## CONTENTS

<table>
<thead>
<tr>
<th>Page</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>xv</td>
<td>Preface</td>
</tr>
<tr>
<td>xx</td>
<td>List of notation</td>
</tr>
<tr>
<td>xxi</td>
<td>BRML Toolbox</td>
</tr>
<tr>
<td>3</td>
<td>Inference in probabilistic models</td>
</tr>
<tr>
<td>3.1</td>
<td>Probabilistic reasoning</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Interpreting conditional probability</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Probability tables</td>
</tr>
<tr>
<td>3.2</td>
<td>Probabilistic reasoning</td>
</tr>
<tr>
<td>3.3</td>
<td>Prior, likelihood and posterior</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Two dice: what were the individual scores?</td>
</tr>
<tr>
<td>3.4</td>
<td>Summary</td>
</tr>
<tr>
<td>3.5</td>
<td>Code</td>
</tr>
<tr>
<td>3.6</td>
<td>Exercises</td>
</tr>
<tr>
<td>22</td>
<td>Basic graph concepts</td>
</tr>
<tr>
<td>4.1</td>
<td>Graphical models</td>
</tr>
<tr>
<td>4.2</td>
<td>Markov networks</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Markov properties</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Markov random fields</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Hammersley–Clifford theorem</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Conditional independence using Markov networks</td>
</tr>
<tr>
<td>4.2.5</td>
<td>Lattice models</td>
</tr>
<tr>
<td>4.3</td>
<td>Chain graphical models</td>
</tr>
<tr>
<td>4.4</td>
<td>Factor graphs</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Conditional independence in factor graphs</td>
</tr>
<tr>
<td>4.5</td>
<td>Expressiveness of graphical models</td>
</tr>
<tr>
<td>4.6</td>
<td>Summary</td>
</tr>
<tr>
<td>4.7</td>
<td>Code</td>
</tr>
<tr>
<td>4.8</td>
<td>Exercises</td>
</tr>
</tbody>
</table>
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 cliques
  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
# Contents

## 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
- 9.5 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
- 11.2 Bayesian methods
- 11.3 Maximum likelihood training of belief networks
- 11.4 Bayesian belief network training
  - 11.4.1 Global and local parameter independence
## Table of 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 $H_{ind}$: model likelihood
    12.6.3 $H_{para}$: 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
  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
  13.2 Supervised learning
    13.2.1 Utility and loss
    13.2.2 Using the empirical distribution
  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
  15.3 PCA algorithm
  15.4 PCA and nearest neighbours classification
  15.5 Comments on PCA
# Contents

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  

© in this web service Cambridge University Press  
www.cambridge.org
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

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

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
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 $p(v_{1:T})$
  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
- 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

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>26.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>26.2</td>
<td>Stochastic Hopfield networks</td>
</tr>
<tr>
<td>26.3</td>
<td>Learning sequences</td>
</tr>
<tr>
<td>26.3.1</td>
<td>A single sequence</td>
</tr>
<tr>
<td>26.3.2</td>
<td>Multiple sequences</td>
</tr>
<tr>
<td>26.3.3</td>
<td>Boolean networks</td>
</tr>
<tr>
<td>26.3.4</td>
<td>Sequence disambiguation</td>
</tr>
<tr>
<td>26.4</td>
<td>Tractable continuous latent variable models</td>
</tr>
<tr>
<td>26.4.1</td>
<td>Deterministic latent variables</td>
</tr>
<tr>
<td>26.4.2</td>
<td>An augmented Hopfield network</td>
</tr>
<tr>
<td>26.5</td>
<td>Neural models</td>
</tr>
<tr>
<td>26.5.1</td>
<td>Stochastically spiking neurons</td>
</tr>
<tr>
<td>26.5.2</td>
<td>Hopfield membrane potential</td>
</tr>
<tr>
<td>26.5.3</td>
<td>Dynamic synapses</td>
</tr>
<tr>
<td>26.5.4</td>
<td>Leaky integrate and fire models</td>
</tr>
<tr>
<td>26.6</td>
<td>Summary</td>
</tr>
<tr>
<td>26.7</td>
<td>Code</td>
</tr>
<tr>
<td>26.8</td>
<td>Exercises</td>
</tr>
</tbody>
</table>

### 27 Sampling

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>27.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>27.1.1</td>
<td>Univariate sampling</td>
</tr>
<tr>
<td>27.1.2</td>
<td>Rejection sampling</td>
</tr>
<tr>
<td>27.1.3</td>
<td>Multivariate sampling</td>
</tr>
<tr>
<td>27.2</td>
<td>Ancestral sampling</td>
</tr>
<tr>
<td>27.2.1</td>
<td>Dealing with evidence</td>
</tr>
<tr>
<td>27.2.2</td>
<td>Perfect sampling for a Markov network</td>
</tr>
<tr>
<td>27.3</td>
<td>Gibbs sampling</td>
</tr>
<tr>
<td>27.3.1</td>
<td>Gibbs sampling as a Markov chain</td>
</tr>
<tr>
<td>27.3.2</td>
<td>Structured Gibbs sampling</td>
</tr>
<tr>
<td>27.3.3</td>
<td>Remarks</td>
</tr>
<tr>
<td>27.4</td>
<td>Markov chain Monte Carlo (MCMC)</td>
</tr>
</tbody>
</table>

### 28 Deterministic approximate inference

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>28.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>28.2</td>
<td>The Laplace approximation</td>
</tr>
<tr>
<td>28.3</td>
<td>Properties of Kullback–Leibler variational inference</td>
</tr>
<tr>
<td>28.3.1</td>
<td>Bounding the normalisation constant</td>
</tr>
<tr>
<td>28.3.2</td>
<td>Bounding the marginal likelihood</td>
</tr>
<tr>
<td>28.3.3</td>
<td>Bounding marginal quantities</td>
</tr>
<tr>
<td>28.3.4</td>
<td>Gaussian approximations using KL divergence</td>
</tr>
<tr>
<td>28.3.5</td>
<td>Marginal and moment matching properties of minimising KL(q)p</td>
</tr>
<tr>
<td>28.4</td>
<td>Variational bounding using KL(q)p</td>
</tr>
<tr>
<td>28.4.1</td>
<td>Pairwise Markov random field</td>
</tr>
<tr>
<td>28.4.2</td>
<td>General mean-field equations</td>
</tr>
<tr>
<td>28.4.3</td>
<td>Asynchronous updating guarantees approximation improvement</td>
</tr>
<tr>
<td>28.4.4</td>
<td>Structured variational approximation</td>
</tr>
<tr>
<td>28.5</td>
<td>Local and KL variational approximations</td>
</tr>
<tr>
<td>28.5.1</td>
<td>Local approximation</td>
</tr>
<tr>
<td>28.5.2</td>
<td>KL variational approximation</td>
</tr>
<tr>
<td>28.6</td>
<td>Mutual information maximisation: a KL variational approach</td>
</tr>
</tbody>
</table>