window of the last k words to generate the next word. Assume that you have a large database of sentences available.*

Solution: One can represent each word as a state and the transition probability to another word is the conditional probability that the next word occurs after the current word. Note that the next word could also be an END state corresponding to the fact that the sentence ends at that state. These conditional probabilities are estimated from data containing sentences by using MLE of categorical transition probabilities (cf. section 6.3.2 of Chapter 6). Each transition emits a symbol corresponding to the word at the destination state during the transition. The START state moves to any of the word states with a probability that is equal to the probability that a sentence starts with that word.

In the generalized model, each possible combination of k words is a state. Note that many of these combinations may not be used in the data and such states can be pruned. Transitions occur in a manner consistent with a window of k consecutive words in a sentence. For example, for a trigram model, a transition may occur from the state “the fox jumped” to “fox jumped over.” The word emitted in this case would be “over.” A transition could also occur to the END state. The conditional probabilities can be estimated in a manner similar to the first case. The START state would have transitions to each possible trigram with a probability estimated from the language database. ■

10.6 Applications of Hidden Markov Models

Hidden Markov models can be used for a wide variety of sequence mining problems, such as clustering, classification, and anomaly detection. The most common application domain for hidden Markov models is computational biology, because many biological compounds like proteins and amino acids can either be directly represented as sequences or flattened into sequences. This section will introduce several of these applications.

10.6.1 Mixture of HMMs for Clustering

This approach can be considered the string analog of the probabilistic models discussed in section 9.2 of Chapter 9 for clustering multidimensional data. Recall that a generative mixture model is used in that case, where each component of the mixture has a distribution defined by the data type at hand. A Gaussian distribution is used for clustering numerical data, whereas a Bernoulli distribution is used for generating binary data. A categorical distribution can be used for generating categorical data, whereas a multinomial distribution can be used for generating sparse numeric data. Therefore, the type of mixture distribution used depends on the type of data set at hand.

Consider a sequence database containing N sequences denoted by $S_1 \dots S_N$. A good generative model for generating each S_i is the hidden Markov model. It is noteworthy that the HMM can itself be considered a kind of mixture model, in which states represent dependent components of the mixture. Therefore, this approach can be considered a *two-level* mixture model. The discussion in this section should be combined with the description of HMMs in section 10.5 to provide a complete picture of HMM-based clustering.

The broad principle of a mixture-based generative model is to assume that the data was generated from a mixture of k distributions with the probability distributions $\mathcal{G}_1 \dots \mathcal{G}_k$, where each \mathcal{G}_i is a hidden Markov model. As in section 9.2 of Chapter 9, the approach assumes the use of prior probabilities $\alpha_1 \dots \alpha_k$ for the different components of the mixture. Therefore, the generative process is described as follows:

1. Select one of the k probability distributions with probability α_i where $i \in \{1 \dots k\}$.
Let us assume that the r th one is selected.
2. Generate a sequence from \mathcal{G}_r , where \mathcal{G}_r is a mixture component. The sequence generated by this mixture component uses a hidden Markov model.

This generative process is repeated N times in order to generate the N observed sequences $S_1 \dots S_N$. From the perspective of the analyst, these sequences are observed outputs of the process, and the parameters of the HMM (as well as prior probabilities) need to be estimated from the observed data. The expectation-maximization approach is therefore used to achieve this goal.

One nice characteristic of the expectation-maximization algorithm is that the change in the data type and corresponding mixture distribution does not affect the broader framework of the algorithm. The analogous steps can be applied in the case of sequence data, as they are applied in multidimensional data. Let S_j represent the j th sequence and Θ be the entire set of parameters to be estimated for the different HMMs. Then, the E-step and M-step are exactly analogous to the case of the multidimensional mixture model.

1. (E-step) Given the current state of the trained HMM, it is possible to compute each likelihood value $P(S_j | \mathcal{G}_i)$ using the procedure discussed in section 10.5.2 on the i th HMM. Given the current values of priors α_i , determine the posterior probability

$P(\mathcal{G}_i|S_j)$ of each sequence S_j using the likelihood values $P(S_j|\mathcal{G}_i)$ in conjunction with the Bayes rule:

$$P(\mathcal{G}_i|S_j) = \frac{\alpha_i P(S_j|\mathcal{G}_i)}{\sum_{r=1}^k \alpha_r P(S_j|\mathcal{G}_r)} \quad \forall i, j \quad (10.8)$$

This is the posterior probability that the sequence S_j was generated by the i th HMM.

2. (M-step) Given the current probabilities of assignments of data points to clusters, use the Baum-Welch algorithm on each HMM to learn its parameters. The assignment probabilities are used as weights for averaging the estimated parameters. The Baum-Welch algorithm is described in section 10.5.4 of this chapter. The main difference in this case is that a weighted variant of this procedure needs to be used, where the weight of the j th sequence is given by its posterior probability (defined by Equation 10.8). The value of each α_i is estimated to be proportional to the average assignment probability of all sequences to mixture component \mathcal{G}_i . Thus, the M-step results in the estimation of the entire set of parameters Θ .

The E-step is executed once at the end of the procedure in order to determine the final assignment probabilities. It is noteworthy that the basic framework used here is the same as the mixture modeling approach for multidimensional data (cf. section 9.2 of Chapter 9). This is not particularly surprising because mixture modeling methods can be generalized to any type of data as long as an appropriate parameter estimation procedure exists for each mixture component. The differences between using different data types with mixture models simply boils down to different estimation procedures within each mixture component because of the difference in data type (and corresponding generating distribution). The major drawback of this approach is that it can be rather slow.

10.6.2 Outlier Detection

Consider a database of sequences denoted by $S_1 \dots S_N$. One can simply apply the Baum-Welch algorithm to the determine the parameters of the Markov model for generating the sequence. In theory, it is possible to compute anomaly scores directly for the test sequence V , once the training model has been constructed from the sequence database $S_1 \dots S_N$. This is achieved by applying the procedure discussed in section 10.5.2 in order to find the likelihood fit, and then reporting those sequences that have high outlier scores as outliers. One can even use this approach to score out-of-sample sequences, although in-sample scores cannot be reasonably compared to out-of-sample scores (because of over-fitting in the former).

One problem with this broad approach is that as the length of the test sequence increases, the robustness of such a model diminishes because of the increasing noise associated with estimating the probability of a long sequence. Therefore, smaller windows of the sequences are extracted, and the hidden Markov model is constructed on top of these smaller windows. These smaller windows are referred to as *comparison units*. Therefore, the comparison units from the training data are used for constructing the model (with the Baum-Welch algorithm), and the comparison units from the test sequences are used for computing the anomaly scores of windows of the sequence (with the likelihood fit procedure discussed in section 10.5.2). The anomaly scores of the different windows can then be combined together by using a simple function such as determining the number of anomalous window units in a sequence. Sequences with a large percentage of anomalous windows can be designated as outliers.

Finally, it is also possible to use a mixture of hidden Markov models (cf. section 10.6.1) in order to find anomalous sequences. It has been discussed in section 9.4.2 of Chapter 9,

how mixture models for clustering can be generalized to outlier detection. The basic idea is to report items with low likelihood fit to the mixture distribution as outliers. Therefore, exactly the same process discussed in section 10.6.1 can be used. The only difference is that a final postprocessing phase is applied in which sequences with low likelihood fit are reported as outliers.

10.6.3 Classification

Consider a sequence (training) database $S_1 \dots S_N$, and each sequence S_i is tagged with label y_i , which is a categorical index drawn from $\{1, \dots, k\}$. The basic approach is to create a separate HMM for each of the classes in the data. Therefore, if there are a total of k classes, this will result in k different Hidden Markov Models. The Baum-Welch algorithm, described in section 10.5.4, is used to train the HMMs for each class. For a given test sequence, the fit of each of the k models to the test sequence is determined using the approach described in section 10.5.2. The best matching class is reported as the relevant one. The overall approach for training and testing with HMMs may be described as follows:

1. (Training) Use Baum-Welch algorithm of section 10.5.4 to construct a separate HMM model for each of the k classes by applying the training algorithm to the appropriate class-specific subset from $S_1 \dots S_N$. The mixture component for the i th class is denoted by \mathcal{G}_i .
2. (Testing) For a given test sequence Y , determine the fit probability the sequence to the k different Hidden Markov Models, using the evaluation procedure discussed in section 10.5.2. Let the fit probability of sequence Y to the i th class be denoted by $P(Y|\mathcal{G}_i)$. Let the relative frequency (prior probability) of class i in the training data be α_i . Then, by using Bayes rule, the posterior probability of the i th class is given by the following:

$$P(\mathcal{G}_i|Y) = \frac{\alpha_i P(Y|\mathcal{G}_i)}{\sum_{r=1}^k \alpha_r P(Y|\mathcal{G}_r)} \quad \forall i$$

The class with the highest posterior probability is reported as the relevant one. It is possible to ignore the denominator in the aforementioned equation when comparing between classes because all classes have the same denominator. In other words, we have:

$$P(\mathcal{G}_i|Y) \propto \alpha_i P(Y|\mathcal{G}_i)$$

The class with the highest posterior probability is reported as the relevant one.

It is noteworthy that the generative model for sequence classification is similar to that used in clustering (cf. section 10.6.1). The main difference is that the parameter-estimation process is greatly simplified by the fact that the sequences are labeled. As a result, one no longer has to perform an iterative expectation-maximization procedure, but the application of a single maximization procedure suffices. This is because the assignment of training sequences to mixture components is known (because of the presence of labels), and one can use the appropriate portion of the sequence database in order to learn the parameters of the relevant mixture component (through the use of the Baum-Welch algorithm).

10.7 Summary

This chapter introduces the discrete-state Markov process and the related Markov chain model. The Markov chain is useful for modeling several real-world applications on the

Web and social networks, including *PageRank* and collective classification. Discrete state Markov processes also form the basis of generative models that can be used to model discrete sequences. This model is referred to as a hidden Markov model. A hidden Markov model is essentially a mixture model in which the repeated generation of mixture components is not achieved through independent trials but by successive states of a Markov chain (which are obviously not independent). This is the reason that Markov chains are ideal for generating sequence data. Hidden Markov models are used in a variety of applications in computational biology, where they are used for traditional machine learning applications like clustering, classification, and outlier detection.

10.8 Further Reading

A general discussion of the linear algebra of graphs is provided in [6]. Numerous applications of Markov chains and Markov models are given in [1, 5, 6]. Markov chains are introduced in a simple way in [54]. The Perron-Frobenius theorem, which is discussed in [6], is important for understanding the uniqueness of the dominant eigenvector of a transition matrix, corresponding to an eigenvalue of 1. The *PageRank* algorithm was introduced in [15], although better descriptions are available in [44]. A study of random-walk methods for collective classification may be found in [11]. A tutorial on hidden Markov models may be found in [43]. The Viterbi algorithm is explained in tutorial style in [24].

10.9 Exercises

1. Calculate the steady-state probabilities of the states in Figure 10.1(a).
2. You have four subjects to study — mathematics, physics, chemistry, and biology. You can study only one subject on a given day. On any given day, you study one of the two subjects that you did *not* study over the last two days. You decide which of these subjects to study with the flip of a fair coin. Model this process with a Markov chain.
3. The *PageRank* section discusses how it is sometimes useful to construct a directed Markov chain from an undirected graph by replacing each undirected edge with two directed edges in opposite directions. Give an example of an undirected graph in which this type of conversion results in a Markov chain in which periodicity exists (and therefore the Markov chain is not ergodic).
4. Suppose that a Markov chain has a transition matrix P that is such that $P^r = I$ for some r and identity matrix I . Note that the Markov chains of cycles of length r show this property. Show that at least one element of P^k must be zero, irrespective of the value of integer k . Can such a Markov chain be ergodic?
5. George is an alcoholic whose tendency to drink is affected by his drinking behavior in the last two days. If he has been sober over the last two days, his probability of getting drunk today is 0.1. Otherwise, his probability of getting drunk today is either 0.5 or 1, depending on whether he was drunk in one of the last two days or on both (respectively). Model George's drinking behavior with a Markov chain. Is this Markov chain ergodic? Show that in steady-state, George will always be drunk.
6. Repeat Exercise 5 with the difference that the probability of George getting drunk today is 0.9 instead of 1.0 in the case of his having been drunk over both of the past

two days. All other aspects of the problem definition remain the same. Calculate the steady-state probability of George being drunk.

7. Suppose that you are given all the transition probabilities and symbol emission probabilities together with the graph topology of a hidden Markov model. This model is designed to generate an infinitely long sequence. How can you use this information to compute the fraction of the time that each symbol is generated in an infinitely long sequence?
8. Consider a possibly non-ergodic Markov chain with states $s_1 \dots s_n$. There is no guarantee as to whether the chain is aperiodic or strongly connected (and therefore there is no guarantee of the existence of steady-state probabilities). The starting state is sampled uniformly at random from the Markov chain. After exactly r transitions from the sampled starting state, the current state of the chain is sampled and determined to be the m th state s_m . Given the known identity of the final state s_m , outline a procedure to determine the conditional probability that the starting state was s_i for each i . [Note the similarity of this exercise to Example 10.15.]
9. Consider the case where you sample the state of the Markov chain in Figure 10.1(a) after a large number of transitions, and find it to be state s_3 . The initial state was sampled uniformly at random among the states. What is the conditional probability that the starting state was s_2 ? [Hint: Too much calculation is not needed.]
10. Consider the case where you sample the state of the Markov chain in Figure 10.1(b) after a large number of transitions. The initial state of the process was sampled so that the probability of state s_i was inversely proportional to i . What is the probability that your sampled state is s_3 ?
11. Consider the case where you sample the state of the Markov chain in Figure 10.1(b) after a large number of transitions, and find it to be state s_3 . The initial state of the process was sampled so that the probability of state s_i was inversely proportional to i . What is the conditional probability that the starting state was s_2 ?
12. Suppose that you create a hidden Markov model that models the generation of a database of gene sequences. A novel mutation exists in one of the gene sequences. How can you detect this novel mutation using the hidden Markov model?
13. Jim can be cheerful, neutral, or upset on a particular day. If he is cheerful on a particular day, his probabilities of being cheerful, neutral, or upset on the next day are 0.7, 0.2, and 0.1, respectively. If Jim is neutral on a particular day, he is equally likely to be cheerful, neutral, or upset on the next day. If Jim is upset on a particular day, his probabilities of being cheerful, neutral, or upset on the next day are 0.2, 0.3, and 0.5, respectively. What is the steady-state probability of Jim's emotional state?
14. Consider the setting of Exercise 13. Jim's emotional state is not directly observable but one can observe what beverage he drinks each morning. If he is cheerful, he has juice with probability 0.6, milk with probability 0.3, and coffee with probability 0.1. If he is neutral, he has juice with probability 0.3, milk with probability 0.5, and coffee with probability 0.2. If he is upset, he has juice with probability 0.2, milk with probability 0.3, and coffee with probability 0.5. Construct a hidden Markov model that generates the sequence of beverages consumed by Jim.

15. Consider the problem in Exercise 14. What fraction of the days does Jim drink milk? If you see him drinking milk, what is the probability that he is cheerful?
16. Treat the hidden Markov model of Figure 10.11 as a Markov chain and compute its steady-state probabilities. You may use any Web calculator for matrix computation.
17. What fraction of the grades produced by the steady-state hidden Markov model in Figure 10.11 are A grades? What fraction are produced by the slacker state?
18. Days are sunny, cloudy, or rainy. If it is sunny today, it will be sunny tomorrow with probability 0.7, cloudy with probability 0.2, and rainy with probability 0.1. If it is cloudy today, it will be sunny tomorrow with probability 0.2, cloudy with probability 0.3, and rainy with probability 0.5. If it is rainy today, it will be sunny tomorrow with probability 0.4, cloudy with probability 0.2, and rainy with probability 0.4. Use a Markov chain to find the fraction of days that are sunny, cloudy, or rainy.
19. You have a fair coin and a biased coin with heads probability of 0.8. You select one of the two coins and keep flipping it until it turns up tails. A tails is a cue to switch to the other coin. This process is repeated a very large number of times. What fraction of all coin flips are heads?
20. After performing the process of Exercise 19 for a long time, a flip of the coin is found to be heads. What is the probability that the flip came from the biased coin?
21. Each of urns A and B contains two balls — a white ball and a black ball. In each step, a ball is selected at random from each urn and the two balls are exchanged between the urns. Model the process using a Markov chain. What is the steady-state probability that each urn contains a white ball and a black ball?
22. A gambler starts with 2 dollars in her pocket. Her probability of winning a game of cards is 0.4 and that of losing is 0.6. If she wins, she gains one dollar or else she loses one dollar. She quits when she is either out of money or has doubled her initial amount of 2 dollars to 4 dollars. What is the probability that she quits broke?
23. For the problem in Exercise 22, what is the expected number of games the gambler will play before they quit (either by doubling or by going broke)?
24. Consider the case where a state-specific biased coin is flipped after each transition of the Markov chain in Figure 10.1(a). The heads probability of the biased coin is $(i/5)$ when the transition occurs *into* state s_i . After a large number of transitions, the last two flips show HH. What is the probability that the last state visited was s_2 ?
25. A game of tennis is won by the first player to score four points as long as they have a two-point lead. Otherwise, they play until one player gets a two-point lead. Jim wins each point with probability 0.6 against Neal. Describe the states, topology, and transition probabilities of a Markov chain that you can use to compute the probability of Jim winning the game. Use MATLAB or Python to compute this probability.
26. For the problem in Exercise 25, use your program to find the expected number of points that a game will last. Find the probability that the score is 3-2 in favor of Jim at some point.



Chapter 11

Probabilistic Inequalities and Approximations

“Dogs laugh, but they laugh with their tails.”— Max Eastman

11.1 Introduction

Numerous probabilistic inequalities are used to bound probabilities of different events. These inequalities generally apply to either special forms of a random variable, or to special regions of a random variable, such as its extreme regions. Some inequalities, such as *Jensen’s inequality*, are dependent on the special properties (e.g., convexity property) of a function that is applied to a random variable. Various types of tail inequalities also exist that are applicable to sums of independent random variables drawn from particular ranges. This chapter will study both types of inequalities.

Tail inequalities can be used in order to bound the probability that the value of a random variable occurs in the tail of a probability distribution. In cases where it is possible to make assumptions such as approximating an aggregation-centric random variable with the normal distribution, the probability of the tail can be quantified by using normal distribution or t -distribution tables. Although such assumptions lead to very accurate average-case estimations of the tail probabilities, they are not worst-case bounds. In some applications, one wishes to have worst-case bounds of the probability of the distribution tail. This is especially true in the field of *randomized algorithms* [50], where one needs to quantify limits on the performance behavior of the algorithms. The creation of bounds on the probability of the tail requires the development of the machinery of *tail inequalities*.

Different tail inequalities can be developed with different types of conditions on the underlying random variables. Restrictive conditions on the random variables allow these bounds to be very *tight*, which means that the bound is very close to the true value. Tight bounds are sometimes referred to as strong bounds, whereas loose bounds are sometimes

referred to as weak bounds. For example, the *Markov* and *Chebychev* inequalities are weak inequalities but they apply to very large classes of random variables. On the other hand, the Chernoff bound and Hoeffding inequality are both stronger inequalities but they apply to restricted classes of random variables. In general, *the more assumptions that one is willing to make on the random variables, the tighter the bound on the underlying variable will be*. The most extreme case is when one assumes that a random variable is drawn from a specific distribution like the normal distribution. If the variable truly belongs to the normal distribution, one can obtain the *exact* probability mass in the tail. Unfortunately, since no real-world distribution is truly normal, the results for the normal distribution has *assumption-centric error* and cannot be viewed as a guarantee. Therefore, such assumptions cannot be used in applications where theoretical results on worst-case guarantees are desirable.

Although the Chernoff bound works on distributions that are more general than the binomial distribution, its most common use is to bound variables that are more general than the binomial distribution. This chapter also discusses a number of approximations of the binomial distribution. In particular, the normal approximation and the Poisson approximation are studied, which are effective for different ranges of parameters. Since these approximations are assumptions, they do not provide worst-case guarantees. At the same time, they provide closer estimations to the sample value of the variable on average in most practical settings.

11.1.1 Chapter Organization

This chapter is organized as follows. The next section introduces Jensen's inequality, which is applicable to a convex function of a random variable. The Markov and Chebychev inequalities are introduced in section 11.3. These are weak tail inequalities but they apply to general forms of random variables. Section 11.4 introduces the Chernoff bound and the Hoeffding inequality. The comparisons of the tail inequalities to the bounds provided by the central limit theorem is given in section 11.5. A summary is given in section 11.6.

11.2 Jensen's Inequality

Jensen's inequality is applicable to convex functions of random variables. A function $F(\cdot)$ is said to be convex if for any set of k points x_1, x_2, \dots, x_k in its domain, *the function of the weighted average of these points is at most equal to the weighted average of the function of the various points*, where the weights are fixed to $\lambda_1, \dots, \lambda_k$ for the k points. Since these weights are relative weights, it is assumed that they sum to 1:

$$\sum_{i=1}^k \lambda_i = 1$$

One can formally state the definition of a convex function as follows:

Definition 11.1 (Convex Function) *Let x_1, x_2, \dots, x_k be k data points with weights $\lambda_1, \lambda_2, \dots, \lambda_k$ summing to 1. The function $F(\cdot)$ is convex if the following is true for any set of points and corresponding weights:*

$$F\left(\sum_{i=1}^k \lambda_i x_i\right) \leq \sum_{i=1}^k \lambda_i F(x_i)$$

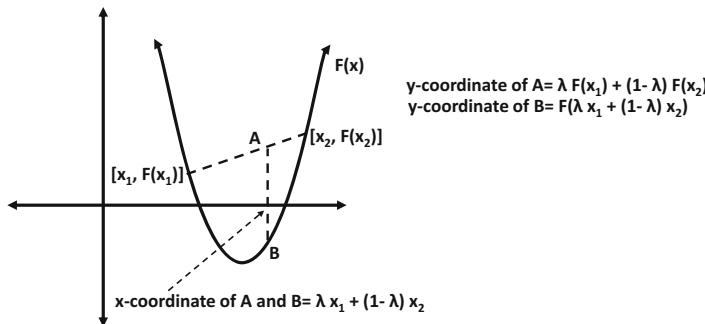


Figure 11.1: The convexity condition is equivalent to saying that a point on the secant of a convex function always lies above the value of the function at the same x-coordinate.

Note that reversing the direction of the inequality in the above equation yields a *concave function*. A convex function is a bowl-shaped function, in which the mouth of the bowl faces upwards. Such a function is also referred to as “concave up.” On the other hand, a concave function is one in which the mouth of the bowl faces downwards. Such a function is also referred to as “concave down.” For example, the function $F(x) = (x - 2)^2 + 3$ is convex because it is concave up for all x . On the other hand, the function $F(x) = -(x - 2)^2 + 10$ is concave because it is concave down for all x . The function $F(x) = x^3$ is neither concave nor convex. This is because the function is concave up for $x > 0$ and concave down for $x < 0$.

The above definition of convex functions can be best understood by considering the case where a weighted average of two points is used. Let x_1 and x_2 be two points, so that their weighted average is $\lambda x_1 + (1 - \lambda)x_2$ for $\lambda \in (0, 1)$. For any convex function $F(\cdot)$ (which is concave up), the secant joining the points x_1 and x_2 always lies above the function. Furthermore, it can be shown using the linear nature of the secant line that for each $\lambda \in (0, 1)$, the value $\lambda F(x_1) + (1 - \lambda)F(x_2)$ lies on the secant exactly at the x-coordinate value of $\lambda x_1 + (1 - \lambda)x_2$. However, since the function value at this point is less than the secant value, it results in the following inequality:

$$F(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda F(x_1) + (1 - \lambda)F(x_2) \quad (11.1)$$

In order to understand the nature of this inequality for convex functions, we illustrate a convex function in Figure 11.1. Two points x_1 and x_2 have been selected and the secant between them is constructed. Note that the secant always lies above the convex function at each point because a convex function is an upright bowl (i.e., concave up). Two points A and B are shown in the figure, so that both points have the same x-coordinate value of $\lambda x_1 + (1 - \lambda)x_2$. However, point A lies on the secant, whereas point B lies on $F(X)$. It can be easily shown that the y-coordinate of A on the secant is $\lambda F(x_1) + (1 - \lambda)F(x_2)$, whereas the y-coordinate of B on the function is $F(\lambda x_1 + (1 - \lambda)x_2)$. Since the point A always lies above B , it is easy to see that Equation 11.1 is always satisfied. The above inequality can easily be extended by induction to k values of $\lambda_1, \dots, \lambda_k$ satisfying $\sum_i \lambda_i = 1$ — this extension is what is stated in Definition 11.1. For twice differentiable functions, a simpler way of checking convexity is by confirming that its double derivative is nonnegative:

$$F''(x) \geq 0 \quad \forall x$$

The aforementioned characterization can be intuitively understood from the fact that convex functions are concave up, so that the derivative keeps increasing with increasing x .

Although Jensen's inequality applies to both discrete and continuous random variables, it is easiest to understand it with the use of discrete random variables. Let X be a discrete random variable with k possible values x_1, x_2, \dots, x_k with corresponding probabilities p_1, p_2, \dots, p_k . Let $F(\cdot)$ be a convex function. By definition, the expectation of the random variable $F(X)$ is as follows:

$$E[F(X)] = \sum_{i=1}^k p_i F(x_i)$$

On the other hand, when the function $F(\cdot)$ is applied to the expected value $E[X] = \sum_{i=1}^k p_i x_i$, we obtain the following:

$$F(E[X]) = F\left(\sum_{i=1}^k p_i x_i\right)$$

Note that the probabilities $p_1 \dots p_k$ can be viewed as weights summing to 1, just like $\lambda_1, \lambda_2, \dots, \lambda_k$. Since the function $F(\cdot)$ is convex, the following inequality automatically arises:

$$F(E[X]) \leq E[F(X)]$$

In other words, the function of the expectation is at most equal to expectation of the function over all values of the random variable. This result is intuitively quite natural because the expectation of a random variable is a weighted average of all possible values of the random variable — the convexity definition is also based on weighted averages. Jensen's inequality can be generalized easily to continuous random variables by observing that the expectation of a continuous random variable is also defined in terms of weighted averages, except that it is done using an integral (i.e., an infinite Riemann summation of the different weighted components). Jensen's inequality may be formally stated as follows:

Theorem 11.1 (Jensen's Inequality for Convex Functions) *Let X be a random variable and $F(\cdot)$ be a convex function. Then, the function of the expectation is at most equal to the expectation of the function over all values of the random variable X :*

$$F(E[X]) \leq E[F(X)]$$

One can easily design a variant of Jensen's inequality by reversing the direction of the inequality:

Corollary 11.1 (Jensen's Inequality for Concave Functions) *Let X be a random variable and $F(\cdot)$ be a concave function. Then, the function of the expectation is at least equal to the expectation of the function over all values of the random variable X :*

$$F(E[X]) \geq E[F(X)]$$

This corollary is relatively easy to show by using the fact that the negative of a concave function is a convex function. Therefore, one can use Jensen's inequality for convex functions to infer the following:

$$-F(E[X]) \leq E[-F(X)]$$

Bringing the negative sign outside the expectation on the right-hand side, one obtains the following:

$$-F(E[X]) \leq -E[F(X)]$$

On rearranging the inequality, one obtains the following:

$$F(E[X]) \geq E[F(X)]$$

It is also possible to generalize Jensen's inequality to multivariate random variables (and corresponding functions).

Example 11.1 (Sample-Level Jensen Inequality) Consider the values $\{1, 2, 3, 4, 5\}$, which are obtained using repeated samples of the face value X of a fair die. Compute the fourth moment $\hat{\mu}_{X^4}$ of the samples and the fourth power of the sample mean $\hat{\mu}_X^4$. Compare these values. Note that these are sample-level moments rather than expected values $E[X^4]$ and $E[X]^4$. Now compute $\hat{\mu}_{\sqrt{X}}$ and $\sqrt{\hat{\mu}_X}$ and compare them. Comment on the results of these comparisons.

Solution: The sample mean $\hat{\mu}_X$ can be computed as $(1+2+3+4+5)/5 = 3$. Therefore, we have $\hat{\mu}_X^4 = 3^4 = 81$. Similarly, we have $\hat{\mu}_{X^4} = (1^4+2^4+3^4+4^4+5^4)/5 \approx 195.8$. Therefore, we have $\hat{\mu}_{X^4} > \hat{\mu}_X^4$. A key point is that $F(x) = x^4$ is a convex function. Therefore, the above comparison is consistent with Jensen's inequality, except that it is computed at the sample level.

Using the sample mean value of 3 computed above, we have $\sqrt{\hat{\mu}_X} = \sqrt{3} \approx 1.732$. Similarly, we have $\hat{\mu}_{\sqrt{X}} = (\sqrt{1} + \sqrt{2} + \sqrt{3} + \sqrt{4} + \sqrt{5})/5 \approx 1.676$. Therefore, we have $\hat{\mu}_{\sqrt{X}} < \sqrt{\hat{\mu}_X}$. In this case, we are using the concave function $G(x) = \sqrt{x}$. Therefore, the above comparison is consistent with Jensen's inequality for concave functions (at the sample level). ■

Example 11.2 The variance σ_X^2 of random variable X is a nonnegative quantity that can be expressed as follows:

$$\sigma_X^2 = E[X^2] - E[X]^2 \geq 0$$

Discuss why this inequality is consistent with Jensen's inequality.

Solution: The function X^2 is convex. Therefore, by using Jensen's inequality, one can deduce that $E[X^2] \geq E[X]^2$. In other words, we have the following:

$$E[X^2] - E[X]^2 \geq 0$$

■

Problem 11.1 Let X be a random variable with zero mean. Show the following for any real value of t :

$$E[e^{tX}] \geq 1$$

Here, e is the base of the natural logarithm.

Problem 11.2 Let X_1 and X_2 be two independent and identically distributed random variables. Show that the following inequality is true for any real value of t :

$$E[e^{t(X_1 - E[X_1])}] \leq E[e^{t(X_1 - X_2)}]$$

Here, e is the base of the natural logarithm.

A hint for solving this problem is to first use Jensen's inequality to show that $e^{-tE[X_1]} \leq E[e^{-tX_1}] = E[e^{-tX_2}]$. Subsequently, multiply both sides with $E[e^{tX_1}]$ and use the independence of X_1 and X_2 to prove the result.

11.3 Markov and Chebychev Inequalities

The *Markov inequality* is one of the most fundamental tail inequalities, and it is defined for distributions that take on only non-negative values. Other than non-negativity, the Markov inequality does not make any assumptions. Let X be a non-negative random variable, with probability distribution $f_X(x)$ that has expected value $E[X]$ and variance $\text{Var}[X]$. The Markov inequality develops a bound on the probability of the upper tail of X .

Theorem 11.2 (Markov Inequality) *Let X be a random variable that takes on only non-negative values. Then, for any constant $\alpha > 0$, the following is true:*

$$P(X > \alpha) \leq E[X]/\alpha \quad (11.2)$$

Proof: Let $f_X(x)$ represent the density function for the random variable X . Then, we have:

$$\begin{aligned} E[X] &= \int_x x \cdot f_X(x) \cdot dx = \int_{0 \leq x \leq \alpha} x \cdot f_X(x) \cdot dx + \int_{x > \alpha} x \cdot f_X(x) \cdot dx \\ &\geq \int_{x > \alpha} x \cdot f_X(x) \cdot dx \geq \int_{x > \alpha} \alpha \cdot f_X(x) \cdot dx \end{aligned}$$

The first inequality follows from the non-negativity of x , and the second follows from the fact that the integral is defined only over the cases in which $x > \alpha$. Therefore, substituting α in place of x preserves the inequality.

The term on the right-hand side of the last equation above is exactly equal to $\alpha \cdot P(X > \alpha)$. Therefore, one can simplify the last equation to obtain the following:

$$E[X] \geq \alpha \cdot P(X > \alpha) \quad (11.3)$$

The aforementioned inequality can be re-arranged in order to obtain the result indicated in the statement of the theorem. ■

The Markov inequality is useful only for cases in which $\alpha > E[X]$, or else the Markov bound on the probability $P(X > \alpha)$ becomes larger than 1 (which provides no useful information since probabilities are never larger than 1). Furthermore, the Markov inequality is only an upper-tail bound. However, variations of the Markov inequality can be used for arbitrary random variables in the case that the random variable is bounded from below.

Corollary 11.2 (Translated Markov Inequality) *Let X be a random variable for which the domain of values is (a, ∞) . The lower-bound a is allowed to be negative. Then, for any constant $\alpha > a$, the following is true:*

$$P(X > \alpha) \leq \frac{E[X] - a}{\alpha - a} \quad (11.4)$$

The above inequality can be easily proven by defining the new random variable $Y = X - a$, and then applying the Markov inequality to this new random variable. We refer to the above inequality as the translated Markov inequality because it is obtained by translating the original random variable to a new one in which the nonnegativity constraint is satisfied. The translated Markov inequality is also useful in cases of nonnegative random variables where $a > 0$ because it tends to provide tighter bounds.

Example 11.3 The average humidity in the Sahara desert over time is 25%. Provide an upper bound on the probability that a randomly selected moment in the Sahara desert will have a humidity of more than 90%.

Solution: This problem is a straightforward application of the Markov inequality on the random variable X corresponding to the humidity level. Therefore, we have $E[X] = 25$ and $\alpha = 90$. This leads to an upper bound of $25/90 = 5/18$. ■

Example 11.4 The minimum height of all freshmen in a college over its entire history has been 3.8 feet. The mean height has been 5.5 feet. Find an upper bound on the probability that a randomly selected adult from its history was at least 7 feet tall as a freshman. Use both the standard version of the Markov inequality and the translated Markov inequality to arrive at two different results and compare them.

Solution: One can define a random variable X that is the height based on sampling from the historical population of college students and reporting their heights as freshmen. Using the standard version of the Markov inequality is at most $5.5/7 = 0.786$. Using the translated Markov inequality, one obtains the corresponding probability bound as $(5.5 - 3.8)/(7 - 3.8) = 0.531$. The second probability bound is clearly tighter. In other words, the second bound is more informative. ■

Problem 11.3 The average humidity in Rio de Janeiro over time is 75%. Provide an upper bound on the probability that a randomly selected moment in Rio de Janeiro will have a humidity of more than 90%. Suppose that you are additionally told that the humidity in Rio de Janeiro is never less than 60%. Use this fact to provide a tighter upper bound on the probability that a randomly selected moment in Rio de Janeiro will have a humidity of more than 90%.

Problem 11.4 Let X be a nonnegative random variable distributed in $[80, 100]$ with an expected value of 90. You wish to find the probability that the random variable is greater than 99. Show that the use of the vanilla Markov inequality leads to the following bound:

$$P(X > 99) \leq 90/99$$

Furthermore, show that the use of the translated Markov inequality leads to a much tighter bound:

$$P(X > 99) \leq 10/19$$

The Markov inequality provides a bound only on the upper tail. It turns out that the Markov inequality can also be used to provide a bound for the lower-tail as long as the random variable is bounded from above. This is achieved with a reflected and translated Markov inequality.

Corollary 11.3 (Translated and Reflected Markov Inequality) Let X be a random variable for which the domain of values is $(-\infty, a)$. Then, for any constant $\alpha < a$, the following is true:

$$P(X < \alpha) \leq \frac{a - E[X]}{a - \alpha} \quad (11.5)$$

The above result is easy to show by defining the translated and reflected random variable $Y = a - X$. The inequality is a lower-tail inequality rather than an upper tail inequality because Y is a decreasing function of X . Therefore, when X is less than a particular value α , Y is greater than the corresponding function of α (as required by the Markov inequality). In order to understand the nature of the above bound, we encourage the reader to peruse the worked example and solve the problem following it:

Example 11.5 *A college will auto-reject all candidates who score less than 70% on a standardized examination. The average score on the examination from the population of candidates is 95%. Find an upper bound on the fraction of candidates who will be auto-rejected.*

Solution: One can assume that the score is out of 100 in order to work with an upper bound on the scores. Using the translated and reflected Markov inequality, an upper bound on the fraction of candidates who will be auto-rejected is $(100 - 95)/(100 - 70) = 1/6$. ■

Problem 11.5 *Let X be a random variables that is always less than 10 and has an expected value of -20 . Show the following lower-tail inequality:*

$$P(X < -80) \leq 1/3$$

In practice, it is often desired to bound the probability contained in both tails of arbitrary distributions, when the domain of the random variable is $(-\infty, +\infty)$. Consider the case where X is an arbitrary random variable, which is not necessarily non-negative or bounded at either tail. In such cases, the Markov inequality and its variants cannot be used directly. However, the (related) *Chebychev inequality* is very useful in such cases. The Chebychev inequality is a direct application of the Markov inequality to a non-negative function of random variable X :

Theorem 11.3 (Chebychev Inequality) *Let X be an arbitrary random variable. Then, for any constant α , the following is true:*

$$P(|X - E[X]| > \alpha) \leq Var[X]/\alpha^2 \quad (11.6)$$

Proof: The inequality $|X - E[X]| > \alpha$ is true if and only if $(X - E[X])^2 > \alpha^2$. By defining $Y = (X - E[X])^2$ as a (non-negative) derivative random variable from X , it is easy to see that $E[Y] = Var[X]$. Then, the expression on the left hand side of the theorem statement is the same as the probability $P(Y > \alpha^2)$. By applying the Markov inequality to the random variable Y , one can obtain the desired result. ■

The main trick used in the aforementioned proof was to apply the Markov inequality to a non-negative function of the random variable. This technique can generally be very useful for proving other types of bounds, when the distribution of X has a specific form (such as the sum of Bernoulli random variables). In such cases, a parameterized function of the random variable can be used in order to obtain a parameterized bound. The underlying parameters can then be optimized for the tightest possible bound. Several well-known bounds such as the Chernoff bound and the Hoeffding inequality are derived with the use of this approach.

The Markov and Chebychev inequalities are relatively weak inequalities and often do not provide tight enough bounds to be useful in many practical scenarios. This is because these inequalities do not make any assumptions on the nature of the random variable X .

Many practical scenarios can however be captured, when stronger assumptions are used on the random variable. For example, if it is known that a random variable is the sum of several i.i.d. random variables with particular boundedness properties, it becomes possible to use this information while computing the bound probabilities. In such cases, much tighter bounds on tail distributions are possible.

Example 11.6 *The mean of a random variable X is 3 and its standard deviation is 4. Show that the probability that its absolute value $|X|$ is greater than 10 is at most 1/4.*

Solution: The statement of this problem looks a lot like the Chebychev inequality except that we are trying to bound $|X|$ rather than $|X - \mu_X|$. Just as the Chebychev inequality works with the variance, this problem works with the second moment. We are already given the mean $\mu_X = 3$ and standard deviation $\sigma_X = 4$, and so the second moment is $E[X^2] = \mu_X^2 + \sigma_X^2 = 25$. Therefore, we have the following:

$$P(|X| > 10) = P(X^2 > 100)$$

Since X^2 is a nonnegative random variable, one can use the Markov inequality to conclude the following:

$$P(X^2 > 100) \leq E[X^2]/100 = 25/100 = 1/4$$

■

Problem 11.6 *It has been discussed several times in this book that more than 99.9% of the data lies within three standard deviations of the mean in the case of a normal distribution. But what happens in the general case, when the distribution has an arbitrary shape? Show that at least 8/9 of the data lies within three standard deviations of the mean (irrespective of the nature or shape of the distribution).*

Problem 11.7 *Consider a bounded random variable X that lies in the bounds (a, b) and expected value $E[X]$. The random variable X is not necessarily nonnegative. Show that it is possible to develop both upper- and lower-tail bounds with tail thresholds $\alpha > E[X]$ and $\beta < E[X]$:*

$$\begin{aligned} P(X > \alpha) &\leq \frac{E[X] - a}{\alpha - a} \\ P(X < \beta) &\leq \frac{b - E[X]}{b - \beta} \end{aligned}$$

The above problem shows that the Markov inequality can be used to develop both upper and lower bounds on the random variable. The main constraint is that the random variable must be bounded from both above and below. The proof of the above problem is not very difficult, and it involves combining reflection and translation results that have already been discussed in this chapter.

11.4 Approximations for Sums of Random Variables

Many practical observations, which are defined in the form of *aggregates*, can be expressed as sums of bounded or binary random variables. The fact that the random variables are bounded or binary is important because it provides a specialized structure to the sum of the random variables. In such cases, the special form of the random variables can be used to tighten the results of the Markov inequality. In other words, all these inequalities bound random variables of the following form:

$$Y = \sum_{i=1}^n X_i$$

Here, the different X_i are i.i.d. random variables, and each X_i is either binary or is a bounded value in a range $[l, u]$. Binary random variables are relevant to the Chernoff bound, whereas Bernoulli random variables are relevant to the Hoeffding inequality. In order to motivate the Chernoff bound and the Hoeffding inequality, we provide some motivating examples:

Example 11.7 (Sports Statistics) *The National Basketball Association (NBA) draft teams have access to college basketball statistics for the different candidate players. For each player and each game, their scoring statistics are maintained. For example, the number of dunks, assists, and rebounds is maintained for each game. For any particular scoring statistic (e.g., number of dunks), we want to find the anomalous players. Show how tight tail inequalities can help identify such players.*

Solution: For a particular statistic (e.g., number of dunks), the aggregate performance of any player can be expressed as the sum of their statistics over n different games:

$$Y = \sum_{i=1}^n X_i$$

All values of X_i lie in the range $[l, u]$. This is because most such scoring statistics can be assumed to lie in particular practical ranges. For example, it is impossible for a single player to make more than 100 dunks in a game, and therefore these values can be assumed to lie in $[0, 100]$. The performances of a player over different games are assumed to be independent of one another. The long-term global mean of the statistic represented by X_i over all players is known to be μ . The NBA draft teams would like to identify the anomalous players on the basis of each statistic.

In this example, the aggregate statistic is represented as a sum of bounded random variables. The corresponding tail bounds can be quantified with the use of the *Hoeffding inequality*. ■

In many cases, the individual random variable components in the aggregation are not only bounded, but also binary 0-1 values. Thus, the aggregate statistic can be expressed as a sum of Bernoulli random variables.

Example 11.8 (Grocery Shopping) *A grocery store keeps track of the number of customers (from its frequent purchaser program), who behave independently of one another. The long-term probability of any customer i attending the store on a given*

day is known to be p_i . For a given day, show how to use tight tail inequalities in order to bound the probability of receiving more than η (frequent purchase program) customers.

Solution: In this example, the number of customers can be expressed as a sum of independent Bernoulli random variables. This is because X_i is a random variable with a 0-1 value indicating whether the i th customer visits the store on that day. In such a case, our job is to find the probability that $\sum_i X_i$ is greater than η . The corresponding tail distributions can be expressed in terms of the *Chernoff bound*. ■

These types of bounds are frequently used in different types of machine learning applications. For example, we provide a very common application of anomaly detection from aggregates, which is that of fault diagnosis in manufacturing in which a *worst-case guarantee* of the number of anomalous (faulty) products is obtained with the use of such bounds:

Example 11.9 (Manufacturing Quality Control) An assembly line creates products, each of which is faulty with probability p . A quality-control process samples n products from the assembly line. Show how to use tight tail inequalities to find an upper-bound on the probability that the sample contains more than η faulty products.

Solution: In this case X_i is a Bernoulli random variable indicating whether or not the i th sampled product has a defect. Therefore, the total number of defects is a random variable given by $\sum_{i=1}^n X_i$. This type of random variable can be bounded using the Chernoff bound, because it is the sum of i.i.d. Bernoulli random variables. ■

The sample size n is typically large, and, therefore, it is possible to use the Central Limit Theorem introduced in Chapter 5. According to this theorem, the sum of a large number of independent and identical normal distributions converges to a normal distribution. By using the central limit theorem, one can use the shape of the probability distribution to obtain a good estimate of the probability of the tail. The quality of this estimate heavily depends on how large the value of n might be. Furthermore, for certain types of random variables that are defined by sums of Bernoulli random variables, a Poisson approximation can be used to estimate the tail bounds. However, the estimate provided by the normal or the Poisson approximation does not provide a worst-case estimate. The accuracy of the approximation depends heavily how well the normal or Poisson assumption fits the scenario at hand. On the other hand, the Chernoff bound can be used to provide a worst-case estimate.

Both the Chernoff and the Hoeffding bounds are descendants of the Markov inequality, which can be tightened considerably for random variables with special structures. Although the Markov inequality is an upper-tail bound, we have already seen that it can be used to develop lower-tail bounds as well in many cases with appropriate functional transformations. As indicated in Problem 11.7, both upper-tail and lower-tail bounds can be developed for random variables defined over finite bounds (a, b) . This is also the case for the Chernoff and Hoeffding bounds, which are defined over bounded random variables.

11.4.1 The Chernoff Bound

Since the expressions for the lower tail and upper tails are slightly different, they will be addressed separately. The lower-tail Chernoff bound is introduced below.

Theorem 11.4 (Lower-Tail Chernoff Bound) *Let X be random variable that can be expressed as the sum of n independent binary (Bernoulli) random variables, each of which takes on the value of 1 with probability p_i .*

$$X = \sum_{i=1}^n X_i$$

Then, for any $\delta \in (0, 1)$, we can show the following:

$$P(X < (1 - \delta) \cdot E[X]) < e^{-E[X] \cdot \delta^2 / 2} \quad (11.7)$$

Here, the notation e denotes the base of the natural logarithm.

Proof: The first step is to show the following inequality:

$$P(X < (1 - \delta) \cdot E[X]) < \left(\frac{e^{-\delta}}{(1 - \delta)^{(1-\delta)}} \right)^{E[X]} \quad (11.8)$$

The unknown parameter $t > 0$ is introduced in order to create a parameterized bound. The purpose of introducing this unknown parameter is to obtain the tightest form of the Markov inequality by choosing t carefully. The lower-tail inequality of X is converted into an upper-tail inequality on the exponentiated expression $e^{-t \cdot X}$. This random expression can be bounded by the Markov inequality, and it provides a bound as a function of t . This function of t can be optimized in order to obtain the tightest possible bound. By using the Markov inequality on the exponentiated form, the following can be derived:

$$P(X < (1 - \delta) \cdot E[X]) \leq \frac{E[e^{-t \cdot X}]}{e^{-t \cdot (1-\delta) \cdot E[X]}}$$

By expanding $X = \sum_{i=1}^n X_i$ in the exponent, the following can be obtained:

$$P(X < (1 - \delta) \cdot E[X]) \leq \frac{\prod_{i=1}^n E[e^{-t \cdot X_i}]}{e^{-t \cdot (1-\delta) \cdot E[X]}} \quad (11.9)$$

The aforementioned simplification uses the fact that the expectation of the product of independent variables is equal to the product of the expectations. Since each X_i is Bernoulli, the following can be shown:

$$\begin{aligned} E[e^{-t \cdot X_i}] &= 1 + E[X_i] \cdot (e^{-t} - 1) \\ &< e^{E[X_i] \cdot (e^{-t} - 1)} \end{aligned}$$

The first relationship follows using the definition of expectation over $X_i = 0$ and $X_i = 1$. The second relationship (inequality) follows from polynomial expansion of $e^{E[X_i] \cdot (e^{-t} - 1)}$. By substituting this inequality back into Equation 11.9, and using $E[X] = \sum_i E[X_i]$, the following may be obtained:

$$P(X < (1 - \delta) \cdot E[X]) \leq \frac{e^{E[X] \cdot (e^{-t} - 1)}}{e^{-t \cdot (1-\delta) \cdot E[X]}} \quad (11.10)$$

The expression on the right is true for any value of $t > 0$. It is desired to determine the value of t that provides the *tightest possible*¹ bound. Such a value of t may be obtained by computing the derivative of the expression with respect to t and setting it to 0. It can be shown that the resulting value of $t = t^*$ from this optimization process is as follows:

$$t^* = \ln(1/(1 - \delta)) \quad (11.11)$$

Substituting the value of t^* in the right-hand side of Equation 11.10, one obtains the following inequality:

$$P(X < (1 - \delta) \cdot E[X]) < \left(\frac{e^{-\delta}}{(1 - \delta)^{(1-\delta)}} \right)^{E[X]} \quad (11.12)$$

This completes the first part of the proof.

In order to further complete the proof, we focus on the denominator of the above expression. The logarithm of the denominator is $(1 - \delta) \cdot \ln(1 - \delta)$. The Taylor expansion of $\ln(1 - \delta)$ can be used to conclude the following:

$$\ln(1 - \delta) = -\delta - \delta^2/2 - \delta^3/3 - \dots - \delta^n/n - \dots -$$

Multiplying both sides with $(1 - \delta)$, one obtains the following:

$$(1 - \delta)\ln(1 - \delta) = -\delta + \delta^2/2 + \text{positive terms} +$$

Therefore, one can conclude that $(1 - \delta)\ln(1 - \delta) > -\delta + \delta^2/2$. Exponentiating this inequality, one obtains the following:

$$e^{(1-\delta)\ln(1-\delta)} > e^{-\delta+\delta^2}$$

$$(1 - \delta)^{(1-\delta)} > e^{-\delta+\delta^2/2}$$

Note that the left-hand side of the above expression is the denominator of the right-hand side of the inequality of Equation 11.12. Therefore, by replacing the denominator on the right-hand side of Equation 11.12 with $e^{-\delta+\delta^2/2}$, one obtains the following:

$$P(X < (1 - \delta) \cdot E[X]) < \left(\frac{e^{-\delta}}{e^{-\delta+\delta^2/2}} \right)^{E[X]} = e^{-E[X] \cdot \delta^2/2}$$

This completes the proof of the lower-tail Chernoff bound. ■

We have already seen in an earlier section that the Markov inequality can provide tighter bounds if the characteristics of the underlying distribution are leveraged. A simple example is one in which the lower bound of a distribution is used in order to translate the Markov inequality. The proof given above is a much more sophisticated example of how the specific characteristics of a distribution are used to tighten the Markov inequality.

A similar result for the upper-tail Chernoff bound may be obtained, albeit in a slightly different form.

Theorem 11.5 (Upper-Tail Chernoff Bound) *Let X be random variable, which is expressed as the sum of n independent binary (Bernoulli) random variables, each of which takes on the value of 1 with probability p_i .*

$$X = \sum_{i=1}^n X_i$$

¹This is yet another example of how the (weak) Markov inequality can be tightened by leveraging the specific characteristics of the random variable at hand. Optimizing t is an indirect way of doing this because the underlying function of t was developed using the characteristics of the distribution of X .

Then, the following bounds are true, depending on the range of the values of δ :

$$P(X > (1 + \delta) \cdot E[X]) < \begin{cases} e^{-E[X] \cdot \delta^2 / 4} & \text{if } 0 \leq \delta \leq 2 \cdot e - 1 \\ 2^{-\delta E[X]} & \text{if } \delta > 2 \cdot e - 1 \end{cases} \quad (11.13)$$

Here, e is the base of the natural logarithm.

Proof: The first step is to show the following inequality:

$$P(X > (1 + \delta) \cdot E[X]) < \left(\frac{e^\delta}{(1 + \delta)^{(1+\delta)}} \right)^{E[X]} \quad (11.14)$$

Note that a similar inequality was shown in the case of the lower-tail Chernoff bound (except that the two inequalities differ in terms of the sign in front of δ). As in the case of the lower-tail Chernoff bound, this inequality can be shown by introducing the unknown parameter $t > 0$, and converting the upper-tail inequality on X into that on $e^{t \cdot X}$. This can be bounded by the Markov inequality as a function of t . This function of t can be optimized, in order to obtain the tightest possible bound. Since these steps are almost identical to the case of the lower-tail Chernoff bound, the specific details have been omitted. Next, we will show that the inequality of Equation 11.14 is equivalent to the final result in each of the two cases $\delta > 2 \cdot e - 1$ and $\delta \in (0, 2 \cdot e - 1)$.

First, we consider the case that $\delta > 2 \cdot e - 1$. First, we use the fact that $(1 + \delta)^{(1+\delta)} > (1 + \delta)^\delta$ to simplify the denominator on the right-hand side of Equation 11.14 as follows:

$$\begin{aligned} P(X > (1 + \delta) \cdot E[X]) &< \left(\frac{e^\delta}{(1 + \delta)^{(1+\delta)}} \right)^{E[X]} \\ &< \left(\frac{e^\delta}{(1 + \delta)^\delta} \right)^{E[X]} \\ &= \left(\frac{e}{1 + \delta} \right)^{\delta E[X]} \end{aligned}$$

However, since we already know that $\delta > 2 \cdot e - 1$ in this case, the value of $e/(1 + \delta)$ in the above expression can be shown to be less than $1/2$. Therefore, one can simplify the above inequality as follows:

$$\begin{aligned} P(X > (1 + \delta) \cdot E[X]) &< \left(\frac{e}{1 + \delta} \right)^{\delta E[X]} \\ &< \left(\frac{1}{2} \right)^{\delta E[X]} = 2^{-\delta E[X]} \end{aligned}$$

Next, we consider the case where $\delta \in (0, 2 \cdot e - 1)$. In this case, the argument is more complex, which is omitted here. However, in such a case, it can be shown that the inequality in Equation 11.14 can be simplified to the following:

$$P(X > (1 + \delta) \cdot E[X]) < e^{-E[X] \cdot \delta^2 / 4}$$

Combining the two cases, one obtains the desired result. ■

Although the bounds introduced in the earlier theorems represent common forms of the Chernoff bound, the intermediate results are also considered the “raw” forms of the Chernoff bound without further weakening:

Theorem 11.6 (Raw form of Chernoff Bound) *Let X be random variable that can be expressed as the sum of n independent binary (Bernoulli) random variables, each of which takes on the value of 1 with probability p_i .*

$$X = \sum_{i=1}^n X_i$$

Then, for any $\delta \in (0, 1)$, we can show the following:

$$\begin{aligned} [\text{Lower Tail}]: P(X < (1 - \delta) \cdot E[X]) &< \left(\frac{e^{-\delta}}{(1 - \delta)^{(1-\delta)}} \right)^{E[X]} \\ [\text{Upper Tail}]: P(X > (1 + \delta) \cdot E[X]) &< \left(\frac{e^\delta}{(1 + \delta)^{(1+\delta)}} \right)^{E[X]} \end{aligned}$$

These raw forms of the Chernoff bound have already been shown in earlier results.

The Chernoff bound is useful only for bounding random variables that are defined by sums of independent Bernoulli random variables (but may not have the same value of the success parameter). An interesting special case of the Chernoff bound is when all Bernoulli random variables have the same value of the Bernoulli success parameter (and therefore belong to the binomial distribution).

Corollary 11.4 (Chernoff Bound for Binomial Distribution) *Consider a binomial distribution with parameters n and p , where n is the number of trials and p is the success probability:*

$$X \sim B(n, p)$$

Then, the upper-tail and lower-tail bounds can be expressed in terms of n and p as follows:

$$\begin{aligned} [\text{Lower-Tail}]: P(X < (1 - \delta) \cdot n \cdot p) &< e^{-n \cdot p \cdot \delta^2 / 2} \\ [\text{Upper-Tail}]: P(X > (1 + \delta) \cdot n \cdot p) &< \begin{cases} e^{-n \cdot p \cdot \delta^2 / 4} & \text{if } 0 \leq \delta \leq 2 \cdot e - 1 \\ 2^{-\delta \cdot n \cdot p} & \text{if } \delta > 2 \cdot e - 1 \end{cases} \end{aligned}$$

The above corollary is easy to prove by replacing each occurrence of $E[X]$ with $n \cdot p$ in the original Chernoff bound.

The binomial distribution is well studied and many average-case approximations exist (including the normal and the Poisson approximation). Some of these approximations will be studied in this section.

Example 11.10 Suppose that you flip a coin 100 times. Use the Chernoff bound to determine a worst-case bound on the probability that the number of heads is less than 40 or more than 60.

Solution: In this case, the values of 40 and 60 are 20% away from the mean value of 50. Therefore, we need to use the Chernoff lower-tail and upper-tail bounds for $\delta = 0.2$ and $E[X] = 50$. In the case of $\delta = 0.2$, the lower-tail bound is given by the following:

$$P(X < 40) < e^{-50*(0.2)^2/2} = 1/e$$

For the upper-tail bound, the value of $\delta = 0.2$ is less than $(2 \cdot e - 1)$, and therefore the bound is given by $e^{-E[X] \cdot \delta^2/4}$. In such a case, the upper-tail bound is given by the following:

$$P(X > 60) < e^{-50*(0.2)^2/4} = 1/\sqrt{e}$$

Therefore, the sum of the tail probabilities is given by *at most* $(1 + \sqrt{e})/e = 0.974$. Note that this type of bound is not particularly informative because it is only an upper bound, and 0.974 is very close to 1 anyway.

One can obtain tighter results with the raw form of the Chernoff bound in Theorem 11.6. According to this form of the Chernoff bound, we have the following:

$$\begin{aligned} P(X < 40) &< \left(\frac{e^{-0.2}}{0.8^{0.8}} \right)^{50} \approx 0.342 \\ P(X > 60) &< \left(\frac{e^{0.2}}{1.2^{1.2}} \right)^{50} \approx 0.400 \end{aligned}$$

The use of the raw Chernoff bound provides a worst-case tail probability of $0.342 + 0.400 = 0.742$. While this bound is tighter than the value of 0.974 obtained earlier, it is still a large probability. ■

Increasing the number of coin flips increases the tightness of the Chernoff bound in an exponential manner. In order to understand this point, we recommend the reader to solve the following problem in which a larger number of flips is used.

Problem 11.8 Suppose that you flip a coin 1000 times. Use the Chernoff bound to determine a worst-case bound on the probability that the number of heads are less than 400 or more than 600.

11.4.2 The Normal Approximation to the Binomial Distribution

The binomial distribution is defined using two parameters n and p , where p is the success probability and n is the number of trials. Consider the random variable X , which is drawn from such a binomial distribution:

$$X \sim \text{B}(n, p)$$

As discussed in Chapter 4, the binomial distribution converges to the normal distribution when the value n is large. However, when p is close to either 0 or 1, much larger values of n are required to approximate a (symmetric) normal distribution with such asymmetric probability values. Note that the sum of i.i.d. variables always converge to a normal distribution, but the convergence to the normal distribution occurs much faster when the underlying Bernoulli distribution is symmetric. A rule of thumb is that the values of both $n \cdot p$ and $n \cdot (1 - p)$ need to be larger than 10. Note that when p is close to 0 or 1, the value of n required will be much larger based on the rule of thumb above. In order to understand this point, we present the binomial distributions introduced in Chapter 4 for varying values of n and p in Figures 11.2 and 11.3. Here, it is evident that the distributions converge to the normal distribution when n is large enough. For asymmetric binomial distributions, large values of n are required in order for the binomial distribution to take on the shape of the normal bell curve. For example, the binomial distribution for $n = 10$ and $p = 1/2$ already starts taking on the bell curve shape at the modest number of trials (cf. Figure 11.2(b)).

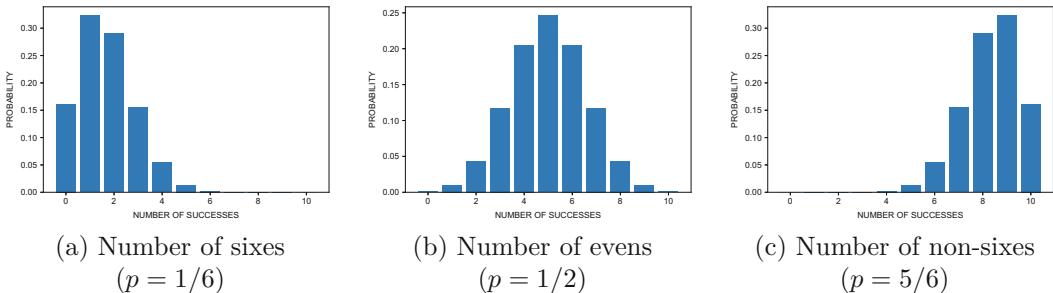


Figure 11.2: Revisiting Figure 4.4: The probability mass functions of different binomial distributions at fixed $n = 10$ and varying p for die roll events

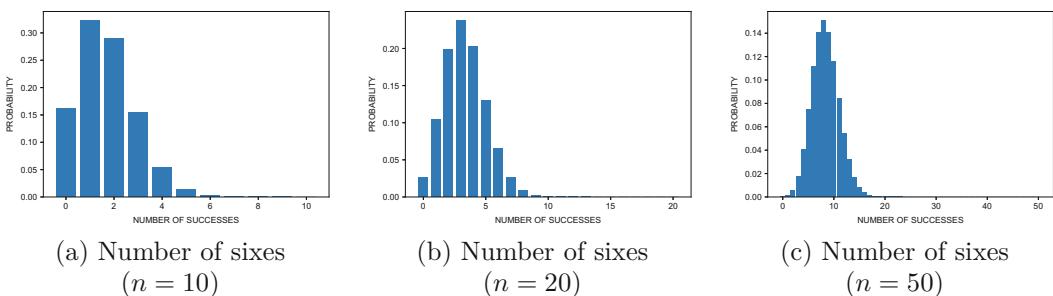


Figure 11.3: Revisiting Figure 4.5: The probability mass functions of different binomial distributions at fixed $p = 1/6$ and varying n for die roll events

However, for $p = 1/6$, a large value of $n = 20$ still looks quite different from the normal distribution (cf. Figure 11.3(b)). In other words, asymmetry can be compensated with larger sample sizes. This pair of desiderata on symmetry and number of trials is best captured using the rule of thumb that $\min\{n \cdot p, n \cdot (1 - p)\} \geq 10$. The normal approximation is least accurate near the tail of the binomial distribution, where the asymmetry tends to get emphasized and the outcome is the aggregation of a smaller number of successes/failures.

The mean and variance of the binomial distribution with parameters n and p is given by $n \cdot p$ and $n \cdot p(1 - p)$. The normal approximation $X_a \approx X$ of the binomial random variable $X \sim B(n, p)$ adopts this mean and variance:

$$X_a \sim \mathcal{N}(np, np(1-p))$$

One problem with using the normal distribution to approximate the binomial distribution is that the normal distribution is continuous, whereas the binomial distribution is discrete. For example, the probability that a discrete random variable takes on a specific value is usually finite or 0, whereas the probability that a continuous random variable takes on a specific value is usually² infinitesimal or 0. In the specific scenario at hand, the value of $P(X = k)$ is always finitely large for any natural number k in $[1, n]$, whereas the value of $P(X_a = k)$ is always infinitesimally small. In order to address this issue, it is necessary to use the *continuity correction factor*. The continuity correction factor relates the discrete

²It is possible to incorporate finite probabilities for specific values with the use of *impulse functions*.

random variable X to the continuous random variable X_a as follows:

$$\begin{aligned} P(X < k) &\approx P(X_a < k - 0.5) \\ P(X \leq k) &\approx P(X_a < k + 0.5) \\ P(X > k) &\approx P(X_a > k + 0.5) = 1 - P(X_a < k + 0.5) \\ P(X \geq k) &\approx P(X_a > k - 0.5) = 1 - P(X_a < k - 0.5) \\ P(X = k) &= P(X \leq k) - P(X < k) \approx P(X_a < k + 0.5) - P(X_a < k - 0.5) \end{aligned}$$

All of the above binomial approximations are expressed in terms of the cumulative distribution function of the normal distribution. This type of normal approximation is typically much tighter than the Chernoff bound. However, an approximation is not a worst-case bound. In order to understand this point, let us revisit Example 11.10, in which a bound on tail probabilities was obtained with the Chernoff bound:

Example 11.11 Suppose that you flip a fair coin 100 times. Use the normal approximation to the binomial distribution to determine an estimate of the probability that the number of heads is either less than 40 or more than 60.

Solution: In this random process, the underlying binomial distribution has a mean of 50 and a variance of 25 (or standard deviation of 5). A corresponding normal distribution with mean of 50 and standard deviation of 5 is used to approximate the binomial distribution. Therefore, the Z-value for the upper-tail is as follows:

$$z_u = \frac{60.5 - 50}{5} = 2.1$$

The probability $P(Z > z_u)$ can be obtained from the normal distribution tables, and it is equal to $(1 - 0.9821) = 0.0179$. In other words, the upper-tail probability estimate is 1.79%. By using a symmetric argument, one can show that the lower-tail probability estimate is also 1.79%. Therefore, the total tail probability obtained by summing these probabilities is 3.58%. ■

It is instructive to compare this low probability value to the Chernoff bound, which provides an *upper-bound* on this tail probability to be 97.4% (cf. Example 11.10). This upper bound is practically useless, as it conveys very little information (because tail probabilities are supposed to be much smaller). The probability estimate using the normal approximation is much tighter than the Chernoff bound, and it is therefore much more informative. On the other hand, the probability estimate using the normal approximation is not guaranteed to be a *worst-case* bound like the Chernoff bound. It is only an *estimate*.

Problem 11.9 Suppose that you flip a fair coin 1000 times. Use the normal approximation to the binomial distribution to determine a worst-case bound on the probability that the number of heads is either less than 400 or more than 600.

11.4.3 The Poisson Approximation to the Binomial Distribution

The Poisson distribution is the limiting case of the binomial distribution, when the Bernoulli arrival rate is applied over infinitesimal lengths of time in an interval in order to estimate the total number of arrivals. This connection between the binomial and Poisson distributions

is used to introduce the Poisson distribution in section 4.9 of Chapter 4. It turns out that this connection also enables the Poisson approximation to the normal distribution. A rule of thumb for the Poisson approximation to the binomial distribution is as follows:

The Poisson approximation is accurate when the number of trials n of the binomial distribution is large (i.e., at least 100) but the success rate p is small enough that the expected number of successes $n \cdot p$ is at most 10.

In such a case, one can assume that an exponential process is occurring with arrival rate $\lambda = n \cdot p$. The probability of k arrivals in one time unit is a Poisson random variable with parameter $\lambda = n \cdot p$. It can be shown that this probability approximates the probability of k successes in the case of the binomial distribution. More formally, let X be the binomial random variable with parameters n and p :

$$X \sim B(n, p)$$

Here, it is assumed that the binomial distribution satisfies the constraints on n and p that are discussed above. Let X_a be the Poisson random variable with parameter $\lambda = n \cdot p$:

$$X_a \sim \text{Poisson}(n p)$$

Note that both X and X_a are discrete random variables, and therefore no continuity correction is needed. In this case, the Poisson approximation implies the following:

$$P(X = k) \approx P(X_a = k)$$

In some cases, it is easier to evaluate the Poisson probability as opposed to a combinatorial binomial expression.

Example 11.12 Consider a biased coin with probability of heads to be 0.002. You flip the coin a thousand times. Find the probability of exactly 3 heads. Use both the binomial distribution and its Poisson approximation to determine the exact and approximate probabilities, respectively.

Solution: If the binomial distribution is used exactly, then the probability of three heads is as follows:

$$P(X = 3) = \binom{1000}{3} 0.998^{997} (0.002)^3 \approx 0.18063$$

One can now perform the same computation with the use of the Poisson approximation. Note that the value of λ is $1000 * 0.002 = 2$. The probability of $X_a = 3$ is as follows:

$$P(X_a = 3) = \frac{e^{-2} \cdot 2^3}{3!} \approx 0.18045$$

The striking accuracy of the Poisson approximation is quite noteworthy. ■

Problem 11.10 Consider a biased coin with probability of heads to be 0.002. You flip the coin a thousand times. Find the probability of at most 3 heads. Use both the binomial distribution to compute this probability exactly and the Poisson approximation to estimate this probability.

11.4.4 The Hoeffding Inequality

The Hoeffding inequality is a more general tail inequality than the Chernoff bound, because it does not require the underlying random variables to be drawn from a Bernoulli distribution. As in the case of the Chernoff bound, these random variables are added to create a new random variable whose tail probabilities are bounded. In this case, the i th random variable needs to be drawn from the *bounded interval* $[l_i, u_i]$. The corresponding probability bound is expressed in terms of the parameters l_i and u_i . Thus, the scenario for the Chernoff bound is a special case of that for the Hoeffding inequality. We state the Hoeffding inequality below, for which the upper- and lower-tail inequalities are identical.

Theorem 11.7 (Hoeffding Inequality) *Let X be a random variable that can be expressed as the sum of n independent random variables, each of which is bounded in the range $[l_i, u_i]$.*

$$X = \sum_{i=1}^n X_i$$

Then, for any $\theta > 0$, the following can be shown:

$$\begin{aligned} P(X - E[X] > \theta) &\leq e^{-\frac{2\cdot\theta^2}{\sum_{i=1}^n (u_i - l_i)^2}} \\ P(E[X] - X > \theta) &\leq e^{-\frac{2\cdot\theta^2}{\sum_{i=1}^n (u_i - l_i)^2}} \end{aligned}$$

Proof: The proof of the upper-tail bound will be briefly described here. The proof of the lower-tail inequality is identical. For any choice parameter $t \geq 0$, the following is true:

$$P(X - E[X] > \theta) = P(e^{t \cdot (X - E[X])} > e^{t \cdot \theta}) \quad (11.15)$$

The Markov inequality can be used to show that the right-hand probability is at most $E[e^{t \cdot (X - E[X])}] \cdot e^{-t \cdot \theta}$. The expression within $E[e^{t \cdot (X - E[X])}]$ can be expanded in terms of the individual components X_i . Since the expectation of the product is equal to the product of the expectations of independent random variables, the following can be shown:

$$P(X - E[X] > \theta) \leq e^{-t \cdot \theta} \cdot \prod_i E[e^{t \cdot (X_i - E[X_i])}] \quad (11.16)$$

The key is to show that the value of $E[e^{t \cdot (X_i - E[X_i])}]$ is at most equal to $e^{t^2 \cdot (u_i - l_i)^2 / 8}$ for any positive value of t . The proof³ of this inequality is somewhat involved and therefore omitted. Exercise 5 provides a guided step-by-step reconstruction of this proof.

By using the resulting inequality $E[e^{t \cdot (X_i - E[X_i])}] \leq e^{t^2 \cdot (u_i - l_i)^2 / 8}$ in order to substitute for the value of $E[e^{t \cdot (X_i - E[X_i])}]$ in Equation 11.16, the following inequality results:

$$P(X - E[X] > \theta) \leq e^{-t \cdot \theta} \cdot \prod_i e^{t^2 \cdot (u_i - l_i)^2 / 8} \quad (11.17)$$

This inequality holds for any positive value of t . Therefore, in order to find the tightest bound, the value of t that minimizes the right-hand side of the above equation needs to be determined. The optimal value of $t = t^*$ can be shown to be the following using differentiation:

$$t^* = \frac{4 \cdot \theta}{\sum_{i=1}^n (u_i - l_i)^2} \quad (11.18)$$

³The proof uses the convexity of the exponential function in conjunction with Taylor's theorem.

By substituting the value of $t = t^*$ in Equation 11.17, the desired result may be obtained for the upper-tail bound. The proof of the lower-tail bound is very similar, and it may be obtained by applying analogous steps to $P(E[X] - X > \theta)$ rather than $P(X - E[X] > \theta)$. ■

The Hoeffding inequality is more general than the Chernoff bound, since Bernoulli random variables are bounded by default. We provide the specialization of the Hoeffding inequality for the case where each variable being added is Bernoulli with success parameter of p . This is precisely the case addressed by the Chernoff bound:

Corollary 11.5 (Hoeffding Inequality for Binomial Distribution) *Consider a binomial distribution with parameters n and p , where n is the number of trials and p is the success probability:*

$$X \sim B(n, p)$$

Then, the upper-tail and lower-tail bounds can be expressed in terms of n and p as follows:

$$[\text{Lower-Tail}]: P(X < np(1 - \delta)) \leq e^{-2np^2\delta^2}$$

$$[\text{Upper-Tail}]: P(X > np(1 + \delta)) \leq e^{-2np^2\delta^2}$$

The above corollary is easy to prove by replacing each occurrence of $E[X]$ with $n \cdot p$ in the original Hoeffding inequality.

Since the Hoeffding inequality is more general than the Chernoff bound, it would stand to reason that the Chernoff bound should be tighter since it incorporates additional information. However, this is not always the case, and there are many settings in which the Hoeffding inequality provides a tighter bound. Therefore, when both bounds apply, it is worthwhile to test both inequalities to check which one provides a tighter bound. We will revisit Example 11.10 and provide a worst-case bound with the Hoeffding inequality instead of the Chernoff bound:

Example 11.13 Suppose that you flip a coin 100 times. Use the Hoeffding inequality to determine a worst-case bound on the probability that the number of heads is less than 40 or more than 60.

Solution: As discussed above, the Hoeffding inequality may be stated as follows:

$$P(X - E[X] > \theta) \leq e^{-\frac{2\cdot\theta^2}{\sum_{i=1}^n (u_i - l_i)^2}}$$

$$P(E[X] - X > \theta) \leq e^{-\frac{2\cdot\theta^2}{\sum_{i=1}^n (u_i - l_i)^2}}$$

Here, we have $n = 100$ and $\theta = 10$. Furthermore, l_i and u_i are 0 and 1, respectively. On substituting these values, both the upper bound $P([X - 50] > 0.2 * 50)$ and the lower bound $P([50 - X] > 0.2 * 50)$ turn out to be $e^{-2 \cdot 10^2 / 100} = 1/e^2$. Therefore, the sum of the two symmetric tail probabilities is $2/e^2 \approx 0.27067$. This is a tighter bound than what is given by the Chernoff bound. Nevertheless, it is still quite weak compared to the estimate provided by the normal approximation to the binomial distribution. Again, this difference is because the normal approximation does not provide a worst-case bound but an averaged estimate. ■

Note that the above example represents a case that can be addressed by either the Chernoff bound or the Hoeffding inequality. However, there are also cases that can be addressed only with the Hoeffding inequality but not with the Chernoff bound. An example is given below.

Example 11.14 A standardized test has an average score of 70 out of 100 across a very large population of students. A sample of 500 students is selected. Find the Hoeffding upper bound on the probability that the average score of this sample is at least 75.

Solution: We need to estimate the upper-tail probability $P(X - E[X] > \theta)$, where $E[X]$ is 70×500 and θ is $5 \times 500 = 2500$. The corresponding bound is as follows:

$$P(X - E[X] > \theta) \leq e^{-\frac{2 \cdot \theta^2}{\sum_{i=1}^n (u_i - l_i)^2}}$$

Since the test is scored between 0 and 100, l_i and u_i are set to these respective values. The sample size n is set to 500. The corresponding bound is as follows:

$$P(X - E[X] > \theta) \leq e^{-\frac{2 \cdot 2500^2}{\sum_{i=1}^{500} 100^2}}$$

The above bound simplifies to $e^{-2.5}$, which is 0.082. ■

Note that the bound of the aforementioned example could not have been done exactly with the normal distribution since the standard deviation is not available. Nevertheless, since the standard deviation is bounded above by 50, one can still use this fact to provide an excellent bound.

Example 11.15 Repeat Example 11.14 using a normal approximation of the random variable corresponding to the sample mean to provide an estimate of the upper bound on the probability that a sample of 500 students will score more than 75. As before, you can assume that the standardized test has an average of 70 out of 100. The key step is to assume an upper bound on the standard deviation of the scores.

Solution: The standard deviation of the scores is guaranteed to be at most 50, since all scores lie in $[0, 100]$. Therefore, the mean of a sample of 500 scores has a standard deviation of at most $50/\sqrt{500} = \sqrt{5}$. One can approximate this mean to be drawn from a normal distribution with a mean of 70 and standard deviation of at most $\sqrt{5}$ using the central limit theorem. Therefore, the Z-value of 75 is at least $(75 - 70)/\sqrt{5} = \sqrt{5}$. The fraction of the standard normal distribution that lies above $\sqrt{5}$ is approximately 0.0127. Therefore, the probability of the score being at least 75 is at most 0.0127. ■

It is noteworthy that one obtains a much tighter estimate using the normal distribution as compared to the Hoeffding inequality. This result is in spite of the fact that the standard deviation was grossly overestimated. In general, the normal distribution provides much better estimates than either the Chernoff bound or the Hoeffding inequality. Even though the normal estimates are formally not considered upper bounds because of the central limit approximation, modest sample sizes like 500 do result in a distribution that is very close to the normal distribution. Therefore, unless one needs to show a theoretical worst-case result,

it is preferable to use the normal approximation for establishing tail estimates. This point is discussed in greater detail in the next section.

Example 11.16 Alice and Bob will play a series of 100 bullet chess games. Based on the Elo rating system, Alice is expected to win each game with probability 0.3 and Bob is expected to win each game with probability 0.2. Otherwise, the game will be drawn. Clearly, Alice is the favorite to win the match. Use the Hoeffding inequality to find an upper bound on the probability that Bob will win the match. Comment on the informativeness of the result.

Solution: Define the random variable X_i for the i th match, which is defined as follows:

$$X_i = \begin{cases} 1 & \text{if Bob wins the } i\text{th game} \\ 0 & \text{if the } i\text{th game is drawn} \\ -1 & \text{if Alice wins the } i\text{th game} \end{cases}$$

Each X_i lies in $[-1, 1]$, which is consistent with the pre-conditions of the Hoeffding inequality. The expected value $E[X_i]$ is $(0.2 - 0.3) = -0.1$. We define the random variable X as follows:

$$X = \sum_{i=1}^{100} X_i$$

The value of $E[X]$ is, therefore, $100 * (-0.1) = -10$. Note that Bob wins if $X \geq 1$. Using the fact that $E[X] = -10$, we have the following:

$$P(X \geq 1) = P(X - E[X] \geq 11)$$

Note that the right-hand side of the above expression is in the form of the upper-tail Hoeffding inequality in which $\theta = 11$, $n = 100$, $l = -1$, and $u = +1$. Using the Hoeffding inequality, we get the following:

$$P(X - E[X] > 11) \leq e^{-\frac{2 \cdot 11^2}{100 \cdot 2^2}} = e^{-0.605} = 0.54$$

The Hoeffding inequality is not very informative in this case because it is providing an upper-bound of 0.54. On the other hand, since Alice is the favored candidate (with the higher probability of winning in each game), we already know that Bob's probability of winning is less than 0.5. Therefore, an upper-bound of 0.54 does not add to this information. ■

11.5 Tail Inequalities Versus Approximation Estimates

The tails of the sums of random variables can be either bounded (using the various inequalities) or they can be estimated by approximating the random variable to belong to a particular type of distribution such as the normal or the Poisson distribution. The different tail inequalities/estimates may apply to scenarios of different generality, and may also have different levels of strength. These different scenarios are presented in Table 11.1.

The normal distribution tails generally provide much tighter estimates than the tail inequalities (under the assumption that the corresponding preconditions are satisfied). How-

Table 11.1: Comparison of different methods used to bound or estimate tail probabilities

Result	Scenario	Bound/ Estimate?	Strength
Chebychev	Any random variable	Bound	Weak
Markov	Nonnegative random variable	Bound	Weak
Hoeffding	Sum of independent bounded random variables	Bound	Strong (Exponentially reduces with samples)
Chernoff	Sum of i.i.d. Bernoulli random variables	Bound	Strong (Exponentially reduces with samples)
CLT	Sum of many i.i.d. variables	Estimate	Almost exact
Generalized CLT	Sum of many independent and bounded variables	Estimate	Almost exact
Poisson Approx.	Sum of n Bernoulli variables with $n > 100$ and $n \cdot p < 10$	Estimate	Almost Exact

ever, the two are not comparable because the tail inequalities work under more general assumptions than the normal distribution. Furthermore, they provide worst-case bounds, whereas the normal distribution provides a possibly inaccurate estimate, because no distribution in the real world is truly normal (and therefore the precondition of a normal distribution is rarely satisfied). The Markov and Chebychev inequalities are extremely weak because they work under very general assumptions (in which the preconditions are easy to satisfy).

An interesting observation is that the Hoeffding tail bounds decay exponentially with θ^2 , which is exactly how the normal distribution behaves. This is not very surprising, because the sum of a large number of independent bounded random variables converges to the normal distribution according to the *Central Limit Theorem (CLT)*. Such a convergence is useful, because the bounds provided by an exact distribution (or a close approximation) are much tighter than any of the aforementioned tail inequalities. One problem with the use of the central limit theorem is that it does not accurately apply when the number n is small. In these cases, bounds such as the Chernoff bound and Hoeffding inequality continue to provide exact estimates in terms of *guaranteed worst-case behavior*. On the other hand, the central-limit theorem provides an *expected* estimate, assuming that the central limit theorem holds. When the central limit theorem does not hold, the quality of the approximation can be poor.

Another issue with using the central-limit theorem is that the standard deviation of the sum needs to be estimated in a data-driven manner. This type of data-driven approach necessitates the use of the t -distribution rather than the normal distribution. The t -distribution has thicker tails than the normal distribution. In general, the following rules of thumb apply when selecting between the use of the normal distribution/ t -distribution with the use of tail inequalities:

1. The tail inequalities have specific requirements in terms of the random variables that are added together. When the random variables do not satisfy these requirements, it is necessary to use the normal distribution or the t -distribution for bounding the tails.
2. It is preferable to use tail inequalities when one is looking for guaranteed worst-case behavior rather than average-case behavior. On the other hand, the central limit theorem is preferable when one is looking for an average-case approximation.
3. The approximation of the central limit theorem may be weak when one has fewer samples than 30 (although the precise quality of the approximation depends on the

type of random variables being added). In such cases, the tail inequalities have the advantage of providing guaranteed worst-case behavior. On the other hand, the worst-case bounds may not be very tight when the number of samples is small. A possible approach in this case is to use both methods to obtain multiple estimates of the tail probabilities.

In most practical settings, the central limit assumption can be used as a rule of thumb to provide a practical estimate on the probability of extreme values. The estimate based on the normal approximation is tighter and provides a much more accurate picture because it provides an *empirical estimate* on average-case behavior. In some cases, this empirical estimate is also an empirical bound when the Z-value is itself computed as a lower bound. By **empirical bound**, we refer to a bound that is not guaranteed but usually holds in most empirical settings. For example, if the Z-value is calculated by using an upper bound on the standard deviation, it creates a lower bound on the Z-value. This bounded Z-value can be used to create an empirical bound. In order to emphasize this point, we prove a normal distribution-based alternative to the Hoeffding bound. This alternative makes the additional assumption corresponding to the central limit theorem (CLT) and is therefore an empirical bound rather than worst-case bound.

Example 11.17 (CLT-Based Alternative to Hoeffding Bounds) Consider a random variable X that is obtained by averaging n independent instantiations of a bounded random variable in $[l, u]$. Show that the upper-tail probability that X is greater than $E[X]$ by $\theta > 0$ is estimated by the following empirical bound using the CLT:

$$P(X - E[X] > \theta) \leq 1 - F_Z\left(\frac{2\sqrt{n}\theta}{u-l}\right)$$

Here, $F_Z(\cdot)$ is the cumulative distribution function of the standard normal distribution. Compare this result to the Hoeffding bound and comment on the relative tightness of the two bounds.

Solution: As shown in Example 3.21 of Chapter 3, the standard deviation of the random variable X drawn from $[l, u]$ is given by **at most** $(u-l)/2$. Averaging n independent instantiations of the random variable yields a standard deviation of at most $(u-l)/(2\sqrt{n})$. Therefore, a lower bound on the Z-value of an averaged quantity that is θ more than the expected value of X is given by the following:

$$z_> = \frac{2\sqrt{n}\theta}{u-l}$$

We have used the notation $z_>$ to indicate that it is an lower bound on the Z-value. The quantity above is a lower bound because the standard deviation of $(u-l)/(2\sqrt{n})$ is an upper bound. Therefore, the upper-tail probability is given by at most $1 - F_Z(z_>)$, assuming that the central limit holds. Here, $F_Z(\cdot)$ is the cumulative distribution of the standard normal distribution. This completes the CLT portion of the example. On the other hand, the average-centric Hoeffding upper-tail bound can be shown to be the following for $X = \sum_{i=1}^n X_i/n$:

$$P(X - E[X] > \theta) \leq \exp\left(-\frac{2n\theta^2}{(u-l)^2}\right) = \exp\left[0.5 \times \left(\frac{2\sqrt{n}\theta}{u-l}\right)^2\right]$$

Therefore, one can express the Hoeffding bound in terms of $z_>$ as follows:

$$P(X - E[X] > \theta) \leq \exp(-z_>^2/2)$$

Since both the empirical CLT-bounds and the Hoeffding bound are expressed in terms of $z_>$, they can be compared as follows:

$z_>$	1	1.96	2.58	3	3.5
CLT Probability	0.159	0.025	0.005	0.001	$\ll 0.001$
Hoeffding Probability	0.606	0.146	0.036	0.011	0.002

The above table shows several values of $z_>$ and compares CLT probabilities with Hoeffding probabilities. In each case, it is evident that the normal approximation gives much tighter (and informative) probabilistic estimations (i.e., lower tail probabilities). This is because the Hoeffding probabilities are guaranteed, whereas the CLT-based bounds are empirical. One can envisage distributions of X , where the CLT does not hold. For example, if X is a Bernoulli random variable with a small value of the success probability $p = 10^{-5}$, and the value of $n = 1000$ is significantly less than $\max\{1/p, 1/(1-p)\} = 10000$, one will not obtain a bell-shaped distribution. For example, the binomial distributions of Figure 11.2(a) and 11.2(c) are visibly asymmetric, even though n is greater than $\max\{1/p, 1/(1-p)\}$. This is because of the asymmetry of the base distribution in these cases, where p is very different from 0.5. Therefore, the CLT-based approximation should be used with caution when dealing with highly asymmetric distributions for the base random variable X being aggregated. ■

11.6 Summary

This chapter introduces several probabilistic inequalities that are used for either bounding expected values of functions of random variables or for bounding the tails of probability distributions. Jensen's inequality is used for bounding expected values of functions with concavity/convexity properties, whereas tail inequalities are used for creating upper and lower bounds on the probabilities in the tails of different types of distributions. The tightness of these inequalities heavily depends on the assumptions underlying the random variables. Examples of tail inequalities include the Markov inequality, the Chebychev inequality, the Hoeffding inequality, and the Chernoff bound. The Markov inequality is a relatively weak inequality (which is developed under general assumptions) but it serves as the basis for all the other inequalities in this chapter. The Markov inequality can be tightened very significantly when specific characteristics of the underlying random variable are known. For example, when the random variable is constructed by adding a very large number of i.i.d. random variables of a bounded nature, it is possible to tighten the Markov inequality into the Chernoff or Hoeffding bound. In such cases, the central limit theorem also applies, which provides an excellent estimate of the tail probability. In many practical settings, the central limit theorem tends to provide much more useful estimations than tail probabilities.

11.7 Further Reading

A visual explanation of Jensen's inequality is provided in [53]. The classical inequalities (e.g., Markov, Chebychev, Chernoff, and Hoeffding) are widely used in probability and statistics for bounding the accuracy of aggregation-based statistics. A detailed discussion of these different methods may be found in [50]. A generalization of the Hoeffding's inequality is the McDiarmid's inequality [46], which can be applied to a more general function of the different values of X_i (beyond a linearly separable sum). The main restriction on this function is that if the i th argument of the function (i.e., the value of X_i) is changed to any other value, the function cannot change by more than c_i .

11.8 Exercises

1. Consider the non-negative random variable X . The function $F(x)$ is defined as $F(x) = x^3 + 4x^2 - 2x - 3$. Show that $E[F(X)] \geq F(E[X])$.
2. The heavy-weight boxing category has a minimum weight threshold of 95 Kg. The average weight of boxers in this category is 102 Kg. Muhammad Ali had a weight of 107 Kg. Find an upper-bound on the fraction of boxers with weight greater than that of Muhammad Ali using (a) the Markov inequality, and (b) a tightened version of the Markov inequality.
3. Mike Powell holds the long-jump world record today at 8.95 meters. The world record in 1900 was 7.50 meters, and was held by Myer Prinstein. The average long jump of (not necessarily winning) athletes at recent major events has been 8.15 meters. Use a modified version of the Markov inequality to find an upper bound on the fraction of athletes at recent major events who fail to beat the 1900 world record.
4. **[Upper-Tail Chernoff Bound]:** The chapter provides a proof sketch of the upper-tail Chernoff bound, but not the full proof. Work out the full proof of bound on the upper tail using the lower-tail proof as a guide. Where do you use the fact that $\delta < 2 \cdot e - 1$?
5. Suppose you flip a fair coin 100 times. Determine the mean and standard deviation of the random variable representing the number of tails. Provide a bound on the probability that you obtain more than 90 tails with the use of the (i) Markov Inequality (ii) Chebychev Inequality (iii) Chernoff Upper Tail Bound, (iv) Chernoff Lower Tail Bound and (v) Hoeffding Inequality. **[Hint:** Either the upper-tail or lower-tail Chernoff bound can be used, depending on which random variable you look at.]
6. Repeat Exercise 5 for a biased coin with tails probability of 8/9.
7. Use the central limit theorem to model the number of tails in Exercises 5 and 6 by normal distributions. Use the cumulative normal distribution to approximate the probabilities that the number of tails is more than 90 in each case.
8. Let Z and Z' be independent random variables satisfying $E[Z] = E[Z'] = 0$, and $Z, Z' \in [a, b]$. Then, show the following steps that lead to a complete proof of the Hoeffding inequality:
 - (a) Show using Jensen's inequality that $E[e^{-tZ'}] \geq 1$ for any real value of t .

- (b) Use the result in the previous part along with the independence of Z and Z' to show the following for any real value of t :

$$E[e^{t(Z-Z')}] \geq E[e^{tZ}]$$

- (c) The Taylor's expansion of the exponential function implies the following:

$$E[e^{t(Z-Z')}] = \sum_{i=0}^{\infty} E\left[\frac{[t(Z-Z')]^i}{i!}\right]$$

Argue why all odd powers of $(Z - Z')$ will disappear in the above summation.

- (d) Use parts (b) and (c) to prove the following for any real value of t :

$$E[e^{tZ}] \leq 1 + \frac{t^2(b-a)^2}{2^2 2!} + \dots + \frac{t^{2k}(b-a)^{2k}}{2^{2k} 2k!} + \dots$$

You may find the result of Problem 3.23 helpful in constructing the proof.

- (e) use the result of part (d) to show that $E[e^{tZ}] \leq e^{t^2 \cdot (b-a)^2 / 8}$.
(f) Use the aforementioned result to complete the proof of the Hoeffding inequality.

- 9.** Let X be a random variable distributed in $[-a, +a]$ with a mean of 0. Show the following for any $f \in (0, 1)$:

$$\begin{aligned} P(X > f \cdot a) &\leq \frac{1}{1+f} \\ P(X < -f \cdot a) &\leq \frac{1}{1+f} \end{aligned}$$

- 10.** Suppose that you toss a fair die 600 times. Use the normal approximation to the binomial distribution to determine an estimate of the probability that the number of sixes is either less than 80 or more than 120.

- 11.** Suppose that you toss a biased coin with heads probability 0.1 a total of 100 times.

- (a) Compute the exact probability that five or less heads will be obtained using the binomial distribution.
(b) Use the Chernoff bound to compute the probability that five or less heads will be obtained.
(c) Now use the normal approximation of the binomial distribution to estimate this probability.
(d) Now use the Poisson approximation to the binomial distribution to estimate this probability. You may use any online calculator of your choice to compute the upper-incomplete Gamma function.

- 12.** Consider the scenario of Example 11.16. Use the CLT-based approach (instead of the Hoeffding inequality used in Example 11.16) to estimate the probability that Bob wins the match. Compare this probability to the result you got in Example 11.16 using the Hoeffding bound and explain why you get a much lower probability with CLT.

- 13.** All lengths of a particular type of fish of a particular age are known to lie between 2 feet and 2.1 feet. The mean is 2.05 feet. Use the generalized Markov inequality to calculate an upper bound on the probability that the length is less than 2.01 feet.
- 14.** Let MAD_X represent the mean absolute deviation of random variable X . Show the following alternative to the Chebychev inequality:

$$P(|X - E[X]| > \alpha) \leq MAD_X/\alpha$$

Suppose that X is a random variable with MAD_X of $\sqrt{2/\pi}$ and variance of 1. For what value of α does this new inequality provide a tighter bound?

- 15.** Exercise 14 provides a generalization of the Chebychev inequality to an inequality that is based on the mean absolute deviation. The mean absolute deviation can be viewed as a variation on central moments based on absolute values. Propose a variation of the Chebychev inequality that is based on a variation of the n th central moment.
- 16.** [Generalized Markov Inequality]: The translated and reflected Markov inequalities provide examples of how specific *nonnegative monotonic functions* of the original random variable can be used to provide bounds on the tail, as long as the expected value of that function is known. This exercise generalizes this principle. Let $g(\cdot)$ be a monotonically increasing and nonnegative function. Then, show the following upper-tail inequality for random variable X and any α :

$$P(X > \alpha) \leq \frac{E[g(X)]}{g(\alpha)}$$

What kind of tail inequality can you derive for a monotonically decreasing function $h(\cdot)$?

- 17.** Let X be a continuous random variable satisfying $E[X] = 3$ for which the probability density is nonzero only for values greater than 1. Show that the expected value of $X^3 - X^2$ is at least 18. [Hint: In what range is $X^3 - X^2$ convex?]
- 18.** Consider a nonnegative random variable X with a mean of 1, variance of 1, third-moment of 5, and fourth moment of 20. Show that the probability $P(X > \alpha)$ may be bounded as follows:

$$P(X > \alpha) \leq \begin{cases} 1/\alpha & \text{if } 0 < \alpha \leq 2 \\ 2/\alpha^2 & \text{if } 2 < \alpha \leq 2.5 \\ 5/\alpha^3 & \text{if } 2.5 < \alpha \leq 4 \\ 20/\alpha^4 & \text{if } \alpha > 4 \end{cases}$$

- 19.** For any positive random variable, show that the expected value of its n th moment is at least equal to $E[X]^n$.
- 20.** Corollary 11.5 shows that the tails of the binomial distribution can be bounded using the Hoeffding inequality in terms of the number of trials n , success parameter p , and offset δ . The Chernoff bound is the most direct way to do it. Create a plot of the upper-tail probability using both the Hoeffding inequality and the Chernoff bound against the value of n on the X-axis. The values of p and δ are fixed to 0.1 and 2, respectively. Repeat this process of generating a plot of the upper-tail probabilities by fixing n to 100 and δ to 2, while varying p from 0 to 0.5 on the X-axis. Make a note of the cases in which each bound provides tighter results.

- 21.** Use the Hoeffding inequality to bound the probability that the sum of n independent rolls of the die is at least $4n$.
- 22.** The proofs of the Hoeffding inequality and the Chernoff bound use the MGF of the random variable to be bounded. We repeat Exercise 21 using more specific information about the random variable X corresponding to the sum of n die throws:
- Derive the MGF $f_X^T(s) = E[\exp(sX)]$ of the random variable X .
 - Combine the MGF of X with the Markov inequality to find an algebraic expression in s that bounds the probability that the sum of n die throws is at least $4n$. Argue why s needs to be greater than 0 for the Markov inequality to hold.
 - Find the value of s at which the bound is tightest using differential calculus. You may use Desmos or any graphing calculator to solve any intermediate equations. Show that $P(X > 4n) < 0.958^n$.

You will find that the resulting bound is tighter than that obtained using the Hoeffding inequality in Exercise 21. Why is this the case?

- 23.** Let X be the sum of n uniform random variables in $(0, 1)$. Use the Hoeffding inequality to find the upper-tail bound that X is at least $0.51n$. Use the same approach to find the lower-tail bound that X is at most $0.49n$.
- 24.** This exercise again examines the power of the MGF in developing tail bounds. Let X be the sum of n i.i.d. uniform random variables in $(0, 1)$ as in Exercise 23. What is the MGF of X ? Use Markov's inequality to develop both an upper-tail and lower-tail bound expression in s that X is at least $0.01n$ away from its expected value. What is the appropriate range of s for each bound. Show that the tightest possibly upper- and lower-tail bounds for each of $P(X > 0.51n)$ and $P(X < 0.49n)$ are both 0.9994^n . You may use Desmos or any other graphing calculator to minimize the expression(s) in s for the probabilistic bound(s). Compare your results with the bounds obtained with the use of the Hoeffding inequality in the previous exercise.
- 25.** Let X be the sum of n i.i.d. exponential variables with parameter 1 and n i.i.d. uniform variables in $[0, 1]$. Use the MGF method of Exercise 22 to bound the probabilities $P(X > 1.6n)$ and $P(X < 1.4n)$ as functions of n . Feel free to use Desmos or any graphing calculator for function minimization. Note that the Hoeffding inequality cannot be used because of the unboundedness of the exponential variable.
- 26.** Consider an Erlang distribution created as the sum of n i.i.d. exponential distributions with arrival rate parameter $\lambda = 1$. Use the MGF method of Exercise 22 to bound the probabilities $P(X > 1.2n)$ and $P(X < 0.8n)$ as functions of n . Feel free to use Desmos or any graphing calculator for function minimization.
- 27.** Use Jensen's inequality to show that (i) the arithmetic mean of a set of positive values x_1, \dots, x_n is at least equal to the geometric mean, and (ii) their geometric mean is at least equal to their harmonic mean.
- 28.** A pathological gambler has a convex utility function $U = f(X)$ for payoff X . The gambler is given two choices. In the first choice, they get paid \$1 with no questions asked. In the second choice, they get paid \$2 for a heads outcome of a fair coin and nothing otherwise. Argue why the gambler's utility is always maximized by selecting the second choice.

References

- [1] C. Aggarwal. Data mining: The textbook. *Springer*, 2015.
- [2] C. Aggarwal. Recommender systems: The textbook. *Springer*, 2016.
- [3] C. Aggarwal. Machine learning for text. *Springer*, 2022.
- [4] C. Aggarwal. Neural networks and deep learning. *Springer*, 2023.
- [5] C. Aggarwal. Outlier analysis. *Springer*, 2017.
- [6] C. Aggarwal. Linear algebra and optimization for machine learning. *Springer*, 2020.
- [7] C. Aggarwal and C. Reddy. Data clustering: Algorithms and Applications. *CRC Press*, 2013.
- [8] A. Azran. The rendezvous algorithm: Multiclass semi-supervised learning with markov random walks. *ICML*, pp. 49–56, 2007.
- [9] D. Bertsekas and J. Tsitsiklis. Introduction to probability. *Athena Scientific*, 2008.
- [10] C. M. Bishop. Pattern recognition and machine learning. *Springer*, 2007.
- [11] S. Bhagat, G. Cormode, and S. Muthukrishnan. Node classification in social networks. *Social Network Data Analytics*, Springer, pp. 115–148. 2011.
- [12] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, pp. 993–1022, 2003.
- [13] J. Blitzstein and J. Hwang. Introduction to probability. *CRC Press*, 2015.
- [14] W. Bolstad and J. Curran. Introduction to Bayesian statistics. *John Wiley and Sons*, 2016.
- [15] S. Brin, and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1–7), pp. 107–117, 1998.
- [16] G. Casella and R. Berger. Statistical inference. *Cengage Learning*, 2021.
- [17] C. O’Cinneide. The mean is within one standard deviation of any median. *The American Statistician*, 44(4), pp. 292–293, 1990.
- [18] M. Deisenroth, A. Faisal, and C. Ong. Mathematics for machine learning. *Cambridge University Press*, 2020.

- [19] A. Drake. Fundamentals of applied probability theory. *McGraw Hill*, 1967.
- [20] N. Draper and H. Smith. Applied regression analysis. *John Wiley and Sons*, 1998.
- [21] D. Fink. A compendium of conjugate priors. *Unpublished Manuscript*, 1997.
- [22] R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7: pp. 179–188, 1936.
- [23] R. Fisher. Statistical methods for research workers. In *Breakthroughs in statistics*, Springer, pp. 66–70, 1992.
- [24] G. D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3), pp. 268–278, 1973.
- [25] J. Fox. Applied regression analysis and generalized linear models. *Sage Publications*, 2015.
- [26] J. Frost. Hypothesis testing, *Statistics by Jim Publishing*, 2020.
- [27] A Gelman, J. Carlin, H. Stern, D. Dunson, A. Vehtari, and D. Rubin. Bayesian data analysis, 3rd edition. *Chapman and Hall/CRC Press*, 2013.
- [28] L. Hannah, D. Blei, and W. Powell. Dirichlet process mixtures of generalized linear models. *Journal of Machine Learning Research*, 12(6), 2011.
- [29] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6), pp. 607–616, 1996.
- [30] H. Hotelling and L. Solomons. The limits of a measure of skewness. *The Annals of Mathematical Statistics*, 3, pp. 141–142, 1932.
- [31] G. James, D. Witten, T. Hastie, and R. Tibshirani. An introduction to statistical learning, *Springer*, 2013.
- [32] J. Han, M. Kamber, and J. Pei. Data mining: concepts and techniques. *Morgan Kaufmann*, 2011.
- [33] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning. *Springer*, 2009.
- [34] T. Hastie and R. Tibshirani. Generalized additive models. *CRC Press*, 2017.
- [35] D. Hawkins. Identification of outliers. *Chapman and Hall*, 1980.
- [36] T. Hofmann. Probabilistic latent semantic indexing. *ACM SIGIR Conference*, pp. 50–57, 1999.
- [37] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 41(1–2), pp. 177–196, 2001.
- [38] R. Hogg, J. McKean, and A. Craig. Introduction to mathematical statistics. *Pearson Education*, 2005.
- [39] G. James, D. Witten, T. Hastie, and R. Tibshirani. An introduction to statistical learning. *Springer*, 2013.
- [40] C. Johnson. Logistic matrix factorization for implicit feedback data. *NeurIPS*, 2014.
- [41] I. Jolliffe. Principal components in regression analysis. Principal component analysis. *Springer*, New York, NY, 1986.
- [42] R. Johnson, I. Miller, and J. Freund. Probability and statistics for engineers. *Pearson Education*, 2000.

- [43] A. Krogh, M. Brown, I. Mian, K. Sjolander, and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *Journal of molecular biology*, 235(5), pp. 1501–1531, 1994.
- [44] C. Manning, P. Raghavan, and H. Schutze. Introduction to information retrieval. *Cambridge University Press*, Cambridge, 2008.
- [45] P. McCullagh and J. Nelder. Generalized linear models. *CRC Press*, 2019.
- [46] C. McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics*, pp. 148–188, *Cambridge University Press*, Cambridge, 1989.
- [47] R. McElreath. Statistical rethinking: A Bayesian course with examples in R and Stan. *Chapman and Hall/CRC Press*, 2020.
- [48] R. McGill, J. Tukey, and W. Larsen. Variations of box plots. *The American Statistician*, 32(1), pp. 12–16, 1978.
- [49] G. McLachlan, and K. Thriyambakam. The EM algorithm and extensions. *John Wiley and Sons*, 2007.
- [50] R. Motwani and P. Raghavan. Randomized algorithms. *Cambridge University Press*, 1995.
- [51] K. Murphy. Machine learning: A probabilistic perspective. *MIT Press*, 2012.
- [52] T. J. Myung. Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47(1), pp. 90–100, 2003.
- [53] T. Needham. A visual explanation of Jensen’s inequality. *American Mathematical Monthly*, 100(8), pp. 768–771, 1993.
- [54] N. Privault. Understanding Markov chains. Examples and Applications. *Springer*, 2018.
- [55] P. Rousseeuw and A. Leroy. Robust regression and outlier detection. *John Wiley and Sons*, 2015.
- [56] H. Scheffe. The analysis of variance. *John Wiley and Sons*, 1999.
- [57] B. W. Silverman. Density estimation for statistics and data analysis. *Routledge*, 2018.
- [58] M. D. Springer. The algebra of random variables. *Wiley*, 1979.
- [59] G. Strang. An introduction to linear algebra. *Wellesley-Cambridge Press*, 2016.
- [60] E. Tufte. The visual display of quantitative information. *Graphics Press*, 2001.
- [61] M. Waskom. Seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60), 2021.
- [62] R. Wilcox. Introduction to robust estimation and hypothesis testing. *Academic Press*, 2011.
- [63] R. Witte and J. Witte. Statistics. *Wiley*, 2016.
- [64] T. Zhang, R. Ramakrishnan, and M. Livny. Fast density estimation using cf-kernel for very large databases. *ACM KDD Conference*, 1999.
- [65] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. *ICML Conference*, pp. 912–919, 2003.
- [66] <https://archive.ics.uci.edu/ml/index.php>

Index

Symbols

- R^2 -statistic, 335
 χ^2 -distribution, 170
 χ^2 -independence test, 222
 k -means algorithm, 401, 402
 k -step transition matrix, 442, 450
 p -value, 197
 s -transform, 182

A

- Additive rule for mutually exclusive events, 75

Alternate hypothesis, 196

Analysis of variance (ANOVA), 224

Aperiodic Markov chain, 447

Attribute, 3

B

Bar chart, 54

Baum-Welch algorithm, 477

Bayesian statistics, 83, 252

Bayes rule, 80

Bayes rule for conditional distributions, 94

Bernoulli distribution, 131

Bessel correction, 32

Beta distribution, 292

Beta function, 292

Bimodal, 30

Binary classification, 18

Binomial distribution, 138

Box plot, 49

C

- Categorical data, 4
Categorical distribution, 132, 141
Categorical predictors in regression, 324
Central limit theorem, 193, 508
Central tendency measures, 6, 26
Chebychev inequality, 492
Chernoff bound (lower tail), 496
Chernoff bound (upper tail), 497
Chi-squared distribution, 170
Classification, 17, 353
Clustering, 16
Cochran's theorem, 171
Collaborative filtering, 412
Collective classification, 466
Complementary events, 75
Compound distributions, 108
Concave function, 487
Conditional independence, 79
Confidence, 198
Confidence intervals, 16, 155, 191, 200
Conjugate prior distribution, 280
Contingency table, 45
Continuity correction factor, 501
Continuous numeric data, 4
Convex function, 486
Convolution operator, 117
Correlation, 38
Countably infinite sample space, 135
Covariance, 34, 104

- Covariance matrix, 42
 Critical test statistic values, 199
 Cumulative distribution function, 87
 Cumulative distribution method for function of random variables, 114
- D**
 Difference in means testing, 208
 Dimension, 3
 Dirichlet distribution, 294
 Discrete numeric data, 4
 Discrete state Markov process, 435
 Discretization, 5, 362, 403
 Discriminative models, 354
 Dispersion measures, 6, 31
 Double exponential distribution, 312
 Dynamic programming in HMM, 477
- E**
 Eigenvalues, 42
 Eigenvectors, 42
 Em algorithm, 267, 395
 Empirical bound, 509
 Equal variance t -test, 213
 Ergodic Markov chain, 448
 Erlang distribution, 290
 E-Step, 267, 395
 Estimator of bar chart, 54
 Event, 68
 Event complement, 75
 Event space, 69
 Exclusive events, 68
 Exhaustive events, 68
 Expectation-maximization algorithm, 267, 395
 Expected number of hits, 442
 Explaining sequences probabilistically, 476
 Exponential distribution, 145
- F**
 F-distribution, 226
 Feature, 3
 Feature engineering, 5
 Fisher discriminant index, 233, 234
 Fliers, 50
 Forward algorithm, 476
 Forward-backward algorithm, 477
 Fourier transform of density function, 183
 Frequentist statistics, 252
- Functions of random variables, 114
 F-value, 226
- G**
 Gamma distribution, 290
 Gamma function, 166, 290, 368
 Gaussian distribution, 152
 Generalized Bernoulli distribution, 133
 Generating process, 67, 71
 Geometric distribution, 135
 Gradient descent, 321
- H**
 Hidden Markov models (HMM), 471
 Hidden variable, 414
 Histogram, 7
 HMM applications, 479
 Hoeffding inequality, 504
 Homophily, 467
 Hypothesis testing, 15, 191
- I**
 Implicit-feedback data, 362, 403
 Independence test, 222
 Independent events, 78
 Inlier, 20
 Inter-quartile range, 29
 Interactions in regression, 339
 Interpolation tie-breaking, 27
 Interquartile range, 101
 Iterative label propagation, 469
- J**
 Jensen's inequality, 486
- K**
 Katz measure, 445
 Kendall rank correlation, 39
 Kernel density estimation, 272
 Kernel trick, 348
 k-th central moment, 180
 k-th moment, 180
 k-th standardized moment, 181
 Kurtosis, 181
- L**
 Laplace distribution, 148, 312
 Laplacian smoothing, 280
 Law of large numbers, 193

- Least Absolute Shrinkage and Selection Operator (LASSO), 331
Least-squares classification, 372
Left-skewed distribution, 48
Likelihood, 82
Linear regression, 303
Line plot, 52
Log-Likelihood function, 248
Logistic matrix factorization, 419
Logistic regression, 377
Logit (Ordered), 385
Loss function, 19, 304
Lower-tail hypothesis test, 206
- M**
- Mahalanobis distance, 422
Markov chain, 436
Markov inequality, 490
Markov process, 435
Markov property, 436
Matrix factorization for outlier detection, 427
Maximum a posteriori estimation (MAP), 278
Maximum likelihood estimation (MLE), 247
Mean, 26, 98
Mean absolute deviation, 31, 103
Median, 27, 100
Memoryless property of geometric distribution, 136
Memoryless property of the exponential distribution, 146
MGF, 182
Mini-batch stochastic gradient descent, 323, 374
Minimum variance estimation, 247
Min-max normalization, 58
Mixed-attribute data type, 4
Mixing in Markov chain, 439
Mixture component, 176
Mixture models, 394
Mode, 30
Moment generating functions, 182
Moments, 102
Moments of random variables, 180
M-Step, 267, 395
Multimodal, 30
Multinomial distribution, 141
Multinomial logistic regression, 382
Multinoulli distribution, 133
Multiple regression, 304
Multivariate analysis, 3
Multivariate data, 3
Multivariate probability distributions, 89
Mutually exclusive events, 68, 75
- N**
- Naïve Bayes classifier, 357
Nonlinear regression, 339
Nonnegative matrix factorization, 416
Normal approximation to binomial distribution, 500
Null hypothesis, 196
Numeric data, 4
- O**
- Observation, 3
One-hot encoding, 5
One-sided confidence interval, 207
One-sided hypothesis test, 205
One-tailed hypothesis test, 205
Ordered logit regression, 385
Order of moment, 180
Ordinal data, 4
Ordinal regression, 385
Outcome, 19
Outlier detection, 20
Out-of-sample instance, 307
Overfitting, 326
- P**
- PageRank, 463
Paired *t*-test, 214
Pareto distribution, 300
Pearson correlation coefficient, 38
Percentile, 28, 29, 100
Personalized PageRank, 466
Plate diagrams, 368, 414
Poisson distribution, 149
Population, 12
Positive semidefinite property, 42
Power method, 464
Powers of random variables, 117
Principal component directions, 42
Probabilistic inequalities, 485
Probabilistic latent semantic analysis, 413

Probability density functions, 71, 86
 Probability mass function, 71
 Probability space, 70

Q

Quality control, 495

R

Randomized algorithms, 485
 Random variable, 70
 Receiver operating characteristic, 41
 Recommender systems, 412
 Regressand, 19
 Regression, 19, 303
 Regression coefficients, 19, 303
 Regressors, 19
 Regularization, 252, 315, 328
 Residual matrix, 428
 Response variable, 19
 Right-skewed distribution, 47
 Root-Mean-Squared Error (RMSE), 334

S

Sample mean, 26
 Sample selection bias, 14
 Sample space, 66
 Sample statistic, 195
 Sampling distribution, 194, 195
 Scatter plot, 7, 53
 Self-supervised learning, 393
 Set union rule for probabilities, 74
 Significance level, 198
 Silverman's approximation rule, 273
 Simple bar chart, 55
 Simple regression, 304
 Skew in distributions, 47
 Skewness, 181
 Spearman rank correlation, 39
 Standard deviation, 32, 102
 Standard error, 195
 Standardization, 57
 Standard normal distribution, 88
 Stationary distribution of Markov chain, 436
 Steady state distribution of Markov chain,
 436
 Stirling's approximation, 189
 Stochastic gradient descent, 323

Sum-of-squared distance, 402
 Sum distribution, 117
 Summarization of data, 25
 Supervision, 17
 Symmetric Dirichlet distribution, 295

T

Tail inequalities, 485
 Tails, 20
 Test data, 307
 Test statistic, 197
 Tikhonov regularization, 328
 Total probability rule, 80, 81
 Total probability rule: continuous hypothesis
 space, 109
 Total probability rule for conditional distri-
 butions, 94
 Training data, 307
 Transience time in Markov chains, 454
 Transition in Markov chain, 436
 Transition matrix, 437
 Translated Markov Inequality, 490
 Tukey box plot, 50
 Two-sided hypothesis test, 196
 Two-tailed hypothesis test, 196

U

Unbiased estimator, 246
 Uncountably infinite sample spaces, 135
 Unequal variance t -test, 208
 Uniform distribution, 128
 Unimodal, 30
 Univariate analysis, 3
 Univariate data, 3
 Unsupervised learning, 393
 Upper-tail hypothesis test, 206

V

Vandermonde matrix, 342
 Variable, 3
 Variance, 6, 32, 102
 Vertex classification, 466
 Visualization of data, 25
 Viterbi algorithm, 476

W

Whitening, 59