

Bayesian Reasoning and Machine Learning

Extracting value from vast amounts of data presents a major challenge to all those working in computer science and related fields. Machine learning technology is already used to help with this task in a wide range of industrial applications, including search engines, DNA sequencing, stock market analysis and robot locomotion. As its usage becomes more widespread, the skills taught in this book will be invaluable to students.

Designed for final-year undergraduate and graduate students, this gentle introduction is ideally suited to readers without a solid background in linear algebra and calculus. It covers basic probabilistic reasoning to advanced techniques in machine learning, and crucially enables students to construct their own models for real-world problems by teaching them what lies behind the methods. A central conceptual theme is the use of Bayesian modelling to describe and build inference algorithms. Numerous examples and exercises are included in the text. Comprehensive resources for students and instructors are available online.

Cambridge University Press
978-0-521-51814-7 - Bayesian Reasoning and Machine Learning
David Barber
Frontmatter
[More information](#)

Cambridge University Press
978-0-521-51814-7 - Bayesian Reasoning and Machine Learning
David Barber
Frontmatter
[More information](#)

Bayesian Reasoning and Machine Learning

David Barber
University College London



Cambridge University Press
978-0-521-51814-7 - Bayesian Reasoning and Machine Learning
David Barber
Frontmatter
[More information](#)

CAMBRIDGE UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning and research at the highest international levels of excellence.

www.cambridge.org

Information on this title: www.cambridge.org/9780521518147

© D. Barber 2012

This publication is in copyright. Subject to statutory exception and to the provisions of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press.

First published 2012

6th printing 2015

Printed in the United Kingdom by TJ International Ltd, Padstow, Cornwall

A catalogue record for this publication is available from the British Library

Library of Congress Cataloguing in Publication data

Barber, David, 1968–

Bayesian reasoning and machine learning / David Barber.

p. cm.

Includes bibliographical references and index.

ISBN 978-0-521-51814-7

1. Machine learning. 2. Bayesian statistical decision theory. I. Title.

QA267.B347 2012

006.3'1 – dc23 2011035553

ISBN 978-0-521-51814-7 Hardback

Additional resources for this publication at www.cambridge.org/brml and at www.cs.ucl.ac.uk/staff/D.Barber/brml

Cambridge University Press has no responsibility for the persistence or accuracy of URLs for external or third-party internet websites referred to in this publication, and does not guarantee that any content on such websites is, or will remain, accurate or appropriate.

CONTENTS

Preface	xv		
List of notation	xx		
BRMLTOOLBOX	xxi		
I Inference in probabilistic models			
1 Probabilistic reasoning	3		
1.1 Probability refresher			
1.1.1 Interpreting conditional probability			
1.1.2 Probability tables			
1.2 Probabilistic reasoning			
1.3 Prior, likelihood and posterior			
1.3.1 Two dice: what were the individual scores?			
1.4 Summary			
1.5 Code			
1.6 Exercises			
2 Basic graph concepts	22		
2.1 Graphs			
2.2 Numerically encoding graphs			
2.2.1 Edge list			
2.2.2 Adjacency matrix			
2.2.3 Clique matrix			
2.3 Summary			
2.4 Code			
2.5 Exercises			
3 Belief networks	29		
3.1 The benefits of structure			
3.1.1 Modelling independencies			
3.1.2 Reducing the burden of specification			
3.2 Uncertain and unreliable evidence			
3.2.1 Uncertain evidence			
3.2.2 Unreliable evidence			
3.3 Belief networks			
3.3.1 Conditional independence			
3.3.2 The impact of collisions			
3.3.3 Graphical path manipulations for independence			
3.3.4 d-separation			
3.3.5 Graphical and distributional in/dependence			
3.3.6 Markov equivalence in belief networks			
3.3.7 Belief networks have limited expressibility			
3.4 Causality			
3.4.1 Simpson's paradox			
3.4.2 The do-calculus			
3.4.3 Influence diagrams and the do-calculus			
3.5 Summary			
3.6 Code			
3.7 Exercises			
4 Graphical models	58		
4.1 Graphical models			
4.2 Markov networks			
4.2.1 Markov properties			
4.2.2 Markov random fields			
4.2.3 Hammersley–Clifford theorem			
4.2.4 Conditional independence using Markov networks			
4.2.5 Lattice models			
4.3 Chain graphical models			
4.4 Factor graphs			
4.4.1 Conditional independence in factor graphs			
4.5 Expressiveness of graphical models			
4.6 Summary			
4.7 Code			
4.8 Exercises			

5	Efficient inference in trees	77	
5.1	Marginal inference		
5.1.1	Variable elimination in a Markov chain and message passing		
5.1.2	The sum-product algorithm on factor graphs		
5.1.3	Dealing with evidence		
5.1.4	Computing the marginal likelihood		
5.1.5	The problem with loops		
5.2	Other forms of inference		
5.2.1	Max-product		
5.2.2	Finding the N most probable states		
5.2.3	Most probable path and shortest path		
5.2.4	Mixed inference		
5.3	Inference in multiply connected graphs		
5.3.1	Bucket elimination		
5.3.2	Loop-cut conditioning		
5.4	Message passing for continuous distributions		
5.5	Summary		
5.6	Code		
5.7	Exercises		
6	The junction tree algorithm	102	
6.1	Clustering variables		
6.1.1	Reparameterisation		
6.2	Clique graphs		
6.2.1	Absorption		
6.2.2	Absorption schedule on clique trees		
6.3	Junction trees		
6.3.1	The running intersection property		
6.4	Constructing a junction tree for singly connected distributions		
6.4.1	Moralisation		
6.4.2	Forming the clique graph		
6.4.3	Forming a junction tree from a clique graph		
6.4.4	Assigning potentials to cliques		
6.5	Junction trees for multiply connected distributions		
6.5.1	Triangulation algorithms		
6.6	The junction tree algorithm		
6.6.1	Remarks on the JTA		
6.6.2	Computing the normalisation constant of a distribution		
6.6.3	The marginal likelihood		
6.6.4	Some small JTA examples		
6.6.5	Shafer–Shenoy propagation		
6.7	Finding the most likely state		
6.8	Reabsorption: converting a junction tree to a directed network		
6.9	The need for approximations		
6.9.1	Bounded width junction trees		
6.10	Summary		
6.11	Code		
6.12	Exercises		
7	Making decisions	127	
7.1	Expected utility		
7.1.1	Utility of money		
7.2	Decision trees		
7.3	Extending Bayesian networks for decisions		
7.3.1	Syntax of influence diagrams		
7.4	Solving influence diagrams		
7.4.1	Messages on an ID		
7.4.2	Using a junction tree		
7.5	Markov decision processes		
7.5.1	Maximising expected utility by message passing		
7.5.2	Bellman’s equation		
7.6	Temporally unbounded MDPs		
7.6.1	Value iteration		
7.6.2	Policy iteration		
7.6.3	A curse of dimensionality		
7.7	Variational inference and planning		
7.8	Financial matters		
7.8.1	Options pricing and expected utility		
7.8.2	Binomial options pricing model		
7.8.3	Optimal investment		
7.9	Further topics		
7.9.1	Partially observable MDPs		
7.9.2	Reinforcement learning		
7.10	Summary		
7.11	Code		
7.12	Exercises		

II Learning in probabilistic models

8 Statistics for machine learning 165

- 8.1 Representing data
 - 8.1.1 Categorical
 - 8.1.2 Ordinal
 - 8.1.3 Numerical
- 8.2 Distributions
 - 8.2.1 The Kullback–Leibler divergence $KL(q|p)$
 - 8.2.2 Entropy and information
- 8.3 Classical distributions
- 8.4 Multivariate Gaussian
 - 8.4.1 Completing the square
 - 8.4.2 Conditioning as system reversal
 - 8.4.3 Whitening and centring
- 8.5 Exponential family
 - 8.5.1 Conjugate priors
- 8.6 Learning distributions
- 8.7 Properties of maximum likelihood
 - 8.7.1 Training assuming the correct model class
 - 8.7.2 Training when the assumed model is incorrect
 - 8.7.3 Maximum likelihood and the empirical distribution
- 8.8 Learning a Gaussian
 - 8.8.1 Maximum likelihood training
 - 8.8.2 Bayesian inference of the mean and variance
 - 8.8.3 Gauss-gamma distribution
- 8.9 Summary
- 8.10 Code
- 8.11 Exercises

9 Learning as inference 199

- 9.1 Learning as inference
 - 9.1.1 Learning the bias of a coin
 - 9.1.2 Making decisions
 - 9.1.3 A continuum of parameters
 - 9.1.4 Decisions based on continuous intervals
- 9.2 Bayesian methods and ML-II
- 9.3 Maximum likelihood training of belief networks
- 9.4 Bayesian belief network training
 - 9.4.1 Global and local parameter independence

- 9.4.2 Learning binary variable tables using a Beta prior
- 9.4.3 Learning multivariate discrete tables using a Dirichlet prior
- 9.5 Structure learning
 - 9.5.1 PC algorithm
 - 9.5.2 Empirical independence
 - 9.5.3 Network scoring
 - 9.5.4 Chow–Liu trees
- 9.6 Maximum likelihood for undirected models
 - 9.6.1 The likelihood gradient
 - 9.6.2 General tabular clique potentials
 - 9.6.3 Decomposable Markov networks
 - 9.6.4 Exponential form potentials
 - 9.6.5 Conditional random fields
 - 9.6.6 Pseudo likelihood
 - 9.6.7 Learning the structure
- 9.7 Summary
- 9.8 Code
- 9.9 Exercises

10 Naive Bayes 243

- 10.1 Naive Bayes and conditional independence
- 10.2 Estimation using maximum likelihood
 - 10.2.1 Binary attributes
 - 10.2.2 Multi-state variables
 - 10.2.3 Text classification
- 10.3 Bayesian naive Bayes
- 10.4 Tree augmented naive Bayes
 - 10.4.1 Learning tree augmented naive Bayes networks
- 10.5 Summary
- 10.6 Code
- 10.7 Exercises

11 Learning with hidden variables 256

- 11.1 Hidden variables and missing data
 - 11.1.1 Why hidden/missing variables can complicate proceedings
 - 11.1.2 The missing at random assumption

15.3 High-dimensional data		17.4.1 Logistic regression	
15.3.1 Eigen-decomposition for $N < D$		17.4.2 Beyond first-order gradient ascent	
15.3.2 PCA via singular value decomposition		17.4.3 Avoiding overconfident classification	
15.4 Latent semantic analysis		17.4.4 Multiple classes	
15.4.1 Information retrieval		17.4.5 The kernel trick for classification	
15.5 PCA with missing data		17.5 Support vector machines	
15.5.1 Finding the principal directions		17.5.1 Maximum margin linear classifier	
15.5.2 Collaborative filtering using PCA with missing data		17.5.2 Using kernels	
15.6 Matrix decomposition methods		17.5.3 Performing the optimisation	
15.6.1 Probabilistic latent semantic analysis		17.5.4 Probabilistic interpretation	
15.6.2 Extensions and variations		17.6 Soft zero-one loss for outlier robustness	
15.6.3 Applications of PLSA/NMF		17.7 Summary	
15.7 Kernel PCA		17.8 Code	
15.8 Canonical correlation analysis		17.9 Exercises	
15.8.1 SVD formulation			
15.9 Summary		18 Bayesian linear models	392
15.10 Code		18.1 Regression with additive Gaussian noise	
15.11 Exercises		18.1.1 Bayesian linear parameter models	
16 Supervised linear dimension reduction	359	18.1.2 Determining hyperparameters: ML-II	
16.1 Supervised linear projections		18.1.3 Learning the hyperparameters using EM	
16.2 Fisher's linear discriminant		18.1.4 Hyperparameter optimisation: using the gradient	
16.3 Canonical variates		18.1.5 Validation likelihood	
16.3.1 Dealing with the nullspace		18.1.6 Prediction and model averaging	
16.4 Summary		18.1.7 Sparse linear models	
16.5 Code		18.2 Classification	
16.6 Exercises		18.2.1 Hyperparameter optimisation	
17 Linear models	367	18.2.2 Laplace approximation	
17.1 Introduction: fitting a straight line		18.2.3 Variational Gaussian approximation	
17.2 Linear parameter models for regression		18.2.4 Local variational approximation	
17.2.1 Vector outputs		18.2.5 Relevance vector machine for classification	
17.2.2 Regularisation		18.2.6 Multi-class case	
17.2.3 Radial basis functions		18.3 Summary	
17.3 The dual representation and kernels		18.4 Code	
17.3.1 Regression in the dual space		18.5 Exercises	
17.4 Linear parameter models for classification			

22.3	Summary	
22.4	Code	
22.5	Exercises	
IV	Dynamical models	
23	Discrete-state Markov models	489
23.1	Markov models	
23.1.1	Equilibrium and stationary distribution of a Markov chain	
23.1.2	Fitting Markov models	
23.1.3	Mixture of Markov models	
23.2	Hidden Markov models	
23.2.1	The classical inference problems	
23.2.2	Filtering $p(h_t v_{1:t})$	
23.2.3	Parallel smoothing $p(h_t v_{1:T})$	
23.2.4	Correction smoothing	
23.2.5	Sampling from $p(h_{1:T} v_{1:T})$	
23.2.6	Most likely joint state	
23.2.7	Prediction	
23.2.8	Self-localisation and kidnapped robots	
23.2.9	Natural language models	
23.3	Learning HMMs	
23.3.1	EM algorithm	
23.3.2	Mixture emission	
23.3.3	The HMM-GMM	
23.3.4	Discriminative training	
23.4	Related models	
23.4.1	Explicit duration model	
23.4.2	Input–output HMM	
23.4.3	Linear chain CRFs	
23.4.4	Dynamic Bayesian networks	
23.5	Applications	
23.5.1	Object tracking	
23.5.2	Automatic speech recognition	
23.5.3	Bioinformatics	
23.5.4	Part-of-speech tagging	
23.6	Summary	
23.7	Code	
23.8	Exercises	
24	Continuous-state Markov models	520
24.1	Observed linear dynamical systems	
24.1.1	Stationary distribution with noise	
24.2	Auto-regressive models	
24.2.1	Training an AR model	
24.2.2	AR model as an OLDS	
24.2.3	Time-varying AR model	
24.2.4	Time-varying variance AR models	
24.3	Latent linear dynamical systems	
24.4	Inference	
24.4.1	Filtering	
24.4.2	Smoothing: Rauch–Tung–Striebel correction method	
24.4.3	The likelihood	
24.4.4	Most likely state	
24.4.5	Time independence and Riccati equations	
24.5	Learning linear dynamical systems	
24.5.1	Identifiability issues	
24.5.2	EM algorithm	
24.5.3	Subspace methods	
24.5.4	Structured LDSs	
24.5.5	Bayesian LDSs	
24.6	Switching auto-regressive models	
24.6.1	Inference	
24.6.2	Maximum likelihood learning using EM	
24.7	Summary	
24.8	Code	
24.9	Exercises	
25	Switching linear dynamical systems	547
25.1	Introduction	
25.2	The switching LDS	
25.2.1	Exact inference is computationally intractable	
25.3	Gaussian sum filtering	
25.3.1	Continuous filtering	
25.3.2	Discrete filtering	
25.3.3	The likelihood $p(\mathbf{v}_{1:T})$	
25.3.4	Collapsing Gaussians	
25.3.5	Relation to other methods	
25.4	Gaussian sum smoothing	
25.4.1	Continuous smoothing	
25.4.2	Discrete smoothing	
25.4.3	Collapsing the mixture	
25.4.4	Using mixtures in smoothing	
25.4.5	Relation to other methods	

25.5	Reset models		27.4.1	Markov chains	
25.5.1	A Poisson reset model		27.4.2	Metropolis–Hastings sampling	
25.5.2	Reset-HMM-LDS		27.5	Auxiliary variable methods	
25.6	Summary		27.5.1	Hybrid Monte Carlo (HMC)	
25.7	Code		27.5.2	Swendson–Wang (SW)	
25.8	Exercises		27.5.3	Slice sampling	
26	Distributed computation	568	27.6	Importance sampling	
26.1	Introduction		27.6.1	Sequential importance sampling	
26.2	Stochastic Hopfield networks		27.6.2	Particle filtering as an approximate forward pass	
26.3	Learning sequences		27.7	Summary	
26.3.1	A single sequence		27.8	Code	
26.3.2	Multiple sequences		27.9	Exercises	
26.3.3	Boolean networks		28	Deterministic approximate inference	617
26.3.4	Sequence disambiguation		28.1	Introduction	
26.4	Tractable continuous latent variable models		28.2	The Laplace approximation	
26.4.1	Deterministic latent variables		28.3	Properties of Kullback–Leibler variational inference	
26.4.2	An augmented Hopfield network		28.3.1	Bounding the normalisation constant	
26.5	Neural models		28.3.2	Bounding the marginal likelihood	
26.5.1	Stochastically spiking neurons		28.3.3	Bounding marginal quantities	
26.5.2	Hopfield membrane potential		28.3.4	Gaussian approximations using KL divergence	
26.5.3	Dynamic synapses		28.3.5	Marginal and moment matching properties of minimising $KL(p q)$	
26.5.4	Leaky integrate and fire models		28.4	Variational bounding using $KL(q p)$	
26.6	Summary		28.4.1	Pairwise Markov random field	
26.7	Code		28.4.2	General mean-field equations	
26.8	Exercises		28.4.3	Asynchronous updating guarantees approximation improvement	
V	Approximate inference		28.4.4	Structured variational approximation	
27	Sampling	587	28.5	Local and KL variational approximations	
27.1	Introduction		28.5.1	Local approximation	
27.1.1	Univariate sampling		28.5.2	KL variational approximation	
27.1.2	Rejection sampling		28.6	Mutual information maximisation: a KL variational approach	
27.1.3	Multivariate sampling				
27.2	Ancestral sampling				
27.2.1	Dealing with evidence				
27.2.2	Perfect sampling for a Markov network				
27.3	Gibbs sampling				
27.3.1	Gibbs sampling as a Markov chain				
27.3.2	Structured Gibbs sampling				
27.3.3	Remarks				
27.4	Markov chain Monte Carlo (MCMC)				

Contents

xiii

28.6.1	The information maximisation algorithm	28.12	Code	
28.6.2	Linear Gaussian decoder	28.13	Exercises	
28.7	Loopy belief propagation	Appendix A: Background mathematics	655	
28.7.1	Classical BP on an undirected graph	A.1	Linear algebra	
28.7.2	Loopy BP as a variational procedure	A.2	Multivariate calculus	
28.8	Expectation propagation	A.3	Inequalities	
28.9	MAP for Markov networks	A.4	Optimisation	
28.9.1	Pairwise Markov networks	A.5	Multivariate optimisation	
28.9.2	Attractive binary Markov networks	A.6	Constrained optimisation using Lagrange multipliers	
28.9.3	Potts model	References		675
28.10	Further reading	Index		689
28.11	Summary			
			<i>Colour plate section between pp. 360 and 361</i>	

Cambridge University Press
978-0-521-51814-7 - Bayesian Reasoning and Machine Learning
David Barber
Frontmatter
[More information](#)

PREFACE

The data explosion

We live in a world that is rich in data, ever increasing in scale. This data comes from many different sources in science (bioinformatics, astronomy, physics, environmental monitoring) and commerce (customer databases, financial transactions, engine monitoring, speech recognition, surveillance, search). Possessing the knowledge as to how to process and extract value from such data is therefore a key and increasingly important skill. Our society also expects ultimately to be able to engage with computers in a natural manner so that computers can ‘talk’ to humans, ‘understand’ what they say and ‘comprehend’ the visual world around them. These are difficult large-scale information processing tasks and represent grand challenges for computer science and related fields. Similarly, there is a desire to control increasingly complex systems, possibly containing many interacting parts, such as in robotics and autonomous navigation. Successfully mastering such systems requires an understanding of the processes underlying their behaviour. Processing and making sense of such large amounts of data from complex systems is therefore a pressing modern-day concern and will likely remain so for the foreseeable future.

Machine learning

Machine learning is the study of data-driven methods capable of mimicking, understanding and aiding human and biological information processing tasks. In this pursuit, many related issues arise such as how to compress data, interpret and process it. Often these methods are not necessarily directed to mimicking directly human processing but rather to enhancing it, such as in predicting the stock market or retrieving information rapidly. In this probability theory is key since inevitably our limited data and understanding of the problem forces us to address uncertainty. In the broadest sense, machine learning and related fields aim to ‘learn something useful’ about the environment within which the agent operates. Machine learning is also closely allied with artificial intelligence, with machine learning placing more emphasis on using data to drive and adapt the model.

In the early stages of machine learning and related areas, similar techniques were discovered in relatively isolated research communities. This book presents a unified treatment via graphical models, a marriage between graph and probability theory, facilitating the transference of machine learning concepts between different branches of the mathematical and computational sciences.

Whom this book is for

The book is designed to appeal to students with only a modest mathematical background in undergraduate calculus and linear algebra. No formal computer science or statistical background is required to follow the book, although a basic familiarity with probability, calculus and linear algebra

would be useful. The book should appeal to students from a variety of backgrounds, including computer science, engineering, applied statistics, physics and bioinformatics that wish to gain an entry to probabilistic approaches in machine learning. In order to engage with students, the book introduces fundamental concepts in inference using only minimal reference to algebra and calculus. More mathematical techniques are postponed until as and when required, always with the concept as primary and the mathematics secondary.

The concepts and algorithms are described with the aid of many worked examples. The exercises and demonstrations, together with an accompanying MATLAB toolbox, enable the reader to experiment and more deeply understand the material. The ultimate aim of the book is to enable the reader to construct novel algorithms. The book therefore places an emphasis on skill learning, rather than being a collection of recipes. This is a key aspect since modern applications are often so specialised as to require novel methods. The approach taken throughout is to describe the problem as a graphical model, which is then translated into a mathematical framework, ultimately leading to an algorithmic implementation in the BRMLTOOLBOX.

The book is primarily aimed at final year undergraduates and graduates without significant experience in mathematics. On completion, the reader should have a good understanding of the techniques, practicalities and philosophies of probabilistic aspects of machine learning and be well equipped to understand more advanced research level material.

The structure of the book

The book begins with the basic concepts of graphical models and inference. For the independent reader Chapters 1, 2, 3, 4, 5, 9, 10, 13, 14, 15, 16, 17, 21 and 23 would form a good introduction to probabilistic reasoning, modelling and machine learning. The material in Chapters 19, 24, 25 and 28 is more advanced, with the remaining material being of more specialised interest. Note that in each chapter the level of material is of varying difficulty, typically with the more challenging material placed towards the end of each chapter. As an introduction to the area of probabilistic modelling, a course can be constructed from the material as indicated in the chart.

The material from Parts I and II has been successfully used for courses on graphical models. I have also taught an introduction to probabilistic machine learning using material largely from Part III, as indicated. These two courses can be taught separately and a useful approach would be to teach first the graphical models course, followed by a separate probabilistic machine learning course.

A short course on approximate inference can be constructed from introductory material in Part I and the more advanced material in Part V, as indicated. The exact inference methods in Part I can be covered relatively quickly with the material in Part V considered in more depth.

A timeseries course can be made by using primarily the material in Part IV, possibly combined with material from Part I for students that are unfamiliar with probabilistic modelling approaches. Some of this material, particularly in Chapter 25, is more advanced and can be deferred until the end of the course, or considered for a more advanced course.

The references are generally to works at a level consistent with the book material and which are in the most part readily available.

		Graphical models course	Probabilistic machine learning course	Approximate inference short course	Timeseries short course	Probabilistic modelling course
Part I: Inference in probabilistic models	1: Probabilistic reasoning	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
	2: Basic graph concepts	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
	3: Belief networks	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
	4: Graphical models	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
	5: Efficient inference in trees	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
	6: The junction tree algorithm	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	7: Making decisions	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Part II: Learning in probabilistic models	8: Statistics for machine learning	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
	9: Learning as inference	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
	10: Naive Bayes	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	11: Learning with hidden variables	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
	12: Bayesian model selection	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Part III: Machine learning	13: Machine learning concepts	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	14: Nearest neighbour classification	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	15: Unsupervised linear dimension reduction	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	16: Supervised linear dimension reduction	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	17: Linear models	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	18: Bayesian linear models	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
	19: Gaussian processes	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	20: Mixture models	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
	21: Latent linear models	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
	22: Latent ability models	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Part IV: Dynamical models	23: Discrete-state Markov models	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
	24: Continuous-state Markov models	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
	25: Switching linear dynamical systems	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
	26: Distributed computation	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Part V: Approximate inference	27: Sampling	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
	28: Deterministic approximate inference	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

Accompanying code

The BRMLTOOLBOX is provided to help readers see how mathematical models translate into actual MATLAB code. There is a large number of demos that a lecturer may wish to use or adapt to help illustrate the material. In addition many of the exercises make use of the code, helping the reader gain confidence in the concepts and their application. Along with complete routines for many machine learning methods, the philosophy is to provide low-level routines whose composition intuitively follows the mathematical description of the algorithm. In this way students may easily match the mathematics with the corresponding algorithmic implementation.

Website

The BRMLTOOLBOX along with an electronic version of the book is available from

www.cs.ucl.ac.uk/staff/D.Barber/brml

Instructors seeking solutions to the exercises can find information at www.cambridge.org/brml, along with additional teaching materials.

Other books in this area

The literature on machine learning is vast with much relevant literature also contained in statistics, engineering and other physical sciences. A small list of more specialised books that may be referred to for deeper treatments of specific topics is:

- Graphical models
 - *Graphical Models* by S. Lauritzen, Oxford University Press, 1996.
 - *Bayesian Networks and Decision Graphs* by F. Jensen and T. D. Nielsen, Springer-Verlag, 2007.
 - *Probabilistic Networks and Expert Systems* by R. G. Cowell, A. P. Dawid, S. L. Lauritzen and D. J. Spiegelhalter, Springer-Verlag, 1999.
 - *Probabilistic Reasoning in Intelligent Systems* by J. Pearl, Morgan Kaufmann, 1988.
 - *Graphical Models in Applied Multivariate Statistics* by J. Whittaker, Wiley, 1990.
 - *Probabilistic Graphical Models: Principles and Techniques* by D. Koller and N. Friedman, MIT Press, 2009.
- Machine learning and information processing
 - *Information Theory, Inference and Learning Algorithms* by D. J. C. MacKay, Cambridge University Press, 2003.
 - *Pattern Recognition and Machine Learning* by C. M. Bishop, Springer-Verlag, 2006.
 - *An Introduction to Support Vector Machines*, N. Cristianini and J. Shawe-Taylor, Cambridge University Press, 2000.
 - *Gaussian Processes for Machine Learning* by C. E. Rasmussen and C. K. I. Williams, MIT Press, 2006.

Acknowledgements

Many people have helped this book along the way either in terms of reading, feedback, general insights, allowing me to present their work, or just plain motivation. Amongst these I would like

to thank Dan Cornford, Massimiliano Pontil, Mark Herbster, John Shawe-Taylor, Vladimir Kolmogorov, Yuri Boykov, Tom Minka, Simon Prince, Silvia Chiappa, Bertrand Mesot, Robert Cowell, Ali Taylan Cemgil, David Blei, Jeff Bilmes, David Cohn, David Page, Peter Sollich, Chris Williams, Marc Toussaint, Amos Storkey, Zakria Hussain, Le Chen, Serafín Moral, Milan Studený, Luc De Raedt, Tristan Fletcher, Chris Vryonides, Tom Furnston, Ed Challis and Chris Bracegirdle. I would also like to thank the many students that have helped improve the material during lectures over the years. I'm particularly grateful to Taylan Cemgil for allowing his GraphLayout package to be bundled with the BRMLTOOLBOX.

The staff at Cambridge University Press have been a delight to work with and I would especially like to thank Heather Bergman for her initial endeavours and the wonderful Diana Gillooly for her continued enthusiasm.

A heartfelt thankyou to my parents and sister – I hope this small token will make them proud. I'm also fortunate to be able to acknowledge the support and generosity of friends throughout. Finally, I'd like to thank Silvia who made it all worthwhile.

NOTATION

\mathcal{V}	A calligraphic symbol typically denotes a set of random variables	page 3
$\text{dom}(x)$	Domain of a variable	3
$x = x$	The variable x is in the state x	3
$p(x = \text{tr})$	Probability of event/variable x being in the state true	3
$p(x = \text{fa})$	Probability of event/variable x being in the state false	3
$p(x, y)$	Probability of x and y	4
$p(x \cap y)$	Probability of x and y	4
$p(x \cup y)$	Probability of x or y	4
$p(x y)$	The probability of x conditioned on y	4
$\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \mathcal{Z}$	Variables \mathcal{X} are independent of variables \mathcal{Y} conditioned on variables \mathcal{Z}	7
$\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \mathcal{Z}$	Variables \mathcal{X} are dependent on variables \mathcal{Y} conditioned on variables \mathcal{Z}	7
$\int_x f(x)$	For continuous variables this is shorthand for $\int_x f(x)dx$ and for discrete variables means summation over the states of x , $\sum_x f(x)$	14
$\mathbb{I}[S]$	Indicator : has value 1 if the statement S is true, 0 otherwise	16
$\text{pa}(x)$	The parents of node x	24
$\text{ch}(x)$	The children of node x	24
$\text{ne}(x)$	Neighbours of node x	24
$\text{dim}(x)$	For a discrete variable x , this denotes the number of states x can take	34
$\langle f(x) \rangle_{p(x)}$	The average of the function $f(x)$ with respect to the distribution $p(x)$	170
$\delta(a, b)$	Delta function. For discrete a, b , this is the Kronecker delta, $\delta_{a,b}$ and for continuous a, b the Dirac delta function $\delta(a - b)$	172
$\text{dim}(\mathbf{x})$	The dimension of the vector/matrix \mathbf{x}	183
$\#(x = s, y = t)$	The number of times x is in state s and y in state t simultaneously	207
$\#_y^x$	The number of times variable x is in state y	293
\mathcal{D}	Dataset	303
n	Data index	303
N	Number of dataset training points	303
\mathbf{S}	Sample Covariance matrix	331
$\sigma(x)$	The logistic sigmoid $1/(1 + \exp(-x))$	371
$\text{erf}(x)$	The (Gaussian) error function	372
$x_{a:b}$	x_a, x_{a+1}, \dots, x_b	372
$i \sim j$	The set of unique neighbouring edges on a graph	624
\mathbf{I}_m	The $m \times m$ identity matrix	644

BRMLTOOLBOX

The BRMLTOOLBOX is a lightweight set of routines that enables the reader to experiment with concepts in graph theory, probability theory and machine learning. The code contains basic routines for manipulating discrete variable distributions, along with more limited support for continuous variables. In addition there are many hard-coded standard machine learning algorithms. The website contains also a complete list of all the teaching demos and related exercise material.

BRMLTOOLKIT

Graph theory

<code>ancestors</code>	- Return the ancestors of nodes x in DAG A
<code>ancestralorder</code>	- Return the ancestral order of the DAG A (oldest first)
<code>descendants</code>	- Return the descendants of nodes x in DAG A
<code>children</code>	- Return the children of variable x given adjacency matrix A
<code>edges</code>	- Return edge list from adjacency matrix A
<code>elimtri</code>	- Return a variable elimination sequence for a triangulated graph
<code>connectedComponents</code>	- Find the connected components of an adjacency matrix
<code>istree</code>	- Check if graph is singly connected
<code>neigh</code>	- Find the neighbours of vertex v on a graph with adjacency matrix G
<code>noselfpath</code>	- Return a path excluding self-transitions
<code>parents</code>	- Return the parents of variable x given adjacency matrix A
<code>spantree</code>	- Find a spanning tree from an edge list
<code>triangulate</code>	- Triangulate adjacency matrix A
<code>triangulatePorder</code>	- Triangulate adjacency matrix A according to a partial ordering

Potential manipulation

<code>condpot</code>	- Return a potential conditioned on another variable
<code>changevar</code>	- Change variable names in a potential
<code>dag</code>	- Return the adjacency matrix (zeros on diagonal) for a belief network
<code>deltapot</code>	- A delta function potential
<code>disptable</code>	- Print the table of a potential
<code>divpots</code>	- Divide potential pot_a by pot_b
<code>drawFG</code>	- Draw the factor graph A
<code>drawID</code>	- Plot an influence diagram
<code>drawJTree</code>	- Plot a junction tree
<code>drawNet</code>	- Plot network
<code>evalpot</code>	- Evaluate the table of a potential when variables are set
<code>exppot</code>	- Exponential of a potential
<code>eyepot</code>	- Return a unit potential
<code>grouppot</code>	- Form a potential based on grouping variables together
<code>groupstate</code>	- Find the state of the group variables corresponding to a given ungrouped state
<code>logpot</code>	- Logarithm of the potential
<code>markov</code>	- Return a symmetric adjacency matrix of Markov network in pot
<code>maxpot</code>	- Maximise a potential over variables
<code>maxsumpot</code>	- Maximise or sum a potential over variables
<code>multpots</code>	- Multiply potentials into a single potential

numstates	- Number of states of the variables in a potential
orderpot	- Return potential with variables reordered according to order
orderpotfields	- Order the fields of the potential, creating blank entries where necessary
potsample	- Draw sample from a single potential
potscontainingonly	- Returns those potential numbers that contain only the required variables
potvariables	- Returns information about all variables in a set of potentials
setevpot	- Sets variables in a potential into evidential states
setpot	- Sets potential variables to specified states
setstate	- Set a potential's specified joint state to a specified value
squeezepots	- Eliminate redundant potentials (those contained wholly within another)
sumpot	- Sum potential pot over variables
sumpotID	- Return the summed probability and utility tables from an ID
sumpots	- Sum a set of potentials
table	- Return the potential table
ungrouppot	- Form a potential based on ungrouping variables
uniquepots	- Eliminate redundant potentials (those contained wholly within another)
whichpot	- Returns potentials that contain a set of variables

Routines also extend the toolbox to deal with Gaussian potentials: `multpotsGaussianMoment.m`, `sumpotGaussianCanonical.m`, `sumpotGaussianMoment.m`, `multpotsGaussianCanonical.m` See `demoSumprodGaussCanon.m`, `demoSumprodGaussCanonLDS.m`, `demoSumprodGaussMoment.m`

Inference

absorb	- Update potentials in absorption message passing on a junction tree
absorption	- Perform full round of absorption on a junction tree
absorptionID	- Perform full round of absorption on an influence diagram
ancestralsample	- Ancestral sampling from a belief network
binaryMRFmap	- Get the MAP assignment for a binary MRF with positive W
bucketelim	- Bucket elimination on a set of potentials
condindep	- Conditional independence check using graph of variable interactions
condindepEmp	- Compute the empirical log Bayes factor and MI for independence/dependence
condindepPot	- Numerical conditional independence measure
condMI	- Conditional mutual information $I(x,y z)$ of a potential
FactorConnectingVariable	- Factor nodes connecting to a set of variables
FactorGraph	- Returns a factor graph adjacency matrix based on potentials
IDvars	- Probability and decision variables from a partial order
jtassignpot	- Assign potentials to cliques in a junction tree
jtrees	- Setup a junction tree based on a set of potentials
jtreesID	- Setup a junction tree based on an influence diagram
LoopyBP	- Loopy belief propagation using sum-product algorithm
MaxFlow	- Ford Fulkerson max-flow min-cut algorithm (breadth first search)
maxNpot	- Find the N most probable values and states in a potential
maxNprodFG	- N-max-product algorithm on a factor graph (returns the Nmax most probable states)
maxprodFG	- Max-product algorithm on a factor graph
MDPemDeterministicPolicy	- Solve MDP using EM with deterministic policy
MDPsolve	- Solve a Markov decision process
MesstoFact	- Returns the message numbers that connect into factor potential
metropolis	- Metropolis sample
mostprobablepath	- Find the most probable path in a Markov chain
mostprobablepathmult	- Find the all source all sink most probable paths in a Markov chain
sumprodFG	- Sum-product algorithm on a factor graph represented by A

Specific models

ARlds	- Learn AR coefficients using a linear dynamical system
ARtrain	- Fit auto-regressive (AR) coefficients of order L to v.
BayesLinReg	- Bayesian linear regression training using basis functions $\phi(x)$
BayesLogRegressionRVM	- Bayesian logistic regression with the relevance vector machine
CanonVar	- Canonical variates (no post rotation of variates)

BRMLTOOLBOX

xxiii

cca	- Canonical correlation analysis
covfnGE	- Gamma exponential covariance function
FA	- Factor analysis
GMMem	- Fit a mixture of Gaussian to the data X using EM
GPclass	- Gaussian process binary classification
GPreg	- Gaussian process regression
HebbML	- Learn a sequence for a Hopfield network
HMMbackward	- HMM backward pass
HMMbackwardSAR	- Backward pass (beta method) for the switching Auto-regressive HMM
HMMem	- EM algorithm for HMM
HMMforward	- HMM forward pass
HMMforwardSAR	- Switching auto-regressive HMM with switches updated only every Tskip timesteps
HMMgamma	- HMM posterior smoothing using the Rauch–Tung–Striebel correction method
yHMMsmooth	- Smoothing for a hidden Markov model (HMM)
HMMsmoothSAR	- Switching auto-regressive HMM smoothing
HMMviterbi	- Viterbi most likely joint hidden state of HMM
kernel	- A kernel evaluated at two points
Kmeans	- K-means clustering algorithm
LDSbackward	- Full backward pass for a latent linear dynamical system (RTS correction method)
LDSbackwardUpdate	- Single backward update for a latent linear dynamical system (RTS smoothing update)
LDSforward	- Full forward pass for a latent linear dynamical system (Kalman filter)
LDSforwardUpdate	- Single forward update for a latent linear dynamical system (Kalman filter)
LDSsmooth	- Linear dynamical system: filtering and smoothing
LDSsubspace	- Subspace method for identifying linear dynamical system
LogReg	- Learning logistic linear regression using gradient ascent
MIXprodBern	- EM training of a mixture of a product of Bernoulli distributions
mixMarkov	- EM training for a mixture of Markov models
NaiveBayesDirichletTest	- Naive Bayes prediction having used a Dirichlet prior for training
NaiveBayesDirichletTrain	- Naive Bayes training using a Dirichlet prior
NaiveBayesTest	- Test Naive Bayes Bernoulli distribution after max likelihood training
NaiveBayesTrain	- Train Naive Bayes Bernoulli distribution using max likelihood
nearNeigh	- Nearest neighbour classification
pca	- Principal components analysis
plsa	- Probabilistic latent semantic analysis
plsaCond	- Conditional PLSA (probabilistic latent semantic analysis)
rbf	- Radial basis function output
SARlearn	- EM training of a switching AR model
SLDSbackward	- Backward pass using a mixture of Gaussians
SLDSforward	- Switching latent linear dynamical system Gaussian sum forward pass
SLDSmargGauss	- Compute the single Gaussian from a weighted SLDS mixture
softloss	- Soft loss function
svdm	- Singular value decomposition with missing values
SVMtrain	- Train a support vector machine

General

argmax	- Performs argmax returning the index and value
assign	- Assigns values to variables
betaXbiggerY	- $p(x>y)$ for $x\sim\text{Beta}(a,b)$, $y\sim\text{Beta}(c,d)$
bar3zcolor	- Plot a 3D bar plot of the matrix Z
avsigmaGauss	- Average of a logistic sigmoid under a Gaussian
cap	- Cap x at absolute value c
chi2test	- Inverse of the chi square cumulative density
count	- For a data matrix (each column is a datapoint), return the state counts
condexp	- Compute normalised p proportional to $\exp(\log p)$
condp	- Make a conditional distribution from the matrix
dirrnd	- Samples from a Dirichlet distribution
field2cell	- Place the field of a structure in a cell
GaussCond	- Return the mean and covariance of a conditioned Gaussian

hinton	- Plot a Hinton diagram
ind2subv	- Subscript vector from linear index
ismember_sorted	- True for member of sorted set
lengthcell	- Length of each cell entry
logdet	- Log determinant of a positive definite matrix computed in a numerically stable manner
logeps	- $\log(x+\text{eps})$
logGaussGamma	- Unnormalised log of the Gauss-Gamma distribution
logsumexp	- Compute $\log(\sum(\exp(a).*b))$ valid for large a
logZdirichlet	- Log normalisation constant of a Dirichlet distribution with parameter u
majority	- Return majority values in each column on a matrix
maxarray	- Maximise a multi-dimensional array over a set of dimensions
maxNarray	- Find the highest values and states of an array over a set of dimensions
mix2mix	- Fit a mixture of Gaussians with another mixture of Gaussians
mvrandn	- Samples from a multivariate Normal (Gaussian) distribution
mygamrnd	- Gamma random variate generator
mynanmean	- Mean of values that are not nan
mynansum	- Sum of values that are not nan
mynchoosek	- Binomial coefficient v choose k
myones	- Same as ones(x), but if x is a scalar, interprets as ones([x 1])
myrand	- Same as rand(x) but if x is a scalar interprets as rand([x 1])
myzeros	- Same as zeros(x) but if x is a scalar interprets as zeros([x 1])
normp	- Make a normalised distribution from an array
randgen	- Generates discrete random variables given the pdf
replace	- Replace instances of a value with another value
sigma	- $1./(1+\exp(-x))$
sigmoid	- $1./(1+\exp(-\text{beta}*x))$
sqdist	- Square distance between vectors in x and y
subv2ind	- Linear index from subscript vector.
sumlog	- $\text{sum}(\log(x))$ with a cutoff at $10e-200$

Miscellaneous

compat	- Compatibility of object F being in position h for image v on grid Gx,Gy
logp	- The logarithm of a specific non-Gaussian distribution
placeobject	- Place the object F at position h in grid Gx,Gy
plotCov	- Return points for plotting an ellipse of a covariance
pointsCov	- Unit variance contours of a 2D Gaussian with mean m and covariance S
setup	- Run me at initialisation – checks for bugs in matlab and initialises path
validgridposition	- Returns 1 if point is on a defined grid