## 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

# Bayesian Reasoning and Machine Learning 

David Barber<br>University College London

CAMBRIDGE UNIVERSITY PRESS<br>University Printing House, Cambridge CB2 8BS, United Kingdom<br>Cambridge University Press is part of the University of Cambridge.<br>It furthers the University's mission by disseminating knowledge in the pursuit of education, learning and research at the highest international levels of excellence.<br>www.cambridge.org<br>Information on this title: www.cambridge.org/9780521518147<br>© D. Barber 2012<br>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.<br>First published 2012<br>6th printing 2015<br>Printed in the United Kingdom by TJ International Ltd, Padstow, Cornwall<br>A catalogue record for this publication is available from the British Library<br>Library of Congress Cataloguing in Publication data<br>Barber, David, 1968-<br>Bayesian reasoning and machine learning / David Barber.<br>p. cm .<br>Includes bibliographical references and index.<br>ISBN 978-0-521-51814-7<br>1. Machine learning. 2. Bayesian statistical decision theory. I. Title.<br>QA267.B347 2012<br>006.3'1 - dc23 2011035553<br>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

[^0]
## CONTENTS

Preface ..... xv
List of notation ..... xx
BRMLtoolbox ..... xxi
I Inference in probabilistic models
1 Probabilistic reasoning
1.1 Probability refresher3
1.1.1 Interpreting conditionalprobability
1.1.2 Probability tables
1.2 Probabilistic reasoning
1.3 Prior, likelihood and posterior
1.3.1 Two dice: what were theindividual scores?
1.4 Summary
1.5 Code
1.6 Exercises
2 Basic graph concepts
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 networks29
3.1 The benefits of structure
3.1.1 Modelling independencies
3.1.2 Reducing the burden ofspecification
3.2 Uncertain and unreliable evidence3.2.1 Uncertain evidence
3.2.2 Unreliable evidence3.3 Belief networks
22
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
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

Cambridge University Press
978-0-521-51814-7-Bayesian Reasoning and Machine Learning
David Barber
Frontmatter
More information

## 5 Efficient inference in trees

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
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
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

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 $\operatorname{KL}(q \mid p)$
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
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
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
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

Cambridge University Press
978-0-521-51814-7-Bayesian Reasoning and Machine Learning
David Barber
Frontmatter
More information

Contents
11.1.3 Maximum likelihood
11.1.4 Identifiability issues
11.2 Expectation maximisation
11.2.1 Variational EM
11.2.2 Classical EM
11.2.3 Application to belief networks
11.2.4 General case
11.2.5 Convergence
11.2.6 Application to Markov networks
11.3 Extensions of EM
11.3.1 Partial M-step
11.3.2 Partial E-step
11.4 A failure case for EM
11.5 Variational Bayes
11.5.1 EM is a special case of variational Bayes
11.5.2 An example: VB for the Asbestos-Smoking-Cancer network
11.6 Optimising the likelihood by gradient methods 11.6.1 Undirected models
11.7 Summary
11.8 Code
11.9 Exercises

12 Bayesian model selection
12.1 Comparing models the Bayesian way
12.2 Illustrations: coin tossing
12.2.1 A discrete parameter space
12.2.2 A continuous parameter space
12.3 Occam's razor and Bayesian complexity penalisation
12.4 A continuous example: curve fitting
12.5 Approximating the model likelihood
12.5.1 Laplace's method
12.5.2 Bayes information criterion
12.6 Bayesian hypothesis testing for outcome analysis
12.6.1 Outcome analysis
12.6.2 $H_{\text {indep }}$ : model likelihood
12.6.3 $H_{\text {same }}$ : model likelihood
12.6.4 Dependent outcome analysis
12.6.5 Is classifier $A$ better than $B$ ?
12.7 Summary
12.8 Code
12.9 Exercises

## III Machine learning

13 Machine learning concepts 305
13.1 Styles of learning
13.1.1 Supervised learning
13.1.2 Unsupervised learning
13.1.3 Anomaly detection
13.1.4 Online (sequential) learning
13.1.5 Interacting with the environment
13.1.6 Semi-supervised learning
13.2 Supervised learning
13.2.1 Utility and loss
13.2.2 Using the empirical distribution
13.2.3 Bayesian decision approach
13.3 Bayes versus empirical decisions
13.4 Summary
13.5 Exercises

14 Nearest neighbour classification
14.1 Do as your neighbour does
14.2 $K$-nearest neighbours
14.3 A probabilistic interpretation of nearest neighbours
14.3.1 When your nearest neighbour is far away
14.4 Summary
14.5 Code
14.6 Exercises

15 Unsupervised linear dimension reduction
15.1 High-dimensional spaces -
low-dimensional manifolds
15.2 Principal components analysis
15.2.1 Deriving the optimal linear reconstruction
15.2.2 Maximum variance criterion
15.2.3 PCA algorithm
15.2.4 PCA and nearest neighbours classification
15.2.5 Comments on PCA

Cambridge University Press
978-0-521-51814-7-Bayesian Reasoning and Machine Learning
David Barber
Frontmatter
More information
15.3 High-dimensional data
15.3.1 Eigen-decomposition for $N<D$
15.3.2 PCA via singular value decomposition
15.4 Latent semantic analysis
15.4.1 Information retrieval
15.5 PCA with missing data
15.5.1 Finding the principal directions
15.5.2 Collaborative filtering using PCA with missing data
15.6 Matrix decomposition methods
15.6.1 Probabilistic latent semantic analysis
15.6.2 Extensions and variations
15.6.3 Applications of PLSA/NMF
15.7 Kernel PCA
15.8 Canonical correlation analysis 15.8.1 SVD formulation
15.9 Summary
15.10 Code
15.11 Exercises

## 16 Supervised linear dimension reduction

16.1 Supervised linear projections
16.2 Fisher's linear discriminant
16.3 Canonical variates
16.3.1 Dealing with the nullspace
16.4 Summary
16.5 Code
16.6 Exercises

17 Linear models
17.1 Introduction: fitting a straight line
17.2 Linear parameter models for regression
17.2.1 Vector outputs
17.2.2 Regularisation
17.2.3 Radial basis functions
17.3 The dual representation and kernels
17.3.1 Regression in the dual space
17.4 Linear parameter models for classification
17.4.1 Logistic regression
17.4.2 Beyond first-order gradient ascent
17.4.3 Avoiding overconfident classification
17.4.4 Multiple classes
17.4.5 The kernel trick for classification
17.5 Support vector machines
17.5.1 Maximum margin linear classifier
17.5.2 Using kernels
17.5.3 Performing the optimisation
17.5.4 Probabilistic interpretation
17.6 Soft zero-one loss for outlier robustness
17.7 Summary
17.8 Code
17.9 Exercises

18 Bayesian linear models
18.1 Regression with additive

Gaussian noise
18.1.1 Bayesian linear parameter models
18.1.2 Determining hyperparameters: ML-II
18.1.3 Learning the hyperparameters using EM
18.1.4 Hyperparameter optimisation: using the gradient
18.1.5 Validation likelihood
18.1.6 Prediction and model averaging
18.1.7 Sparse linear models
18.2 Classification
18.2.1 Hyperparameter optimisation
18.2.2 Laplace approximation
18.2.3 Variational Gaussian approximation
18.2.4 Local variational approximation
18.2.5 Relevance vector machine for classification
18.2.6 Multi-class case
18.3 Summary
18.4 Code
18.5 Exercises

Cambridge University Press
978-0-521-51814-7-Bayesian Reasoning and Machine Learning
David Barber
Frontmatter
More information

## 19 Gaussian processes

19.1 Non-parametric prediction 19.1.1 From parametric to non-parametric
19.1.2 From Bayesian linear models to Gaussian processes
19.1.3 A prior on functions
19.2 Gaussian process prediction
19.2.1 Regression with noisy training outputs
19.3 Covariance functions
19.3.1 Making new covariance functions from old
19.3.2 Stationary covariance functions
19.3.3 Non-stationary covariance functions
19.4 Analysis of covariance
functions
19.4.1 Smoothness of the functions
19.4.2 Mercer kernels
19.4.3 Fourier analysis for stationary kernels
19.5 Gaussian processes for classification
19.5.1 Binary classification
19.5.2 Laplace's approximation
19.5.3 Hyperparameter optimisation
19.5.4 Multiple classes
19.6 Summary
19.7 Code
19.8 Exercises

## 20 Mixture models

20.1 Density estimation using mixtures
20.2 Expectation maximisation for mixture models
20.2.1 Unconstrained discrete tables
20.2.2 Mixture of product of Bernoulli distributions
20.3 The Gaussian mixture model
20.3.1 EM algorithm
20.3.2 Practical issues
20.3.3 Classification using Gaussian mixture models
20.3.4 The Parzen estimator
20.3.5 K-means

Contents
20.3.6 Bayesian mixture models
20.3.7 Semi-supervised learning
20.4 Mixture of experts
20.5 Indicator models
20.5.1 Joint indicator approach: factorised prior
20.5.2 Polya prior
20.6 Mixed membership models
20.6.1 Latent Dirichlet allocation
20.6.2 Graph-based representations of data
20.6.3 Dyadic data
20.6.4 Monadic data
20.6.5 Cliques and adjacency matrices for monadic binary data
20.7 Summary
20.8 Code
20.9 Exercises

21 Latent linear models
462
21.1 Factor analysis
21.1.1 Finding the optimal bias
21.2 Factor analysis: maximum
likelihood
21.2.1 Eigen-approach likelihood optimisation
21.2.2 Expectation maximisation
21.3 Interlude: modelling faces
21.4 Probabilistic principal components analysis
21.5 Canonical correlation analysis and factor analysis
21.6 Independent components analysis
21.7 Summary
21.8 Code
21.9 Exercises

22 Latent ability models
22.1 The Rasch model
22.1.1 Maximum likelihood training
22.1.2 Bayesian Rasch models
22.2 Competition models
22.2.1 Bradley-Terry-Luce model
22.2.2 Elo ranking model
22.2.3 Glicko and TrueSkill

Cambridge University Press
978-0-521-51814-7-Bayesian Reasoning and Machine Learning
David Barber
Frontmatter
More information

## Contents

22.3 Summary
22.4 Code
22.5 Exercises

## IV Dynamical models

23 Discrete-state Markov models
489
23.1 Markov models
$\begin{array}{ll}\text { 23.1.1 } & \text { Equilibrium and } \\ \text { stationary distribution of }\end{array}$ 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 $p\left(h_{t} \mid v_{1: t}\right)$
23.2.3 Parallel smoothing $p\left(h_{t} \mid v_{1: T}\right)$
23.2.4 Correction smoothing
23.2.5 Sampling from $p\left(h_{1: T} \mid v_{1: T}\right)$
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
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

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 $p\left(\mathbf{v}_{1: T}\right)$
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

Cambridge University Press
978-0-521-51814-7-Bayesian Reasoning and Machine Learning
David Barber
Frontmatter
More information
25.5 Reset models
25.5.1 A Poisson reset model
25.5.2 Reset-HMM-LDS
25.6 Summary
25.7 Code
25.8 Exercises

26 Distributed computation
568
26.1 Introduction
26.2 Stochastic Hopfield networks
26.3 Learning sequences
26.3.1 A single sequence
26.3.2 Multiple sequences
26.3.3 Boolean networks
26.3.4 Sequence disambiguation
26.4 Tractable continuous latent variable models
26.4.1 Deterministic latent variables
26.4.2 An augmented Hopfield network
26.5 Neural models
26.5.1 Stochastically spiking neurons
26.5.2 Hopfield membrane potential
26.5.3 Dynamic synapses
26.5.4 Leaky integrate and fire models
26.6 Summary
26.7 Code
26.8 Exercises

## V Approximate inference

## 27 Sampling

27.1 Introduction
27.1.1 Univariate sampling
27.1.2 Rejection sampling
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)
27.4.1 Markov chains
27.4.2 Metropolis-Hastings sampling
27.5 Auxiliary variable methods
27.5.1 Hybrid Monte Carlo (HMC)
27.5.2 Swendson-Wang (SW)
27.5.3 Slice sampling
27.6 Importance sampling
27.6.1 Sequential importance sampling
27.6.2 Particle filtering as an approximate forward pass
27.7 Summary
27.8 Code
27.9 Exercises

## 28 Deterministic approximate inference

 61728.1 Introduction
28.2 The Laplace approximation
28.3 Properties of Kullback-

Leibler variational inference
28.3.1 Bounding the normalisation constant
28.3.2 Bounding the marginal likelihood
28.3.3 Bounding marginal quantities
28.3.4 Gaussian approximations using KL divergence
28.3.5 Marginal and moment matching properties of minimising $\operatorname{KL}(p \mid q)$
28.4 Variational bounding using $\operatorname{KL}(q \mid p)$
28.4.1 Pairwise Markov random field
28.4.2 General mean-field equations
28.4.3 Asynchronous updating guarantees approximation improvement
28.4.4 Structured variational approximation
28.5 Local and KL variational approximations
28.5.1 Local approximation
28.5.2 KL variational approximation
28.6 Mutual information
maximisation: a KL
variational approach

Contents
28.6.1 The information maximisation algorithm
28.6.2 Linear Gaussian decoder
28.7 Loopy belief propagation
28.7.1 Classical BP on an undirected graph
28.7.2 Loopy BP as a variational procedure

### 28.8 Expectation propagation

28.9 MAP for Markov networks
28.9.1 Pairwise Markov networks
28.9.2 Attractive binary Markov networks
28.9.3 Potts model
28.10 Further reading
28.11 Summary
28.12 Code
28.13 Exercises

## Appendix A: Background mathematics

655A. 1 Linear algebra
A. 2 Multivariate calculus
A. 3 Inequalities
A. 4 Optimisation
A. 5 Multivariate optimisation
A. 6 Constrained optimisation using Lagrange multipliers
References 675
Index 689689

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.

Cambridge University Press
978-0-521-51814-7-Bayesian Reasoning and Machine Learning
David Barber
Frontmatter
More information

## Preface

Part I:
Inference in probabilistic models

Part II:
Learning in probabilistic models

Part III:
Machine learning

Part IV:
Dynamical models

Part V:
Approximate inference


| 1: Probabilistic reasoning | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ |
| :--- | :--- | :--- | :--- | :--- |
| 2: Basic graph concepts | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ |
| 3: Belief networks | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ |
| 4: Graphical models | $\bigcirc$ | $\bigcirc$ |  |  |
| 5: Efficient inference in trees | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ |
| 6: The junction tree algorithm | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ |
| 7: Making decisions |  | $\bigcirc$ | $\bigcirc$ |  |

8: Statistics for machine learning
9: Learning as inference
10: Naive Bayes
11: Learning with hidden variables
12: Bayesian model selection


| 23: Discrete-state Markov models | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ |
| :--- | :--- | :--- | :--- | :--- |
| 24: Continuous-state Markov models | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ |
| 25: Switching linear dynamical systems | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ |
| 26: Distributed computation | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ |


| 27: Sampling | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 28: Deterministic approximate inference | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ |

## 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

## Preface

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 Furmston, 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 |
| :---: | :---: | :---: |
| $\operatorname{dom}(x)$ | Domain of a variable | 3 |
| $x=\mathrm{x}$ | The variable $x$ is in the state x | 3 |
| $p(x=\operatorname{tr})$ | Probability of event/variable $x$ being in the state true | 3 |
| $p(x=\mathrm{fa})$ | 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 |
| $p(x \mid y)$ | The probability of $x$ conditioned on $y$ | 4 |
| $\mathcal{X} \Perp \mathcal{Y} \mid \mathcal{Z}$ | Variables $\mathcal{X}$ are independent of variables $\mathcal{Y}$ conditioned on variables $\mathcal{Z}$ | 7 |
| $\mathcal{X} \Pi \mathcal{Y} \mid \mathcal{Z}$ | 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 $\int_{x} f(x) d x$ 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 |
| $\mathrm{pa}(x)$ | The parents of node $x$ | 24 |
| $\operatorname{ch}(x)$ | The children of node $x$ | 24 |
| ne( $x$ ) | Neighbours of node $x$ | 24 |
| $\operatorname{dim}(x)$ | 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 |
| $\operatorname{dim}(\mathbf{x})$ | The dimension of the vector/matrix $\mathbf{x}$ | 183 |
| $\sharp(x=s, y=t)$ | 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 |
| 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 |
| $\operatorname{erf}(x)$ | The (Gaussian) error function | 372 |
| $x_{a: b}$ | $x_{a}, x_{a+1}, \ldots, x_{b}$ | 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

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

## Potential manipulation

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

Cambridge University Press
978-0-521-51814-7-Bayesian Reasoning and Machine Learning
David Barber
Frontmatter
More information

| 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, sumpotGaus- |  |
| sianCanonical.m, sumpotGaussianMoment.m, multpotsGaussianCanonical.m See demosumprodGaussCanon.m, demo- |  |
| SumprodGaussCanonlds.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,ylz) 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 |
| jtree | - Setup a junction tree based on a set of potentials |
| jtreeID | - 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

| ARIds | - 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) |

Cambridge University Press
978-0-521-51814-7-Bayesian Reasoning and Machine Learning
David Barber
Frontmatter
More information

## BRMLToolbox

| 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 | - $\mathrm{p}(\mathrm{x}>\mathrm{y})$ for $\mathrm{x} \sim \operatorname{Beta}(\mathrm{a}, \mathrm{b}), \mathrm{y} \sim \operatorname{Beta}(\mathrm{c}, \mathrm{d})$ |
| 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(logp) |
| 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 |

Cambridge University Press
978-0-521-51814-7-Bayesian Reasoning and Machine Learning
David Barber
Frontmatter
More information

| 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+eps) |
| logGaussGamma | - Unnormalised log of the Gauss-Gamma distribution |
| logsumexp | - Compute $\log (\operatorname{sum}(\exp (\mathrm{a}) . * \mathrm{~b}))$ 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 |
| mix 2 mix | - 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(-beta*x)) |
| sqdist | - Square distance between vectors in x and y |
| subv2ind | - Linear index from subscript vector. |
| sumlog | - $\operatorname{sum}(\log (\mathrm{x})$ ) with a cutoff at $10 \mathrm{e}-200$ |

## Miscellaneous

compat
logp
placeobject
plotCov
pointsCov
setup
validgridposition

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


[^0]:    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.

