MAX A. LITTLE

Machine Learning for Signal Processing

Data Science, Algorithms, and Computational Statistics

OXFORD

Machine Learning for Signal Processing

Machine Learning for Signal Processing

Data Science, Algorithms, and Computational Statistics

Max A. Little

OXFORD
UNIVERSITY PRESS

OXFORD
UNIVERSITY PRESS

Great Clarendon Street, Oxford, OX2 6DP,
United Kingdom

Oxford University Press is a department of the University of Oxford.
It furthers the University's objective of excellence in research, scholarship,
and education by publishing worldwide. Oxford is a registered trade mark of
Oxford University Press in the UK and in certain other countries

First Edition published in 2019

Impression: 1

Published in the United States of America by Oxford University Press
198 Madison Avenue, New York, NY 10016, United States of America

British Library Cataloguing in Publication Data
Data available

Library of Congress Control Number: 2019944777

ISBN 978–0–19–871493–4

DOI: 10.1093/oso/9780198714934.001.0001

Printed and bound by
CPI Group (UK) Ltd, Croydon, CR0 4YY

Preface

Digital signal processing (DSP) is one of the 'foundational' but somewhat invisible, engineering topics of the modern world, without which, many of the technologies we take for granted: the digital telephone, digital radio, television, CD and MP3 players, WiFi, radar, to name just a few, would not be possible. A relative newcomer by comparison, statistical machine learning is the theoretical backbone of exciting technologies that are by now starting to reach a level of ubiquity, such as automatic techniques for car registration plate recognition, speech recognition, stock market prediction, defect detection on assembly lines, robot guidance and autonomous car navigation. Statistical machine learning has origins in the recent merging of classical probability and statistics with artificial intelligence, which exploits the analogy between intelligent information processing in biological brains and sophisticated statistical modelling and inference.

DSP and statistical machine learning are of such wide importance to the knowledge economy that both have undergone rapid changes and seen radical improvements in scope and applicability. Both DSP and statistical machine learning make use of key topics in applied mathematics such as probability and statistics, algebra, calculus, graphs and networks. Therefore, intimate formal links between the two subjects exist and because of this, an emerging consensus view is that DSP and statistical machine learning should not be seen as separate subjects. The many overlaps that exist between the two subjects can be exploited to produce new digital signal processing tools of surprising utility and efficiency, and wide applicability, highly suited to the contemporary world of pervasive digital sensors and high-powered and yet cheap, computing hardware. This book gives a solid mathematical foundation to the topic of statistical machine learning for signal processing, including the contemporary concepts of the *probabilistic graphical model* (PGM) and *nonparametric Bayes*, concepts which have only more recently emerged as important for solving DSP problems.

The book is aimed at advanced undergraduates or first-year PhD students as well as researchers and practitioners. It addresses the foundational mathematical concepts, illustrated with pertinent and practical examples across a range of problems in engineering and science. The aim is to enable students with an undergraduate background in mathematics, statistics or physics, from a wide range of quantitative disciplines, to get quickly up to speed with the latest techniques and concepts in this fast-moving field. The accompanying software will enable readers to test out the techniques to their own signal analysis problems. The presentation of the mathematics is much along the lines of a standard undergraduate physics or statistics textbook, free of distracting technical complexities and jargon, while not sacrificing rigour. It would be an excellent textbook for emerging courses in machine learning for signals.

Contents

Preface **v**

List of Algorithms **xiii**

List of Figures **xv**

1 Mathematical foundations **1**
1.1 Abstract algebras 1
Groups 1
Rings 3
1.2 Metrics 4
1.3 Vector spaces 5
Linear operators 7
Matrix algebra 7
Square and invertible matrices 8
Eigenvalues and eigenvectors 9
Special matrices 10
1.4 Probability and stochastic processes 12
Sample spaces, events, measures and distributions 12
Joint random variables: independence, conditionals, and marginals 14
Bayes' rule 16
Expectation, generating functions and characteristic functions 17
Empirical distribution function and sample expectations 19
Transforming random variables 20
Multivariate Gaussian and other limiting distributions 21
Stochastic processes 23
Markov chains 25
1.5 Data compression and information theory 28
The importance of the information map 31
Mutual information and Kullback-Leibler (K-L) divergence 32
1.6 Graphs 34
Special graphs 35
1.7 Convexity 36
1.8 Computational complexity 37
Complexity order classes and *big-O* notation 38

Tractable versus intractable problems: NP-completeness 38

2 Optimization 41
2.1 Preliminaries 41
Continuous differentiable problems and critical points 41
Continuous optimization under equality constraints: Lagrange multipliers 42
Inequality constraints: duality and the Karush-Kuhn-Tucker conditions 44
Convergence and convergence rates for iterative methods 45
Non-differentiable continuous problems 46
Discrete (combinatorial) optimization problems 47
2.2 Analytical methods for continuous convex problems 48
L_2-norm objective functions 49
Mixed L_2-L_1 norm objective functions 50
2.3 Numerical methods for continuous convex problems 51
Iteratively reweighted least squares (IRLS) 51
Gradient descent 53
Adapting the step sizes: line search 54
Newton's method 56
Other gradient descent methods 58
2.4 Non-differentiable continuous convex problems 59
Linear programming 59
Quadratic programming 60
Subgradient methods 60
Primal-dual interior-point methods 62
Path-following methods 64
2.5 Continuous non-convex problems 65
2.6 Heuristics for discrete (combinatorial) optimization 66
Greedy search 67
(Simple) tabu search 67
Simulated annealing 68
Random restarting 69

3 Random sampling 71
3.1 Generating (uniform) random numbers 71
3.2 Sampling from continuous distributions 72
Quantile function (inverse CDF) and inverse transform sampling 72
Random variable transformation methods 74
Rejection sampling 74
Adaptive rejection sampling (ARS) for log-concave densities 75
Special methods for particular distributions 78
3.3 Sampling from discrete distributions 79
Inverse transform sampling by sequential search 79

	Rejection sampling for discrete variables	80
	Binary search inversion for (large) finite sample spaces	81
3.4	Sampling from general multivariate distributions	81
	Ancestral sampling	82
	Gibbs sampling	83
	Metropolis-Hastings	85
	Other MCMC methods	88
4	**Statistical modelling and inference**	**93**
4.1	Statistical models	93
	Parametric versus nonparametric models	93
	Bayesian and non-Bayesian models	94
4.2	Optimal probability inferences	95
	Maximum likelihood and minimum K-L divergence	95
	Loss functions and empirical risk estimation	98
	Maximum a-posteriori and regularization	99
	Regularization, model complexity and data compression	101
	Cross-validation and regularization	105
	The bootstrap	107
4.3	Bayesian inference	108
4.4	Distributions associated with metrics and norms	110
	Least squares	111
	Least L_q-norms	111
	Covariance, weighted norms and Mahalanobis distance	112
4.5	The exponential family (EF)	115
	Maximum entropy distributions	115
	Sufficient statistics and canonical EFs	116
	Conjugate priors	118
	Prior and posterior predictive EFs	122
	Conjugate EF prior mixtures	123
4.6	Distributions defined through quantiles	124
4.7	Densities associated with piecewise linear loss functions	126
4.8	Nonparametric density estimation	129
4.9	Inference by sampling	130
	MCMC inference	130
	Assessing convergence in MCMC methods	130
5	**Probabilistic graphical models**	**133**
5.1	Statistical modelling with PGMs	133
5.2	Exploring conditional independence in PGMs	136
	Hidden versus observed variables	136
	Directed connection and separation	137
	The Markov blanket of a node	138
5.3	Inference on PGMs	139

Exact inference 140
Approximate inference 143

6 Statistical machine learning 149
6.1 Feature and kernel functions 149
6.2 Mixture modelling 150
Gibbs sampling for the mixture model 150
E-M for mixture models 152
6.3 Classification 154
Quadratic and linear discriminant analysis (QDA and LDA) 155
Logistic regression 156
Support vector machines (SVM) 158
Classification loss functions and misclassification count 161
Which classifier to choose? 161
6.4 Regression 162
Linear regression 162
Bayesian and regularized linear regression 163
Linear-in parameters regression 164
Generalized linear models (GLMs) 165
Nonparametric, nonlinear regression 167
Variable selection 169
6.5 Clustering 171
K-means and variants 171
Soft K-means, mean shift and variants 174
Semi-supervised clustering and classification 176
Choosing the number of clusters 177
Other clustering methods 178
6.6 Dimensionality reduction 178
Principal components analysis (PCA) 179
Probabilistic PCA (PPCA) 182
Nonlinear dimensionality reduction 184

7 Linear-Gaussian systems and signal processing 187
7.1 Preliminaries 187
Delta signals and related functions 187
Complex numbers, the unit root and complex exponentials 189
Marginals and conditionals of linear-Gaussian models 190
7.2 Linear, time-invariant (LTI) systems 191
Convolution and impulse response 191
The discrete-time Fourier transform (DTFT) 192
Finite-length, periodic signals: the discrete Fourier transform (DFT) 198
Continuous-time LTI systems 201
Heisenberg uncertainty 203
Gibb's phenomena 205
Transfer function analysis of discrete-time LTI systems 206

	Fast Fourier transforms (FFT)	208
7.3	LTI signal processing	212
	Rational filter design: FIR, IIR filtering	212
	Digital filter recipes	220
	Fourier filtering of very long signals	222
	Kernel regression as discrete convolution	224
7.4	Exploiting statistical stability for linear-Gaussian DSP	226
	Discrete-time Gaussian processes (GPs) and DSP	226
	Nonparametric power spectral density (PSD) estimation	231
	Parametric PSD estimation	236
	Subspace analysis: using PCA in DSP	238
7.5	The Kalman filter (KF)	242
	Junction tree algorithm (JT) for KF computations	243
	Forward filtering	244
	Backward smoothing	246
	Incomplete data likelihood	247
	Viterbi decoding	247
	Baum-Welch parameter estimation	249
	Kalman filtering as signal subspace analysis	251
7.6	Time-varying linear systems	252
	Short-time Fourier transform (STFT) and perfect reconstruction	253
	Continuous-time wavelet transforms (CWT)	255
	Discretization and the discrete wavelet transform (DWT)	257
	Wavelet design	261
	Applications of the DWT	262
8	**Discrete signals: sampling, quantization and coding**	**265**
8.1	Discrete-time sampling	266
	Bandlimited sampling	267
	Uniform bandlimited sampling: Shannon-Whittaker interpolation	267
	Generalized uniform sampling	270
8.2	Quantization	273
	Rate-distortion theory	275
	Lloyd-Max and entropy-constrained quantizer design	278
	Statistical quantization and dithering	282
	Vector quantization	286
8.3	Lossy signal compression	288
	Audio companding	288
	Linear predictive coding (LPC)	289
	Transform coding	291
8.4	Compressive sensing (CS)	293
	Sparsity and incoherence	294
	Exact reconstruction by convex optimization	295
	Compressive sensing in practice	296

9 Nonlinear and non-Gaussian signal processing 299
9.1 Running window filters 299
Maximum likelihood filters 300
Change point detection 301
9.2 Recursive filtering 302
9.3 Global nonlinear filtering 302
9.4 Hidden Markov models (HMMs) 304
Junction tree (JT) for efficient HMM computations 305
Viterbi decoding 306
Baum-Welch parameter estimation 306
Model evaluation and structured data classification 309
Viterbi parameter estimation 309
Avoiding numerical underflow in message passing 310
9.5 Homomorphic signal processing 311

10 Nonparametric Bayesian machine learning and signal processing 313
10.1 Preliminaries 313
Exchangeability and de Finetti's theorem 314
Representations of stochastic processes 316
Partitions and equivalence classes 317
10.2 Gaussian processes (GP) 318
From basis regression to kernel regression 318
Distributions over function spaces: GPs 319
Bayesian GP kernel regression 321
GP regression and Wiener filtering 325
Other GP-related topics 326
10.3 Dirichlet processes (DP) 327
The Dirichlet distribution: canonical prior for the categorical distribution 328
Defining the Dirichlet and related processes 331
Infinite mixture models (DPMMs) 334
Can DP-based models actually infer the number of components? 343

Bibliography 345

Index 353

List of Algorithms

2.1 Iteratively reweighted least squares (IRLS). 52
2.2 Gradient descent. 53
2.3 Backtracking line search. 54
2.4 Golden section search. 55
2.5 Newton's method. 56
2.6 The subgradient method. 61
2.7 A primal-dual interior-point method for linear programming (LP). 64
2.8 Greedy search for discrete optimization. 67
2.9 (Simple) tabu search. 68
2.10 Simulated annealing. 69
2.11 Random restarting. 70
3.1 Newton's method for numerical inverse transform sampling. 73
3.2 Adaptive rejection sampling (ARS) for log-concave densities. 76
3.3 Sequential search inversion (simplified version). 80
3.4 Binary (subdivision) search sampling. 81
3.5 Gibb's Markov-Chain Monte Carlo (MCMC) sampling. . . 83
3.6 Markov-Chain Monte Carlo (MCMC) Metropolis-Hastings (MH) sampling. 86
5.1 The junction tree algorithm for (semi-ring) marginalization inference of a single variable clique. 142
5.2 Iterative conditional modes (ICM). 144
5.3 *Expectation-maximization* (E-M). 146
6.1 Expectation-maximization (E-M) for general i.i.d. mixture models. 153
6.2 The K-means algorithm. 172
7.1 Recursive, Cooley-Tukey, radix-2, decimation in time, fast Fourier transform (FFT) algorithm. 210
7.2 Overlap-add FIR convolution. 224
8.1 The Lloyd-Max algorithm for fixed-rate quantizer design. 279
8.2 The K-means algorithm for fixed-rate quantizer design. . 280
8.3 Iterative variable-rate entropy-constrained quantizer design. 281
9.1 *Baum-Welch* expectation-maximization (E-M) for hidden Markov models (HMMs). 308
9.2 *Viterbi training* for hidden Markov models (HMMs). . . . 310
10.1 Gaussian process (GP) *informative vector machine* (IVM) regression. 325
10.2 Dirichlet process mixture model (DPMM) Gibbs sampler. 337

10.3 Dirichlet process means (DP-means) algorithm. 340
10.4 Maximum a-posteriori Dirichlet process mixture collapsed (MAP-DP) algorithm for conjugate exponential family distributions. 342

List of Figures

1.1 Mapping between abstract groups 2
1.2 Rectangle symmetry group 2
1.3 Group Cayley tables 3
1.4 Metric 2D circles for various distance metrics 4
1.5 Important concepts in 2D vector spaces 5
1.6 Linear operator flow diagram 7
1.7 Invertible and non-invertible square matrices 8
1.8 Diagonalizing a matrix 9
1.9 Distributions for discrete and continuous random variables 13
1.10 Empirical cumulative distribution and density functions 20
1.11 The 2D multivariate Gausian PDF 21
1.12 Markov and non-Markov chains 25
1.13 Shannon information map 28
1.14 Undirected and directed graphs 33
1.15 Convex functions 35
1.16 Convexity, smoothness, non-differentiability 37

2.1 Lagrange multipliers in constrained optimization 42
2.2 Objective functions with differentiable and non-differentiable points 46
2.3 Analytical shrinkage 50
2.4 Iteratively reweighted least squares (IRLS) 52
2.5 Gradient descent with constant and line search step sizes 54
2.6 Constant step size, backtracking and golden section search 55
2.7 Convergence of Newton's method with and without line search 57
2.8 Piecewise linear objective function (1D) 61
2.9 Convergence of the subgradient method 62
2.10 Convergence of primal-dual interior point 64
2.11 Regularization path of total variation denoising 65

3.1 Rejection sampling 75
3.2 Adaptive rejection sampling 77
3.3 Ancestral sampling 82
3.4 Gibb's sampling for the Gaussian mixture model 83
3.5 Metropolis sampling for the Gaussian mixture model 85
3.6 Slice sampling 90

4.1 Overfitting and underfitting: polynomial models 103

4.2 Minimum description length (MDL) 104
4.3 Cross-validation for model complexity selection 105
4.4 Quantile matching for distribution fitting 125
4.5 Convergence properties of Metropolis-Hastings (MH) sampling 131

5.1 Simple 5-node probabilistic graphical model (PGM) 133
5.2 Notation for repeated connectivity in graphical models 134
5.3 Some complex graphical models 135
5.4 Three basic probabilistic graphical model patterns 137
5.5 D-connectivity in graphical models 138
5.6 The Markov blanket of a node 139
5.7 Junction trees 143
5.8 Convergence of iterative conditional modes (ICM) 145

6.1 Gaussian mixture model for Gamma ray intensity stratification 151
6.2 Convergence of expectation-maximization (E-M) for the Gaussian mixture model 152
6.3 Classifier decision boundaries 160
6.4 Loss functions for classification 161
6.5 Bayesian linear regression 163
6.6 Linear-in-parameters regression 165
6.7 Logistic regression 166
6.8 Nonparametric kernel regression 167
6.9 Overfitted linear regression on random data: justifying variable selection 169
6.10 Lasso regression 170
6.11 Mean-shift clustering of human movement data 176
6.12 Semi-supervised versus unsupervised clustering 178
6.13 Dimensionality reduction 179
6.14 Principal components analysis (PCA): traffic count signals 181
6.15 Kernel principal components analysis (KPCA): human movement data 184
6.16 Locally linear embedding (LLE): chirp signals 185

7.1 The sinc function 189
7.2 Heisenberg uncertainty 204
7.3 Gibb's phenomena 206
7.4 Transfer function analysis 208
7.5 Transfer function of the simple FIR low-pass filter 214
7.6 Transfer function of the truncated ideal low-pass FIR filter 216
7.7 The bilinear transform 220
7.8 Computational complexity of long-duration FIR implementations 224
7.9 Impulse response and transfer function of discretized kernel regression 225
7.10 Periodogram power spectral density estimator 233

7.11 Various nonparametric power spectral density estimators 234
7.12 Linear predictive power spectral density estimators 236
7.13 Regularized linear prediction analysis model selection 238
7.14 Wiener filtering of human walking signals 239
7.15 Sinusoidal subspace principal components analysis (PCA) filtering 241
7.16 Subspace MUSIC frequency estimation 242
7.17 Typical 2D Kalman filter trajectories 243
7.18 Kalman filter smoothing by Tikhonov regularization 252
7.19 Short-time Fourier transform analysis 254
7.20 Time-frequency tiling: Fourier versus wavelet analysis 256
7.21 A selection of wavelet functions 261
7.22 Vanishing moments of the Daubechies wavelet 262
7.23 Discrete wavelet transform wavelet shrinkage 263

8.1 Digital MEMS sensors 265
8.2 Discrete-time sampling hardware block diagram 267
8.3 Non-uniform and uniform and bandlimited sampling 268
8.4 Shannon-Whittaker uniform sampling in the frequency domain 269
8.5 Digital Gamma ray intensity signal from a drill hole 271
8.6 Quadratic B-spline uniform sampling 272
8.7 Rate-distortion curves for an i.i.d. Gaussian source 277
8.8 Lloyd-Max scalar quantization 280
8.9 Entropy-constrained scalar quantization 282
8.10 Shannon-Whittaker reconstruction of density functions 283
8.11 Quantization and dithering 285
8.12 Scalar vesus vector quantization 286
8.13 Companding quantization 287
8.14 Linear predictive coding for data compression 289
8.15 Transform coding: bit count versus coefficient variance 291
8.16 Transform coding of musical audio signals 292
8.17 Compressive sensing: discrete cosine transform basis 295
8.18 Random demodulation compressive sensing 297

9.1 Quantile running filter for exoplanetary light curves 301
9.2 First-order log-normal recursive filter: daily rainfall signals 302
9.3 Total variation denoising (TVD) of power spectral density time series 303
9.4 Bilateral filtering of human walking data 304
9.5 Hidden Markov modelling (HMM) of power spectral time series data 306
9.6 Cepstral analysis of voice signals 312

10.1 Infinite exchangeability in probabilistic graphical model form 315
10.2 Linking kernel and regularized basis function regression 319

10.3 Random function draws from the linear basis regression model 321
10.4 Gaussian process (GP) function draws with various covariance functions 322
10.5 Bayesian GP regression in action 323
10.6 Gaussian process (GP) regression for classification 326
10.7 Draws from the Dirichlet distribution for $K = 3$ 327
10.8 The Dirichlet process (DP) 331
10.9 Random draws from a Dirichlet process (DP) 332
10.10 Comparing partition sizes of Dirichlet versus Pitman-Yor processes 334
10.11 Parametric versus nonparametric estimation of linear predictive coding (LPC) coefficients 339
10.12 DP-means for signal noise reduction: exoplanet light curves 341
10.13 Probabilistic graphical model (PGM) for the Bayesian mixture model 342
10.14 Performance of MAP-DP versus DP-means nonparametric clustering 344

Mathematical foundations

1

Statistical machine learning and signal processing are topics in applied mathematics, which are based upon many abstract mathematical concepts. Defining these concepts clearly is the most important first step in this book. The purpose of this chapter is to introduce these foundational mathematical concepts. It also justifies the statement that much of the art of statistical machine learning as applied to signal processing, lies in the choice of convenient mathematical models that happen to be useful in practice. Convenient in this context means that the algebraic consequences of the choice of mathematical modeling assumptions are in some sense manageable. The seeds of this manageability are the elementary mathematical concepts upon which the subject is built.

1.1 Abstract algebras

We will take the simple view in this book that mathematics is based on logic applied to *sets*: a set is an unordered collection of objects, often *real numbers* such as the set $\{\pi, 1, e\}$ (which has three *elements*), or the set of all real numbers $\mathbb{R}$ (with an infinite number of elements). From this modest origin it is a remarkable fact that we can build the entirety of the mathematical methods we need. We first start by reviewing some elementary principles of (abstract) *algebras*.

Groups

An *algebra* is a structure that defines the rules of what happens when pairs of elements of a set are acted upon by *operations*. A kind of algebra known as a *group* $(+, \mathbb{R})$ is the usual notion of addition with pairs of real numbers. It is a group because it has an *identity*, the number zero (when zero is added to any number it remains unchanged, i.e.$a+0 = 0+a = a$, and every element in the set has an *inverse* (for any number a, there is an inverse $-a$ which means that $a+(-a) = 0$. Finally, the operation is *associative*, which is to say that when operating on three or more numbers, addition does not depend on the order in which the numbers are added (i.e. $a + (b + c) = (a + b) + c$). Addition also has the intuitive property that $a + b = b + a$, i.e. it does not matter if the numbers are swapped: the operator is called *commutative*, and the group is then called an *Abelian group*. Mirroring addition is multiplication acting on the set of real numbers with zero removed $(\times, \mathbb{R} - \{0\})$, which is also an Abelian group. The identity element is 1, and the inverses

Machine Learning for Signal Processing: Data Science, Algorithms, and Computational Statistics. Max A. Little.
 DOI: 10.1093/oso/9780198714934.001.0001

are the reciprocals of each number. Multiplication is also associative, and commutative. Note that we cannot include zero because this would require the inclusion of the inverse of zero $1/0$, which does not exist (Figure 1.1).

Groups are naturally associated with *symmetries*. For example, the set of rigid geometric transformations of a rectangle that leave the rectangle unchanged in the same position, form a group together with compositions of these transformations (there are flips along the horizontal and vertical midlines, one clockwise rotation through 180° about the centre, and the identity transformation that does nothing). This group can be denoted as $V_4 = (\circ, \{e, h, v, r\})$, where e is the identity, h the horizontal flip, v the vertical flip, and r the rotation, with the composition operation $\circ$. For the rectangle, we can see that $h \circ v = r$, i.e. a horizontal followed by a vertical flip corresponds to a 180° rotation (Figure 1.2).

Very often, the fact that we are able to make some convenient algebraic calculations in statistical machine learning and signal processing, can be traced to the existence of one or more symmetry groups that arise due to the choice of mathematical assumptions, and we will encounter many examples of this phenomena in later chapters, which often lead to significant computational efficiencies. A striking example of the consequences of groups in classical algebra is the explanation for why there are no solutions that can be written in terms of addition, multiplication and roots, to the general polynomial equation $\sum_{i=0}^{N} a_i x^i = 0$ when $N \geq 5$. This fact has many practical consequences, for example, it is possible to find the eigenvalues of a general matrix of size $N \times N$ using simple analytical calculations when $N < 5$ (although the analytical calculations do become prohibitively complex), but there is no possibility of using similar analytical techniques when $N \geq 5$, and one must resort to numerical methods, and these methods sometimes cannot guarantee to find all solutions!

Many simple groups with the same number of elements are *isomorphic* to each other, that is, there is a unique function that maps the elements of one group to the elements of the other, such that the operations can be applied consistently to the mapped elements. Intuitively then, the identity in one group is mapped to that of the other group. For example, the rotation group V_4 above is isomorphic to the group $M_8 = (\times_8, \{1, 3, 5, 7\})$, where$\times_8$ indicates multiplication modulo 8 (that is, taking the remainder of the multiplication on division by 8, see Figure 1.3).

Whilst two groups might not be isomorphic, they are sometimes *homomorphic*: there is a function between one group and the other that maps each element in the first group to one or more elements in the second group, but the mapping is still consistent under the second operation. A very important example is the exponential map, $\exp(x)$, that converts addition over the set of real numbers, to multiplication over the set of positive real numbers: $e^{a+b} = e^a e^b$; a powerful variant of this map is widely used in statistical inference to simplify and stabilize calculations involving the probabilities of independent statistical events,

$(+, \mathbb{R})$	$(\times, \mathbb{R} - \{0\})$
$\ln(a) = b$	$e^b = a$
$\ln(a_1) + \ln(a_2)$ $= \ln(a_1 \times a_2)$	$e^{b_1} \times e^{b_2}$ $= e^{b_1 + b_2}$
$\ln 1 = 0$	$e^0 = 1$

Fig. 1.1: Illustrating abstract groups and mapping between them. Shown are the two continuous groups of real numbers under addition is (left column) and multiplication (right column), with identities 0 and 1 respectively. The homomorphism of exponentiation maps addition onto multiplication (left to right column), and the inverse, the logarithm, maps multiplication back onto addition (right to left column). Therefore, these two groups are homomorphic.

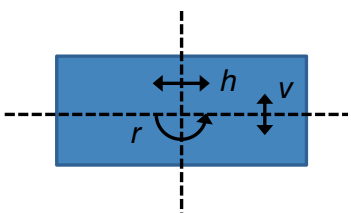

Fig. 1.2: The group of symmetries of the rectangle, $V_4 = (\circ, \{e, h, v, r\})$. It consists of horizontal and vertical flips, and a rotation of 180° about the centre. This group is isomorphic to the group $M_8 = (\times_8, \{1, 3, 5, 7\})$ (see Figure 1.3).

by converting them into calculations with their associated information content. This is the negative logarithmic map $-\ln(x)$, that converts probabilities under multiplication, into entropies under addition. This map is very widely used in statistical inference as we shall see.

For a more detailed but accessible background to group theory, read Humphreys (1996).

Rings

Whilst groups deal with one operation on a set of numbers, *rings* are a slightly more complex structure that often arises when two operations are applied to the same set. The most immediately tangible example is the operations of addition and multiplication on the set of integers $\mathbb{Z}$ (the positive and negative whole numbers with zero). Using the definition above, the set of integers under addition form an Abelian group, whereas under multiplication the integers form a simple structure known as a *monoid* – a group without inverses. Multiplication with the integers is associative, and there is an identity (the positive number one), but the multiplicative inverses are not integers (they are fractions such as $1/2$, $-1/5$ etc.) Finally, in combination, integer multiplication *distributes* over integer addition: $a \times (b + c) = a \times b + a \times c = (b + c) \times a$. These properties define a ring: it has one operation that together with the set forms an Abelian group, and another operation that, with the set, forms a monoid, and the second operation distributes over the first. As with integers, the set of real numbers under the usual addition and multiplication also has the structure of a ring. Another very important example is the set of *square matrices* all of size $N \times N$ with real elements under normal matrix addition and multiplication. Here the multiplicative identity element is the *identity matrix* of size $N \times N$, and the additive identity element is the same size square matrix with all zero elements.

Rings are powerful structures that can lead to very substantial computational savings for many statistical machine learning and signal processing problems. For example, if we remove the condition that the additive operation must have inverses, then we have a pair of monoids that are distributive. This structure is known as a *semiring* or *semifield* and it turns out that the existence of this structure in many machine learning and signal processing problems makes these otherwise computationally intractable problems feasible. For example, the classical *Viterbi algorithm* for determining the most likely sequence of hidden states in a *Hidden Markov Model* (HMM) is an application of the *max-sum semifield* on the *dynamic Bayesian network* that defines the stochastic dependencies in the model.

Both Dummit and Foote (2004) and Rotman (2000) contain detailed introductions to abstract algebra including groups and rings.

$\circ$	***e***	***h***	***v***	***r***
e	*e*	*h*	*v*	*r*
h	*h*	*e*	*r*	*v*
v	*v*	*r*	*e*	*h*
r	*r*	*v*	*h*	*e*

$\times_8$	**1**	**3**	**5**	**7**
1	1	3	5	7
3	3	1	7	5
5	5	7	1	3
7	7	5	3	1

Fig. 1.3: The table for the symmetry group $V_4 = (\circ, \{e, h, v, r\})$ (top), and the group $M_8 = (\times_8, \{1, 3, 5, 7\})$ (bottom), showing the isomorphism between them obtained by mapping $e \mapsto 1$, $h \mapsto 3$, $v \mapsto 5$ and $r \mapsto 7$.

1.2 Metrics

Distance is a fundamental concept in mathematics. Distance functions play a key role in machine learning and signal processing, particularly as measures of similarity between objects, for example, digital signals encoded as items of a set. We will also see that a statistical model often implies the use of a particular measure of distance, and this measure determines the properties of statistical inferences that can be made.

A *geometry* is obtained by attaching a notion of distance to a set: it becomes a *metric space*. A *metric* takes two points in the set and returns a single (usually real) value representing the distance between them. A metric must have the following properties to satisfy intuitive notions of distance:

(1) Non-negativity: $d(x, y) \geq 0$,
(2) Symmetry: $d(x, y) = d(y, x)$,
(3) Coincidence: $d(x, x) = 0$, and
(4) Triangle inequality: $d(x, z) \leq d(x, y) + d(y, z)$.

Respectively, these requirements are that (1) distance cannot be negative, (2) the distance going from x to y is the same as that from y to x, (3) only points lying on top of each other have zero distance between them, and (4) the length of any one side of a triangle defined by three points cannot be greater than the sum of the length of the other two sides. For example, the *Euclidean metric* on a D-dimensional set is:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{i=1}^{D} (x_i - y_i)^2} \tag{1.1}$$

This represents the notion of distance that we experience in everyday geometry. The defining properties of distance lead to a vast range of possible geometries, for example, the *city-block geometry* is defined by the absolute distance metric:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{D} |x_i - y_i| \tag{1.2}$$

City-block distance is so named because it measures distances on a grid parallel to the co-ordinate axes. Distance need not take on real values, for example, the *discrete metric* is defined as $d(x, y) = 0$ if $x = y$ and $d(x, y) = 1$ otherwise. Another very important metric is the *Mahalanobis distance*:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{y})} \tag{1.3}$$

This distance may not be axis-aligned: it corresponds to the finding the Euclidean distance after applying an arbitrary stretch or compression along each axis followed by an arbitrary D-dimensional rotation (indeed if $\boldsymbol{\Sigma} = \mathbf{I}$, the identity matrix, this is identical to the Euclidean distance).

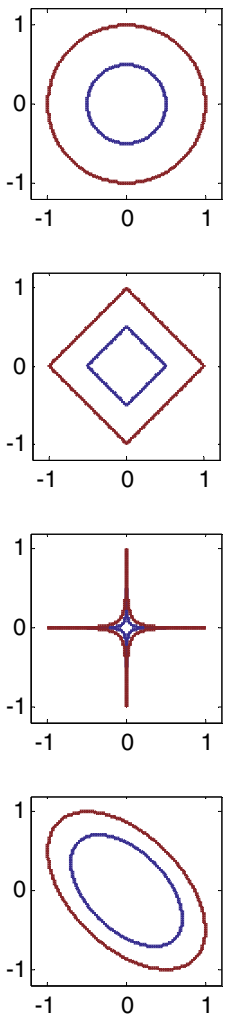

Fig. 1.4: Metric 2D circles $d(\boldsymbol{x}, \mathbf{0}) = c$ for various distance metrics. From top to bottom, Euclidean distance, absolute distance, the distance $d(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{D} |x_i - y_i|^{0.3}\right)^{0.3^{-1}}$, and the Mahalanobis distance for $\Sigma_{11} = \Sigma_{22} = 1.0$, $\Sigma_{12} = \Sigma_{21} = -0.5$. The contours are $c = 1$ (red lines) and $c = 0.5$ (blue lines).

Figure 1.4 shows plots of 2D *circles* $d(\boldsymbol{x}, \mathbf{0}) = c$ for various metrics, in particular $c = 1$ which is known as the *unit circle* in that metric space.

For further reading, metric spaces are introduced in Sutherland (2009) in the context of real analysis and topology.

1.3 Vector spaces

A *space* is just the name given to a set endowed with some additional mathematical structure. A (real) *vector space* is the key structure of linear algebra that is a central topic in most of classical signal processing — all digital signals are vectors, for example. The definition of a (finite) vector space begins with an ordered set (often written as a column) of N real numbers called a *vector*, and a single real number called a *scalar*. To that vector we attach the addition operation which is both associative and commutative, that simply adds every corresponding element of the numbers in each vector together, written as $\boldsymbol{v} + \boldsymbol{u}$. The identity for this operation is the vector with N zeros, $\mathbf{0}$. Additionally, we define a scalar multiplication operation that multiplies each element of the vector with a scalar, λ. Using the scalar multiplication by $\lambda = -1$, we can then form inverses of any vector. Scalar multiplication should not matter in which order two scalar multiplications occur, e.g. $\lambda(\mu\boldsymbol{v}) = (\lambda\mu)\boldsymbol{v} = (\mu\lambda)\boldsymbol{v} = \mu(\lambda\boldsymbol{v})$. We also require that scalar multiplication distributes over vector addition, $\lambda(\boldsymbol{v} + \boldsymbol{u}) = \lambda\boldsymbol{v} + \lambda\boldsymbol{u}$, and scalar addition distributes over scalar multiplication, $(\lambda + \mu)\boldsymbol{v} = \lambda\boldsymbol{v} + \mu\boldsymbol{v}$.

Every vector space has at least one *basis* for the space: this is a set of linearly independent vectors, such that every vector in the vector space can be written as a unique linear combination of these basis vectors (Figure 1.5). Since our vectors have N entries, there are always N vectors in the basis. Thus, N is the *dimension* of the vector space. The simplest basis is the so-called *standard basis*, consisting of the N vectors $\boldsymbol{e}_1 = (1, 0, \ldots, 0)^T$, $\boldsymbol{e}_2 = (0, 1, \ldots, 0)^T$ etc. It is easy to see that a vector $\boldsymbol{v} = (v_1, v_2, \ldots, v_N)^T$ can be expressed in terms of this basis as $\boldsymbol{v} = v_1\boldsymbol{e}_1 + v_2\boldsymbol{e}_2 + \cdots + v_N\boldsymbol{e}_N$.

By attaching a *norm* to a vector space (see below), we can measure the length of any vector, the vector space is then referred to as a *normed space*. To satisfy intuitive notions of length, a norm $V(\boldsymbol{u})$ must have the following properties:

(1) Non-negativity: $V(\boldsymbol{u}) \geq 0$,

(2) Positive scalability: $V(\alpha\boldsymbol{u}) = |\alpha| V(\boldsymbol{u})$,

(3) Separation: If $V(\boldsymbol{u}) = 0$ then $\boldsymbol{u} = \mathbf{0}$, and

(4) Triangle inequality: $V(\boldsymbol{u} + \boldsymbol{v}) \leq V(\boldsymbol{u}) + V(\boldsymbol{v})$.

Often, the notation $\|\boldsymbol{u}\|$ is used. Probably most familiar is the Euclidean norm $\|\boldsymbol{u}\|_2 = \sqrt{\sum_{i=1}^{N} u_i^2}$, but another norm that gets heavy use in statistical machine learning is the L_p-norm:

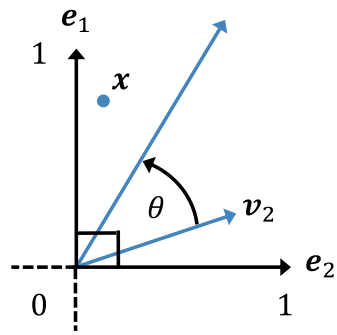

Fig. 1.5: Important concepts in 2D vector spaces. The standard basis $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ is aligned with the axes. Two other vectors $(\boldsymbol{v}_1, \boldsymbol{v}_2)$ can also be used as a basis for the space, all that is required is they are linearly independent of each other (in the 2D case, they are not simply scalar multiples of each other). Then $\boldsymbol{x}$ can be represented in either basis. The dot product between two vectors is proportional to the cosine of the angle between them: $\cos(\theta) \propto \langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle$. Because $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ are at mutual right angles, they have zero dot product and therefore the basis is orthogonal. Additionally, it is an orthonormal basis because they are unit norm (length) $(\|\boldsymbol{e}_1\| = \|\boldsymbol{e}_2\| = 1)$. The other basis vectors are neither orthogonal nor unit norm.

$$\|\boldsymbol{u}\|_p = \left(\sum_{i=1}^{N} |u_i|^p \right)^{\frac{1}{p}} \tag{1.4}$$

of which the Euclidean ($p = 2$) and city-block ($p = 1$) norms are special cases. Also of importance is the max-norm $\|\boldsymbol{u}\|_\infty = \max_{i=1\ldots N} |u_i|$, which is just the length of the largest co-ordinate.

There are several ways in which a product of vectors can be formed. The most important in our context is the *inner product* between two vectors:

$$\alpha = \langle \boldsymbol{u}, \boldsymbol{v} \rangle = \sum_{i=1}^{N} u_i v_i \tag{1.5}$$

This is sometimes also described as the *dot product* $\boldsymbol{u} \cdot \boldsymbol{v}$. For complex vectors, this is defined as:

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \sum_{i=1}^{N} u_i \bar{v}_i \tag{1.6}$$

where $\bar{a}$ is the complex conjugate of $a \in \mathbb{C}$.

We will see later that the dot product plays a central role in the statistical notion of correlation. When two vectors have zero inner product, they are said to be *orthogonal*; geometrically they meet at a right-angle. This also has a statistical interpretation: for certain random variables, orthogonality implies statistical independence. Thus, orthogonality leads to significant simplifications in common calculations in classical DSP.

A special and very useful kind of basis is an *orthogonal basis* where the inner product between every pair of distinct basis vectors is zero:

$$\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = 0 \qquad \text{for all } i \neq j,\ i, j = 1, 2 \ldots N \tag{1.7}$$

In addition, the basis is *orthonormal* if every basis vector has unit norm $\|\boldsymbol{v}_i\| = 1$ – the standard basis is orthonormal, for example (Figure 1.5). Orthogonality/orthonormality dramatically simplifies many calculations over vector spaces, partly because it is straightforward to find the N scalar coefficients a_i of an arbitrary vector $\boldsymbol{u}$ in this basis using the inner product:

$$a_i = \frac{\langle \boldsymbol{u}, \boldsymbol{v}_i \rangle}{\|\boldsymbol{v}_i\|^2} \tag{1.8}$$

which simplifies to $a_i = \langle \boldsymbol{u}, \boldsymbol{v}_i \rangle$ in the orthonormal case. Orthonormal bases are the backbone of many methods in DSP and machine learning.

We can express the Euclidean norm using the inner product: $\|\boldsymbol{u}\|_2 = \sqrt{\boldsymbol{u} \cdot \boldsymbol{u}}$. An inner product satisfies the following properties:

(1) Non-negativity: $\boldsymbol{u} \cdot \boldsymbol{v} \geq 0$,

(2) Symmetry: $\boldsymbol{u} \cdot \boldsymbol{v} = \boldsymbol{v} \cdot \boldsymbol{u}$, and

(3) Linearity: $(\alpha \boldsymbol{u}) \cdot \boldsymbol{v} = \alpha (\boldsymbol{u} \cdot \boldsymbol{v})$.

There is an intuitive connection between distance and length: assuming that the metric is homogeneous $d(\alpha \boldsymbol{u}, \alpha \boldsymbol{v}) = |\alpha| \, d(\boldsymbol{u}, \boldsymbol{v})$ and translation invariant $d(\boldsymbol{u}, \boldsymbol{v}) = d(\boldsymbol{u} + \boldsymbol{a}, \boldsymbol{v} + \boldsymbol{a})$, a norm can be defined as the distance to the origin $\|\boldsymbol{u}\| = d(\boldsymbol{0}, \boldsymbol{u})$. A commonly occurring example of this the so-called *squared L_2 weighted norm* $\|\boldsymbol{u}\|_{\mathbf{A}}^2 = \boldsymbol{u}^T \mathbf{A} \boldsymbol{u}$ which is just the squared Mahalanobis distance $d(\boldsymbol{0}, \boldsymbol{u})^2$ discussed earlier with $\boldsymbol{\Sigma}^{-1} = \mathbf{A}$.

On the other hand, there is one sense in which every norm *induces* an associated metric with the construction $d(\boldsymbol{u}, \boldsymbol{v}) = \|\boldsymbol{u} - \boldsymbol{v}\|$. This construction enjoys extensive use in machine learning and statistical DSP to quantify the "discrepancy" or "error" between two signals. In fact, since norms are *convex* (discussed later), it follows that metrics constructed this way from norms, are also convex, a fact of crucial importance in practice.

A final product we will have need for in later chapters is the *elementwise product* $\boldsymbol{w} = \boldsymbol{u} \circ \boldsymbol{v}$ which is obtained by multiplying each element in the vector together $w_n = u_n v_n$.

Linear operators

A linear operator or *map* acts on vectors to create other vectors, and while doing so, preserve the operations of vector addition and scalar multiplication. They are homomorphisms between vector spaces. Linear operators are fundamental to classical digital signal processing and statistics, and so find heavy use in machine learning. Linear operators L have the *linear combination* property:

$$\begin{aligned} L[\alpha_1 \boldsymbol{u}_1 + \alpha_2 \boldsymbol{u}_2 + \cdots + \alpha_N \boldsymbol{u}_N] &= \\ \alpha_1 L[\boldsymbol{u}_1] + \alpha_2 L[\boldsymbol{u}_2] + \cdots + \alpha_N L[\boldsymbol{u}_N] \end{aligned} \tag{1.9}$$

What this says is that the operator commutes with scalar multiplication and vector addition: we get the same result if we first scale, then add the vectors, and then apply the operator to the result, or, apply the operator to each vector, then scale them, and them add up the results (Figure 1.6).

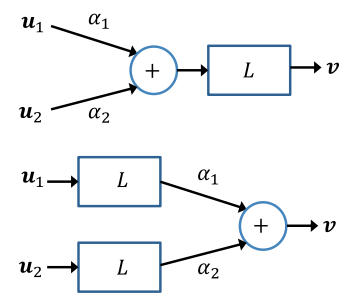

Fig. 1.6: A 'flow diagram' depicting linear operators. All linear operators share the property that the operator L applied to the scaled sum of (two or more) vectors $\alpha_1 \boldsymbol{u}_1 + \alpha_2 \boldsymbol{u}_2$ (top panel), is the same as the scaled sum of the same operator applied to each of these vectors first (bottom panel). In other words, it does not matter whether the operator is applied before or after the scaled sum.

Matrices (which we discuss next), differentiation and integration, and expectation in probability are all examples of linear operators. The linearity of integration and differentiation are standard rules which can be derived from the basic definitions. Linear maps in two-dimensional space have a nice geometric interpretation: straight lines in the vector space are mapped onto other straight lines (or onto a point if they are *degenerate* maps). This idea extends to higher dimensional vector spaces in the natural way.

Matrix algebra

When vectors are 'stacked together' they form a powerful structure

which is a central topic of much of signal processing, statistics and machine learning: *matrix algebra*. A *matrix* is a 'rectangular' array of $N \times M$ elements, for example, the 3×2 matrix $\mathbf{A}$ is:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \tag{1.10}$$

This can be seen to be two length three vectors stacked side-by-side. The elements of a matrix are often written using the subscript notation a_{ij} where $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$. Matrix addition of two matrices, is commutative: $\mathbf{C} = \mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$, where the addition is element-by-element i.e. $c_{ij} = a_{ij} + b_{ij}$.

As with vectors, there are many possible ways in which matrix multiplication could be defined: the one most commonly encountered is the row-by-column inner product. For two matrices $\mathbf{A}$ of size $N \times M$ and $\mathbf{B}$ of size $M \times P$, the product $\mathbf{C} = \mathbf{A} \times \mathbf{B}$ is a new matrix of size $N \times P$ defined as:

$$c_{ij} = \sum_{k=1}^{M} a_{ik} b_{kj} \qquad i = 1, 2, \ldots, N,\ j = 1, 2, \ldots, P \tag{1.11}$$

This can be seen to be the matrix of all possible inner products of each row of $\mathbf{A}$ by each column of $\mathbf{B}$. Note that the number of columns of the left hand matrix must match the number of rows of the right hand one. Matrix multiplication is associative, it distributes over matrix addition, and it is compatible with scalar multiplication: $\alpha \mathbf{A} = \mathbf{B}$ simply gives the new matrix with entries $b_{ij} = \alpha a_{ij}$, i.e. it is just columnwise application of vector scalar multiplication. Matrix multiplication is, however, *non-commutative*: it is not true in general that $\mathbf{A} \times \mathbf{B}$ gives the same result as $\mathbf{B} \times \mathbf{A}$.

A useful matrix operator is the *transpose* that swaps rows with columns; if $\mathbf{A}$ is an $N \times M$ matrix then $\mathbf{A}^T = \mathbf{B}$ is the $M \times N$ matrix $b_{ji} = a_{ij}$. Some of the properties of the transpose are: it is self-inverse $\left(\mathbf{A}^T\right)^T = \mathbf{A}$; respects addition $(\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$; and it reverses the order of factors in multiplication $(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$.

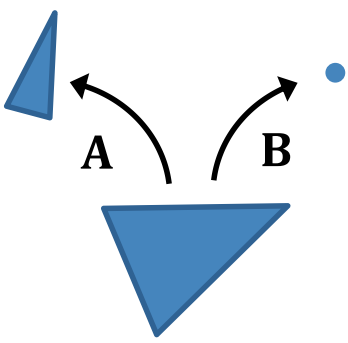

Fig. 1.7: A depiction of the geometric effect of invertible and non-invertible square matrices. The invertible square matrix A maps the triangle at the bottom to the 'thinner' triangle (for example, by transforming the vector for each vertex). It scales the area of the triangle by the determinant $|\mathbf{A}| \neq 0$. However, the non-invertible square matrix B collapses the triangle onto a single point with no area because $|\mathbf{B}| = 0$. Therefore, $\mathbf{A}^{-1}$ is well-defined, but $\mathbf{B}^{-1}$ is not.

Square and invertible matrices

So far, we have not discussed how to solve matrix equations. The case of addition is easy because we can use scalar multiplication to form the negative of a matrix, i.e. given $\mathbf{C} = \mathbf{A} + \mathbf{B}$, finding $\mathbf{B}$ requires us to calculate $\mathbf{B} = \mathbf{C} - \mathbf{A} = \mathbf{C} + (-1)\,\mathbf{A}$. In the case of multiplication, we need to find the "reciprocal" of a matrix, e.g. to solve $\mathbf{C} = \mathbf{AB}$ for $\mathbf{B}$ we would naturally calculate $\mathbf{A}^{-1}\mathbf{C} = \mathbf{A}^{-1}\mathbf{AB} = \mathbf{B}$ by the usual algebraic rules. However, things become more complicated because $\mathbf{A}^{-1}$ does not exist in general. We will discuss the conditions under which a matrix does have a multiplicative inverse next.

All *square matrices* of size $N \times N$ can be summed or multiplied to-

gether in any order. A square matrix $\mathbf{A}$ with all zero elements except for the *main diagonal*, i.e. $a_{ij} = 0$ unless $i = j$ is called a *diagonal matrix*. A special diagonal matrix, $\mathbf{I}$, is the *identity matrix* where the main diagonal entries $a_{ii} = 1$. Then, if the equality $\mathbf{AB} = \mathbf{I} = \mathbf{BA}$ holds, the matrix $\mathbf{B}$ must be well-defined (and unique) and it is the inverse of $\mathbf{A}$, i.e. $\mathbf{B} = \mathbf{A}^{-1}$. We then say that $\mathbf{A}$ is *invertible;* if it is not invertible then it is *degenerate* or *singular*.

The $N \times N$ identity matrix is denoted $\mathbf{I}_N$ or simply $\mathbf{I}$ when the context is clear and the size can be omitted.

There are many equivalent conditions for matrix invertibility, for example, the only solution to the equation $\mathbf{A}\boldsymbol{x} = \mathbf{0}$ is the vector $\boldsymbol{x} = \mathbf{0}$ or the columns of $\mathbf{A}$ are linearly independent. But one particularly important way to test the invertibility of a matrix is to calculate the *determinant* $|\mathbf{A}|$: if the matrix is singular, the determinant is zero. It follows that all invertible matrices have $|\mathbf{A}| \neq 0$. The determinant calculation is quite elaborate for a general square matrix, formulas exist but geometric intuition helps to understand these calculations: when a linear map defined by a matrix acts on a geometric object in vector space with a certain volume, the determinant is the *scaling factor* of the mapping. Volumes under the action of the map are scaled by the magnitude of the determinant. If the determinant is negative, the *orientation* of any geometric object is reversed. Therefore, invertible transformations are those that do not collapse the volume of any object in the vector space to zero (Figure 1.7).

Another matrix operator which finds significant use is the *trace* $tr\,(\mathbf{A})$ of a square matrix: this is just the sum of the diagonals, e.g. $\mathrm{tr}\,(\mathbf{A}) = \sum_{i=1}^{N} a_{ii}$. The trace is invariant to addition $\mathrm{tr}\,(\mathbf{A}+\mathbf{B}) = \mathrm{tr}\,(\mathbf{A}) + \mathrm{tr}\,(\mathbf{B})$, transpose $\mathrm{tr}\left(\mathbf{A}^T\right) = \mathrm{tr}\,(\mathbf{A})$ and multiplication $\mathrm{tr}\,(\mathbf{AB}) = \mathrm{tr}\,(\mathbf{BA})$. With products of three or more matrices the trace is invariant to cyclic permutations, with three matrices: $\mathrm{tr}\,(\mathbf{ABC}) = \mathrm{tr}\,(\mathbf{CAB}) = \mathrm{tr}\,(\mathbf{BCA})$.

Eigenvalues and eigenvectors

A ubiquitous computation that arises in connection with algebraic problems in vector spaces is the *eigenvalue problem* for the given $N \times N$ square matrix $\mathbf{A}$:

$$\mathbf{A}\boldsymbol{v} = \lambda \boldsymbol{v} \tag{1.12}$$

Any non-zero $N \times 1$ vector $\boldsymbol{v}$ which solves this equation is known as an *eigenvector* of $\mathbf{A}$, and the scalar value λ is known as the associated *eigenvalue*. Eigenvectors are not unique: they can be multiplied by any non-zero scalar and still remain eigenvectors with the same eigenvalues. Thus, often the unit length eigenvectors are sought as the solutions to (1.12).

It should be noted that (1.12) arises for vector spaces in general e.g. linear operators. An important example occurs in the vector space of functions $f\,(x)$ with differential the operator $L = \frac{d}{dx}$. Here, the corresponding eigenvalue problem is the differential equation $L\,[f\,(x)] = \lambda f\,(x)$, for which the solution is $f\,(x) = ae^{\lambda x}$ for any (non-zero) scalar value a. This is known as an *eigenfunction* of the differential operator

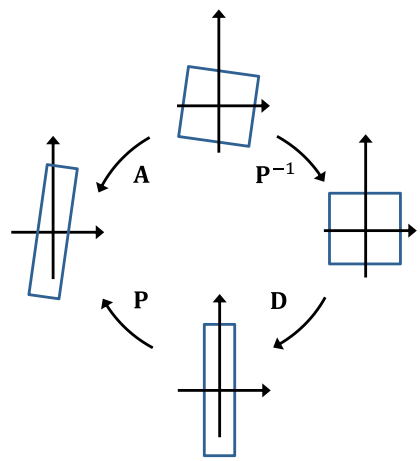

Fig. 1.8: An example of diagonalizing a matrix. The diagonalizable square matrix A has diagonal matrix D containing the eigenvalues, and transformation matrix P containing the eigenbasis, so $\mathbf{A} = \mathbf{PDP}^{-1}$. A maps the rotated square (top), to the rectangle in the same orientation (at left). This is equivalent to first 'unrotating' the square (the effect of $\mathbf{P}^{-1}$) such that it is aligned with the co-ordinate axes, then stretching/compressing the square along each axis (the effect of D), and finally rotating back to the original orientation (the effect of P).

L.

If they exist, the eigenvectors and eigenvalues of a square matrix $\mathbf{A}$ can be found by obtaining all scalar values λ such that$|(\mathbf{A} - \lambda\mathbf{I})| = 0$. This holds because $A\boldsymbol{v} - \lambda\boldsymbol{v} = \mathbf{0}$ if and only if $|(\mathbf{A} - \lambda\mathbf{I})| = 0$. Expanding out this determinant equation leads to an N-th order polynomial equation in λ, namely $a_N\lambda^N + a_{N-1}\lambda^{N-1} + \cdots + a_0 = 0$, and the roots of this equation are the eigenvalues.

This polynomial is known as the *characteristic polynomial* for $\mathbf{A}$ and determines the existence of a set of eigenvectors that is also a basis for the space, in the following way. The fundamental theorem of algebra states that this polynomial has exactly N roots, but some may be *repeated* (i.e. occur more than once). If there are no repeated roots of the characteristic polynomial, then the eigenvalues are all distinct, and so there are N eigenvectors which are all linearly independent. This means that they form a basis for the vector space, which is the *eigenbasis* for the matrix.

Not all matrices have an eigenbasis. However matrices that do are also *diagonalizable*, that is, they have the same geometric effect as a diagonal matrix, but in a different basis other than the standard one. This basis can be found by solving the eigenvalue problem. Placing all the eigenvectors into the columns of a matrix $\mathbf{P}$ and all the corresponding eigenvalues into a diagonal matrix $\mathbf{D}$, then the matrix can be rewritten:

$$\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1} \tag{1.13}$$

See Figure 1.8. A diagonal matrix simply scales all the coordinates of the space by a different, fixed amount. They are very simple to deal with, and have important applications in signal processing and machine learning. For example, the Gaussian distribution over multiple variables, one of the most important distributions in practical applications, encodes the probabilistic relationship between each variable in the problem with the *covariance matrix*. By diagonalizing this matrix, one can find a linear mapping which makes all the variables statistically independent of each other: this dramatically simplifies many subsequent calculations.

Despite the central importance of the eigenvectors and eigenvalues a linear problem, it is generally not possible to find all the eigenvalues by analytical calculation. Therefore one generally turns to iterative numerical algorithms to obtain an answer to a certain precision.

Special matrices

Beyond what has been already discussed, there is not that much more to be said about general matrices which have $N \times M$ degrees of freedom. Special matrices with fewer degrees of freedom have very interesting properties and occur frequently in practice.

Some of the most interesting special matrices are *symmetric matrices* with real entries – *self-transpose* and so square by definition, i.e. $\mathbf{A}^T = \mathbf{A}$. These matrices are always diagonalizable, and have an orthogonal eigenbasis. The eigenvalues are always real. If the inverse exists, it is

also symmetric. A symmetric matrix has $\frac{1}{2}N(N+1)$ unique entries, on the order of half the N^2 entries of an arbitrary square matrix.

Positive-definite matrices are a special kind of symmetric matrix for which $\boldsymbol{v}^T\mathbf{A}\boldsymbol{v} > 0$ for any non-zero vector $\boldsymbol{v}$. All the eigenvalues are positive. Take any real, invertible matrix $\mathbf{B}$ (so that $\mathbf{B}\boldsymbol{v} \neq \mathbf{0}$ for all such $\boldsymbol{v}$) and let $\mathbf{A} = \mathbf{B}^T\mathbf{B}$, then $\boldsymbol{v}^T\mathbf{B}^T\mathbf{B}\boldsymbol{v} = (\mathbf{B}\boldsymbol{v})^T(\mathbf{B}\boldsymbol{v}) = \|\mathbf{B}\boldsymbol{v}\|_2^2 > 0$ making $\mathbf{A}$ positive-definite. As will be described in the next section, these kinds of matrices are very important in machine learning and signal processing because the covariance matrix of a set of random variables is positive-definite for exactly this reason.

Orthonormal matrices have all columns which are vectors that form an orthonormal basis for the space. The determinant of these matrices is either $+1$ or -1. Like symmetric matrices they are always diagonalizable, although the eigenvectors are generally complex with modulus 1. An orthonormal matrix is always invertible, the inverse is also orthonormal and equal to the transpose, $\mathbf{A}^T = \mathbf{A}^{-1}$. The subset with determinant $+1$, correspond to *rotations* in the vector space.

For *upper* (*lower*) *triangular matrices*, the diagonal and the entries above (below) the diagonal are non-zero, the rest zero. These matrices often occur when solving matrix problems such as $\mathbf{A}\boldsymbol{v} = \boldsymbol{b}$, because the matrix equation $\mathbf{L}\boldsymbol{v} = \boldsymbol{b}$ is simple to solve by *forward substitution* if $\mathbf{L}$ is lower-triangular. Forward substitution is a straightforward sequential procedure which first obtains v_1 in terms of b_1 and l_{11}, then v_2 in terms of b_1, l_{21} and l_{22} etc. The same holds for upper triangular matrices and *backward substitution*. Because of the simplicity of these substitution procedures, there exist methods for decomposing a matrix into a product of upper or lower triangular matrices and a companion matrix.

Toeplitz matrices are matrices with $2N-1$ degrees of freedom that have constant diagonals, that is, the elements of $\mathbf{A}$ have entries $a_{ij} = c_{i-j}$. All *discrete convolutions* can be represented as Toeplitz matrices, and as we will discuss later, this makes them of fundamental importance in DSP. Because of the reduced degrees of freedom and special structure of the matrix, a Toeplitz matrix problem $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ is computationally easier to solve than a general matrix problem: a method known as the Levinson recursion dramatically reduces the number of arithmetic operations needed.

Circulant matrices are Toeplitz matrices where each row is obtained from the row above by rotating it one element to the right. With only N degrees of freedom they are highly structured and can be understood as discrete circular convolutions. The eigenbasis which diagonalizes the matrix is the *discrete Fourier basis* which is one of the cornerstones of classical DSP. It follows that any circulant matrix problem can be very efficiently solved using the *fast Fourier transform* (FFT).

Dummit and Foote (2004) contains an in-depth exposition of vector spaces from an abstract point of view, whereas Kaye and Wilson (1998) is an accessible and more concrete introduction.

1.4 Probability and stochastic processes

Probability is a formalization of the intuitive notion of uncertainty. Statistics is built on probability. Therefore, statistical DSP and machine learning has, at it's root, the quantitative manipulation of uncertainties. *Probability theory* contains the axiomatic foundation of uncertainty.

Sample spaces, events, measures and distributions

We start with a set of elements, say, Ω, which are known as *outcomes*. This is what we get as the result of a measurement or experiment. The set Ω is known as the *sample space* or *universe*. For example, the die has six possible outcomes, so the sample space is $\Omega = \{1, 2, 3, 4, 5, 6\}$. Given these outcomes, we want to quantify the probability of certain *events* occurring, for example, that we get a six or an even number in any throw. These events form an abstract $\sigma-$*algebra*, $\mathcal{F}$, which is, all the sets of subsets of outcomes that can be constructed by applying the elementary set operations of complement and (countable) unions to a selection of the elements in 2^{Ω} (the set of all subsets of Ω). The elements of $\mathcal{F}$ are the events. For example, in the coin toss, there are two possible outcomes, heads and tails, so $\Omega = \{H, T\}$. A set of events that are of interest make up a$\sigma-$algebra $\mathcal{F} = \{\varnothing, \{H\}, \{T\}, \Omega\}$, so that we can calculate the probability of heads or tails, none, or heads *or* tails occurring (N.B. the last two events are in some senses 'obvious' — the first is impossible and the second inevitable — so they require no calculation to evaluate, but we will see that to do probability calculus we always need the empty set and the set of all outcomes).

Given the pair $\Omega, \mathcal{F}$ we want to assign *probabilities* to events, which are real numbers lying between 0 and 1. An event with probability 0 is impossible and will never occur, whereas if the event has probability 1 then it is certain to occur. A mapping that determines the probability of any event is known as a *measure function* $\mu : \mathcal{F} \to \mathbb{R}$. An example would be the measure function for the fair coin toss which is $\mu(\{\varnothing\}) = 0$, $\mu(\{H, T\}) = 1$, $\mu(\{H\}) = \mu(\{T\}) = \frac{1}{2}$. A measure satisfies the following rules:

(1) Non-negativity: $\mu(A) \geq 0$ for all $A \in \mathcal{F}$,

(2) Unit measure: $\mu(\Omega) = 1$ and,

(3) Disjoint additivity: $\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mu(A_i)$ if the events $A_i \in \mathcal{F}$ do not overlap with each other (that is, they are mutually disjoint and so contain no elements from the sample space in common).

We mainly use the notation $P(A)$for the probability (measure) of event A. We can derive some important consequences of these rules. For example, if one event is wholly contained inside another, it must have smaller probability: if $A \subseteq B$ then $P(A) \leq P(B)$ with equality if $A = B$. Similarly, the probability of the event not occurring is one minus the probability of that event: $P\left(\bar{A}\right) = 1 - P(A)$.

Of great importance to statistics is the sample space of real numbers. A useful σ-algebra is the *Borel algebra* formed from all possible (open) intervals of the real line. With this algebra we can assign probabilities to ranges of real numbers, e.g. $P([a,b])$ for the real numbers $a \leq b$. An important consequence of the axioms is that $P(\{a\}) = 0$, i.e. point set events have zero probability. This differs from discrete (countable) sample spaces where the probability of any single element from the sample space can be non-zero.

Given a set of all possible events it is often natural to associate numerical 'labels' to each event. This is extremely useful because then we can perform meaningful numerical computations on the events. *Random variables* are functions that map the outcomes to numerical values, for example the random variable that maps coin tosses into the set $\{0,1\}$, $X(\{T\}) = 0$ and $X(\{H\}) = 1$. *Cumulative distribution functions* (CDFs) are measures as defined above, but where the events are selected through the random variable. For example, the CDF of the (fair) coin toss as described above would be:

$$P(\{A \in \{H,T\} : X(A) \leq x\}) = \begin{cases} \frac{1}{2} & \text{for } x = 0 \\ 1 & \text{for } x = 1 \end{cases} \tag{1.14}$$

This is a special case of the *Bernoulli distribution* (see below). Two common shorthand ways of writing the CDF are $F_X(x)$ and $P(X \leq x)$.

When the sample space is the real line, the random variable is *continuous*. The associated probability of an event, in this case, a (half open) interval of the real line is:

$$P((a,b]) = P(\{A \in \mathbb{R} : a < X(A) \leq b\}) = F_X(b) - F_X(a) \tag{1.15}$$

Often we can also define a distribution through a *probability density function* (PDF) $f_X(x)$:

$$P(A) = \int_A f_X(x)\, dx \tag{1.16}$$

where $\mathcal{A} \in \mathcal{F}$, and in practice statistical DSP and machine learning is most often (though not exclusively) concerned with $\mathcal{F}$ being the set of all open intervals of the real line (the Borel algebra), or some subset of the real line such as $[0,\infty)$. To satisfy the unit measure requirement, we must have that $\int_{\mathbb{R}} f_X(x)\, dx = 1$ (for the case of the whole real line). In the discrete case, the equivalent is the *probability mass function* (PMF) that assigns a probability measure to each separate outcome. To simplify the notation, we often drop the random variable subscript when the context is clear, writing e.g. $F(x)$, $f(x)$.

We can deduce certain properties of CDFs. Firstly, they must be non-decreasing, because the associated PMF/PDFs must be non-negative. Secondly, if X is defined on the range $[a,b]$, we must have that $F_X(a) = 0$ and $F_X(b) = 1$ (in the commonly occurring case where either a or b are infinite, then we would have, e.g. $\lim_{x\to-\infty} F_X(x) = 0$ and/or

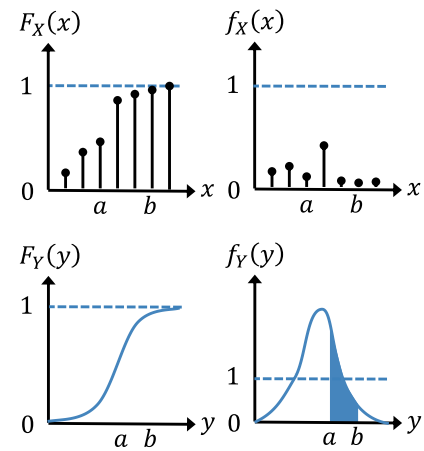

Fig. 1.9: Distribution functions and probabilities for ranges of discrete (top) and continuous (bottom) random variables X and Y respectively. Cumulative distribution functions (CDF) F_X and F_Y are shown on the left, and the associated probability mass (PMF) and probability density functions (PDF) f_X and f_Y on the right. For discrete X defined on the integers, the probability of the event A such that $a \leq X(A) \leq b$ is $\sum_{x=a}^{b} f_X(x)$ which is the same as $F_X(b) - F_X(a-1)$. For the continuous Y defined on the reals, the probability of the event $A = (a,b]$ is the given area under the curve of f_Y, i.e. $P(A) = \int_A f_Y(y)\, dy$. This is just $F_Y(b) - F_Y(a)$ by the fundamental theorem of calculus.

$\lim_{x\to\infty} F_X(x) = 1$). An important distinction to make here between discrete and continuous random variables, is that the PDF can have $f(x) > 1$ for some x in the range of the random variable, whereas PMFs must have $0 \leq f(x) \leq 1$ for all values of x in range. In the case of PMFs this is necessary to satisfy the unit measure property. These concepts are illustrated in Figure 1.9.

An elementary example of a PMF is the fair coin for which:

$$f(x) = \frac{1}{2} \quad \text{for } x \in \{0, 1\} \tag{1.17}$$

To satisfy unit measure, we must have $\sum_{a\in X(\Omega)} f(a) = 1$. The measure of an event is similarly:

$$P(A) = \sum_{a\in X(A)} f(a) \tag{1.18}$$

Some ubiquitous PMFs include the Bernoulli distribution which represents the binary outcome:

$$f(x) = \begin{cases} 1-p & \text{for } x = 0 \\ p & \text{for } x = 1 \end{cases} \tag{1.19}$$

A compact representation is $f(x) = (1-p)^{1-x} p^x$. A very important continuous distribution is the *Gaussian distribution*, whose density function is:

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{\sigma^2}\right) \tag{1.20}$$

The semicolon is used to separate the random variable from the adjustable (non-random) parameters that determine the form of the precise distribution of X. When the parameters are considered random variables, the bar notation $f(x|\,\mu, \sigma)$ is used instead, indicating that X depends, in a consistent probabilistic sense, on the value of the parameters. This latter situation occurs in the *Bayesian framework* as we will discuss later.

Joint random variables: independence, conditionals, and marginals

Often we are interested in the probability of multiple simultaneous events occurring. A consistent way to construct an underlying sample space is to form the set of all possible combinations of events. This is known as the *product sample space*. For example, the product sample space of two coin tosses is the set $\Omega = \{(H,H), (H,T), (T,H), (T,T)\}$ with $\sigma-$algebra $\mathcal{F} = \{\varnothing, \{(H,H)\}, \{(H,T)\}, \{(T,H)\}, \{(T,T)\}, \Omega\}$. As with single outcomes, we want to define a probability measure so that we can evaluate the probability of any joint outcome. This measure is known as the *joint CDF*:

$$F_{XY}(x,y) = P(X \leq x \text{ and } Y \leq y) \tag{1.21}$$

In words, the joint CDF is the probability that the pair of random variables X, Y simultaneously take on values that are equal to x, y at the most. For the case of continuous random variables, each defined on the whole real line, this probability is a multiple integration:

$$F_{XY}(x,y) = \int_{-\infty}^{y} \int_{-\infty}^{x} f(u,v)\, du\, dv \tag{1.22}$$

where $f(u,v)$ is the *joint PDF*. The sample space is now the plane $\mathbb{R}^2$, and so in order to satisfy the unit measure axiom, it must be that $\int_{\mathbb{R}} \int_{\mathbb{R}} f(u,v)\, du\, dv = 1$. The probability of any region A of $\mathbb{R}^2$ is then the multiple integral over that region: $P(A) = \int_A f(u,v)\, du\, dv$.

The corresponding discrete case has that $P(A \times B) = \sum_{a \in X(A)} \sum_{b \in Y(B)} f(a,b)$ for any product of events where $A \in \Omega_X$, $B \in \Omega_Y$, and Ω_X,Ω_Y are the sample spaces of X and Y respectively, and $f(a,b)$ is the *joint PMF*. The joint PMF must sum to one over the whole product sample space: $\sum_{a \in X(\Omega_X)} \sum_{b \in Y(\Omega_Y)} f(a,b) = 1$.

More general joint events over N variables are defined similarly and associated with multiple CDFs, PDFs and PMFs, e.g. $f_{X_1 X_2 \ldots X_N}(x_1, x_2 \ldots x_N)$ and, when the context is clear from the arguments of the function, we drop the subscript in the name of the function for notational simplicity. This naturally allows us to define distribution functions over vectors of random variables, e.g. $f(\boldsymbol{x})$ for $\boldsymbol{X} = (X_1, X_2 \ldots X_N)^T$ where typically, each element of the vector comes from the same sample space.

Given the joint PMF/PDF, we can always 'remove' one or more of the variables in the joint set by *integrating out* this variable, e.g.:

$$f(x_1, x_3, \ldots, x_N) = \int_{\mathbb{R}} f(x_1, x_2, x_3, \ldots, x_N)\, dx_2 \tag{1.23}$$

This computation is known as *marginalization*.

When considering joint events, we can perform calculations about the *conditional probability* of one event occurring, when another has already occurred (or is otherwise fixed). This conditional probability is written using the bar notation $P(X = x | Y = y)$: described as the 'probability that the random variable $X = x$, given that $Y = y$'. For PMFs and PDFs we will shorten this to $f(x|y)$. This probability can be calculated from the joint and single distributions of the conditioning variable:

$$f(x|y) = \frac{f(x,y)}{f(y)} \tag{1.24}$$

In effect, the conditional PMF/PDF is what we obtain from restricting the joint sample space to the set for which $Y = y$, and calculating the measure of the intersection of the joint sample space for any chosen x. The division by $f(y)$ensures that the conditional distribution is itself a normalized measure on this restricted sample space, as we can show by

marginalizing out X from the right hand side of the above equation.

If the distribution of X does not depend upon Y, we say that X is *independent of* Y. In this case $f(x|y) = f(x)$. This implies that $f(x,y) = f(x)f(y)$, i.e. the joint distribution over X, Y factorizes into a product of the marginal distributions over X, Y. Independence is a central topic in statistical DSP and machine learning because whenever two or more variables are independent, this can lead to very significant simplifications that in some cases, make the difference between whether a problem is tractable at all. In fact, it is widely recognized these days that the main goal of statistical machine learning is to find *good factorizations* of the joint distribution over all the random variables of a problem.

Bayes' rule

If we have the distribution function of a random variable conditioned on another, is it possible to swap the role of conditioned and conditioning variables? The answer is yes: provided that we have all the marginal distributions. This leads us into the territory of *Bayesian reasoning*. The calculus is straightforward, but the consequences are of profound importance to statistical DSP and machine learning. We will illustrate the concepts using continuous random variables, but the principles are general and apply to random variables over any sample space. Suppose we have two random variables X, Y and we know the conditional distribution of X given Y, then the conditional distribution of Y on X is:

$$f(y|x) = \frac{f(x|y)f(y)}{f(x)} \tag{1.25}$$

This is known as *Bayes' rule*. In the Bayesian formalism, $f(x|y)$ is known as the *likelihood*, $f(y)$ is known as the *prior*, $f(x)$ is the *evidence* and $f(y|x)$ is the *posterior*.

Often, we do not know the distribution over X; but since the numerator in Bayes' rule is the joint probability of X and Y, this can be obtained by marginalizing out Y from the numerator:

$$f(y|x) = \frac{f(x|y)f(y)}{\int_{\mathbb{R}} f(x|y)f(y)\,dy} \tag{1.26}$$

This form of Bayes' rule is ubiquitous because it allows calculation of the posterior knowing only the likelihood and the prior.

Unfortunately, one of the hardest and most computationally intractable problems in applying the Bayesian formalism arises when attempting to evaluate integrals over many variables to calculate the posterior in (1.26). Fortunately however, there are common situations in which it is not necessary to know the evidence probability. A third restatement of Bayes' rule makes it clear that the evidence probability can be considered a 'normalizer' for the posterior, ensuring that the posterior satisfies the unit measure property:

$$f(y|x) \propto f(x|y)\, f(y) \tag{1.27}$$

This form is very commonly encountered in many statistical inference problems in machine learning. For example, when we wish to know the value of a parameter or random variable given some data which maximizes the posterior, and the evidence probability is independent of this variable or parameter, then we can exclude the evidence probability from the calculations.

Expectation, generating functions and characteristic functions

There are many ways of *summarizing* the distribution of a random variable. Of particular importance are *measures of central tendency* such as the *mean* and *median*. The mean of a (continuous) random variable X is the sum over all possible outcomes weighted by the probability of that outcome:

$$E[X] = \int_{\Omega} x\, f(x)\, dx \tag{1.28}$$

Where not obvious from the context, we write $E_X[X]$ to indicate that this integral is with respect to the random variable X. In the case of discrete variables this is $E[X] = \sum_{a \in X(\Omega)} x\, f(x)$. As discussed earlier, expectation is a linear operator, i.e. $E\left[\sum_i a_i X_i\right] = \sum_i a_i E[X_i]$ for arbitrary constants a_i. A constant is invariant under expectation: $E[a] = a$. The mean is also known as the *expected value*, and the integral is called the *expectation*. The expectation plays a central role in probability and statistics, and can in fact be used to construct an entirely different axiomatic view on probability. The expectation with respect to an arbitrary transformation of a random variable, $g(X)$ is:

$$E[g(X)] = \int_{\Omega} g(x)\, f(x)\, dx \tag{1.29}$$

Using this we can define a hierarchy of summaries of the distribution of a random variable, known as the *k-th moments*:

$$E\left[X^k\right] = \int_{\Omega} x^k f(x)\, dx \tag{1.30}$$

From the unit measure property of probability it can be seen that the zeroth moment $E\left[X^0\right] = 1$. The first moment coincides with the mean. *Central moments* are those defined "around" the mean:

$$\mu_k = E\left[(X - E[X])^k\right] = \int_{\Omega} (x-\mu)^k f(x)\, dx \tag{1.31}$$

where $\mu$is the mean. A very import central moment is the variance,$\mathrm{var}[X] = \mu_2$, which is a *measure of spread* (about the mean) of the distribution. The *standard deviation* is the square root of this $\mathrm{std}[X] = \sqrt{\mu_2}$. Higher order central moments such as skewness (μ_3) and kurtosis (μ_4) measure

aspects such as the asymmetry and sharpness of distribution, respectively.

For joint distributions with joint density function $f(x, y)$, the expectation is:

$$E[g(X, Y)] = \int_{\Omega_Y} \int_{\Omega_X} g(x, y) f(x, y) \, dx \, dy \tag{1.32}$$

From this, we can derive the *joint moments*:

$$E\left[X^j Y^k\right] = \int_{\Omega_Y} \int_{\Omega_X} x^j y^k f(x, y) \, dx \, dy \tag{1.33}$$

An important special case is the *joint second central moment*, known as the *covariance*:

$$\begin{aligned} \operatorname{cov}[X, Y] &= E[(X - E[X])(Y - E[Y])] \\ &= \int_{\Omega_Y} \int_{\Omega_X} (x - \mu_X)(y - \mu_Y) f(x, y) \, dx \, dy \end{aligned} \tag{1.34}$$

where μ_X, μ_Y are the means of X, Y respectively.

Sometimes, the hierarchy of moments of a distribution serve to define the distribution uniquely. A very important kind of expectation is the *moment generating function* (MGF), for discrete variables:

$$M(s) = E[\exp(sX)] = \sum_{x \in X(\Omega)} \exp(sx) f(x) \, dx \tag{1.35}$$

The real variable s becomes the new independent variable replacing the discrete variable x. When the sum (1.35) converges absolutely, then the MGF exists and can be used to find all the moments for the distribution of X:

$$E\left[X^k\right] = \frac{d^k M}{dt^k}(0) \tag{1.36}$$

This can be shown to follow from the series expansion of the exponential function. Using the Bernoulli example above, the MGF is $M(s) = 1 - p + p \exp(s)$. Often, the distribution of a random variable has a simple form under the MGF that makes the task of manipulating random variables relatively easy. For example, given a linear combination of independent random variables:

$$X_N = \sum_{n=1}^{N} a_n X_n \tag{1.37}$$

it is not a trivial matter to calculate the distribution of X_N. However, the MGF of the sum is just:

$$M_{X_N}(s) = \prod_{n=1}^{N} M_{X_n}(a_n s) \tag{1.38}$$

from which the distribution of the sum can sometimes be recognized immediately. As an example, the MGF for an (unweighted) sum of N i.i.d. Bernoulli random variables with parameter p, is:

$$M_{X_N}(s) = (1 - p + p\exp(s))^N \tag{1.39}$$

which is just the MGF of the *binomial distribution*.

A similar expectation is the *characteristic function* (CF), for continuous variables:

$$\psi(s) = E[\exp(isX)] = \int_\Omega \exp(isx) f(x)\, dx \tag{1.40}$$

where $i = \sqrt{-1}$. This can be understood as the Fourier transform of the density function. An advantage over the MGF is that the CF always exists. It can therefore be used as an alternative way to define a distribution, a fact which is necessary for some well-known distributions such as the *Levy* or *alpha-stable* distributions. Well-known properties of Fourier transforms make it easy to use the CF to manipulate random variables. For example, given a random variable X with CF $\psi_X(s)$, the random variable $Y = X + m$ where m is a constant is:

$$\psi_Y(s) = \psi_X(s)\exp(ism) \tag{1.41}$$

From this, given that the CF of the standard normal Gaussian with mean zero and unit variance, is $\exp\left(-\frac{1}{2}s^2\right)$, the shifted random variable Y has CF $\psi_Y(s) = \exp\left(ism - \frac{1}{2}s^2\right)$. Another property, similar to the MGF, is the linear combination property:

$$\psi_{X_N}(s) = \prod_{i=1}^{N} \psi_{X_i}(a_i s) \tag{1.42}$$

We can use this to show that for a linear combination (1.37) of independent Gaussian random variables with mean $\mu_n$and variance σ_n^2, the CF of the sum is:

$$\psi_{X_N}(s) = \exp\left(is\sum_{n=1}^{N} a_n\mu_n - \frac{1}{2}s^2\sum_{n=1}^{N} a_n^2\sigma_n^2\right) \tag{1.43}$$

which can be recognized as another Gaussian with mean $\sum_{n=1}^{N} a_n\mu_n$ and variance $\sum_{n=1}^{N} a_n^2\sigma_n^2$. This shows that the Gaussian is *invariant* to linear transformations, a property known as (*statistical*) *stability*, which is of fundamental importance in classical statistical DSP.

Empirical distribution function and sample expectations

If we start with a PDF or PMF, then the specific values of the parameters of these functions determine the mathematical form of the distribution. However, often we want some given data to "speak for itself" and determine a distribution function directly. An important, and simple way