

PROBABILISTIC GRAPHICAL MODELS

PRINCIPLES AND TECHNIQUES



DAPHNE KOLLER AND NIR FRIEDMAN

Probabilistic Graphical Models

Adaptive Computation and Machine Learning

Thomas Dietterich, Editor

Christopher Bishop, David Heckerman, Michael Jordan, and Michael Kearns, Associate Editors

Bioinformatics: The Machine Learning Approach, Pierre Baldi and Søren Brunak

Reinforcement Learning: An Introduction, Richard S. Sutton and Andrew G. Barto

Graphical Models for Machine Learning and Digital Communication, Brendan J. Frey

Learning in Graphical Models, Michael I. Jordan

Causation, Prediction, and Search, 2nd ed., Peter Spirtes, Clark Glymour, and Richard Scheines

Principles of Data Mining, David Hand, Heikki Mannila, and Padhraic Smyth

Bioinformatics: The Machine Learning Approach, 2nd ed., Pierre Baldi and Søren Brunak

Learning Kernel Classifiers: Theory and Algorithms, Ralf Herbrich

Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, Bernhard Schölkopf and Alexander J. Smola

Introduction to Machine Learning, Ethem Alpaydin

Gaussian Processes for Machine Learning, Carl Edward Rasmussen and Christopher K. I. Williams

Semi-Supervised Learning, Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, eds.

The Minimum Description Length Principle, Peter D. Grünwald

Introduction to Statistical Relational Learning, Lise Getoor and Ben Taskar, eds.

Probabilistic Graphical Models: Principles and Techniques, Daphne Koller and Nir Friedman

Probabilistic Graphical Models

Principles and Techniques

Daphne Koller

Nir Friedman

The MIT Press
Cambridge, Massachusetts
London, England

©2009 Massachusetts Institute of Technology

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

For information about special quantity discounts, please email special_sales@mitpress.mit.edu

This book was set by the authors in $\text{\LaTeX}2_{\epsilon}$.
Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Koller, Daphne.

Probabilistic Graphical Models: Principles and Techniques / Daphne Koller and Nir Friedman.

p. cm. – (Adaptive computation and machine learning)

Includes bibliographical references and index.

ISBN 978-0-262-01319-2 (hardcover : alk. paper)

1. Graphical modeling (Statistics) 2. Bayesian statistical decision theory—Graphic methods. I.

Koller, Daphne. II. Friedman, Nir.

QA279.5.K65 2010

519.5'420285—dc22

2009008615

To our families

*my parents Dov and Ditz
my husband Dan
my daughters Natalie and Maya
D.K.*

*my parents Noga and Gad
my wife Yael
my children Roy and Lior
N.F.*

As far as the laws of mathematics refer to reality, they are not certain, as far as they are certain, they do not refer to reality.

Albert Einstein, 1921

When we try to pick out anything by itself, we find that it is bound fast by a thousand invisible cords that cannot be broken, to everything in the universe.

John Muir, 1869

The actual science of logic is conversant at present only with things either certain, impossible, or entirely doubtful . . . Therefore the true logic for this world is the calculus of probabilities, which takes account of the magnitude of the probability which is, or ought to be, in a reasonable man's mind.

James Clerk Maxwell, 1850

The theory of probabilities is at bottom nothing but common sense reduced to calculus; it enables us to appreciate with exactness that which accurate minds feel with a sort of instinct for which ofttimes they are unable to account.

Pierre Simon Laplace, 1819

Misunderstanding of probability may be the greatest of all impediments to scientific literacy.

Stephen Jay Gould

Contents

Acknowledgments **xxiii**

List of Figures **xxv**

List of Algorithms **xxxi**

List of Boxes **xxxiii**

1 Introduction **1**

- 1.1 Motivation 1
- 1.2 Structured Probabilistic Models 2
 - 1.2.1 Probabilistic Graphical Models 3
 - 1.2.2 Representation, Inference, Learning 5
- 1.3 Overview and Roadmap 6
 - 1.3.1 Overview of Chapters 6
 - 1.3.2 Reader's Guide 9
 - 1.3.3 Connection to Other Disciplines 11
- 1.4 Historical Notes 12

2 Foundations **15**

- 2.1 Probability Theory 15
 - 2.1.1 Probability Distributions 15
 - 2.1.2 Basic Concepts in Probability 18
 - 2.1.3 Random Variables and Joint Distributions 19
 - 2.1.4 Independence and Conditional Independence 23
 - 2.1.5 Querying a Distribution 25
 - 2.1.6 Continuous Spaces 27
 - 2.1.7 Expectation and Variance 31
- 2.2 Graphs 34
 - 2.2.1 Nodes and Edges 34
 - 2.2.2 Subgraphs 35
 - 2.2.3 Paths and Trails 36

2.2.4	Cycles and Loops	36
2.3	Relevant Literature	39
2.4	Exercises	39

I Representation 43

3 The Bayesian Network Representation 45

3.1	Exploiting Independence Properties	45
3.1.1	Independent Random Variables	45
3.1.2	The Conditional Parameterization	46
3.1.3	The Naive Bayes Model	48
3.2	Bayesian Networks	51
3.2.1	The Student Example Revisited	52
3.2.2	Basic Independencies in Bayesian Networks	56
3.2.3	Graphs and Distributions	60
3.3	Independencies in Graphs	68
3.3.1	D-separation	69
3.3.2	Soundness and Completeness	72
3.3.3	An Algorithm for d-Separation	74
3.3.4	I-Equivalence	76
3.4	From Distributions to Graphs	78
3.4.1	Minimal I-Maps	79
3.4.2	Perfect Maps	81
3.4.3	Finding Perfect Maps ★	83
3.5	Summary	92
3.6	Relevant Literature	93
3.7	Exercises	96

4 Undirected Graphical Models 103

4.1	The Misconception Example	103
4.2	Parameterization	106
4.2.1	Factors	106
4.2.2	Gibbs Distributions and Markov Networks	108
4.2.3	Reduced Markov Networks	110
4.3	Markov Network Independencies	114
4.3.1	Basic Independencies	114
4.3.2	Independencies Revisited	117
4.3.3	From Distributions to Graphs	120
4.4	Parameterization Revisited	122
4.4.1	Finer-Grained Parameterization	123
4.4.2	Overparameterization	128
4.5	Bayesian Networks and Markov Networks	134
4.5.1	From Bayesian Networks to Markov Networks	134
4.5.2	From Markov Networks to Bayesian Networks	137

4.5.3	Chordal Graphs	139	
4.6	Partially Directed Models	142	
4.6.1	Conditional Random Fields	142	
4.6.2	Chain Graph Models ★	148	
4.7	Summary and Discussion	151	
4.8	Relevant Literature	152	
4.9	Exercises	153	
5	<i>Local Probabilistic Models</i>	157	
5.1	Tabular CPDs	157	
5.2	Deterministic CPDs	158	
5.2.1	Representation	158	
5.2.2	Independencies	159	
5.3	Context-Specific CPDs	162	
5.3.1	Representation	162	
5.3.2	Independencies	171	
5.4	Independence of Causal Influence	175	
5.4.1	The Noisy-Or Model	175	
5.4.2	Generalized Linear Models	178	
5.4.3	The General Formulation	182	
5.4.4	Independencies	184	
5.5	Continuous Variables	185	
5.5.1	Hybrid Models	189	
5.6	Conditional Bayesian Networks	191	
5.7	Summary	193	
5.8	Relevant Literature	194	
5.9	Exercises	195	
6	<i>Template-Based Representations</i>	199	
6.1	Introduction	199	
6.2	Temporal Models	200	
6.2.1	Basic Assumptions	201	
6.2.2	Dynamic Bayesian Networks	202	
6.2.3	State-Observation Models	207	
6.3	Template Variables and Template Factors	212	
6.4	Directed Probabilistic Models for Object-Relational Domains	216	
6.4.1	Plate Models	216	
6.4.2	Probabilistic Relational Models	222	
6.5	Undirected Representation	228	
6.6	Structural Uncertainty ★	232	
6.6.1	Relational Uncertainty	233	
6.6.2	Object Uncertainty	235	
6.7	Summary	240	
6.8	Relevant Literature	242	
6.9	Exercises	243	

7	<i>Gaussian Network Models</i>	247
7.1	Multivariate Gaussians	247
7.1.1	Basic Parameterization	247
7.1.2	Operations on Gaussians	249
7.1.3	Independencies in Gaussians	250
7.2	Gaussian Bayesian Networks	251
7.3	Gaussian Markov Random Fields	254
7.4	Summary	257
7.5	Relevant Literature	258
7.6	Exercises	258
8	<i>The Exponential Family</i>	261
8.1	Introduction	261
8.2	Exponential Families	261
8.2.1	Linear Exponential Families	263
8.3	Factored Exponential Families	266
8.3.1	Product Distributions	266
8.3.2	Bayesian Networks	267
8.4	Entropy and Relative Entropy	269
8.4.1	Entropy	269
8.4.2	Relative Entropy	272
8.5	Projections	273
8.5.1	Comparison	274
8.5.2	M-Projections	277
8.5.3	I-Projections	282
8.6	Summary	282
8.7	Relevant Literature	283
8.8	Exercises	283
II	Inference	285
9	<i>Exact Inference: Variable Elimination</i>	287
9.1	Analysis of Complexity	288
9.1.1	Analysis of Exact Inference	288
9.1.2	Analysis of Approximate Inference	290
9.2	Variable Elimination: The Basic Ideas	292
9.3	Variable Elimination	296
9.3.1	Basic Elimination	297
9.3.2	Dealing with Evidence	303
9.4	Complexity and Graph Structure: Variable Elimination	305
9.4.1	Simple Analysis	306
9.4.2	Graph-Theoretic Analysis	306
9.4.3	Finding Elimination Orderings ★	310
9.5	Conditioning ★	315

9.5.1	The Conditioning Algorithm	315
9.5.2	Conditioning and Variable Elimination	318
9.5.3	Graph-Theoretic Analysis	322
9.5.4	Improved Conditioning	323
9.6	Inference with Structured CPDs ★	325
9.6.1	Independence of Causal Influence	325
9.6.2	Context-Specific Independence	329
9.6.3	Discussion	335
9.7	Summary and Discussion	336
9.8	Relevant Literature	337
9.9	Exercises	338
10	<i>Exact Inference: Clique Trees</i>	345
10.1	Variable Elimination and Clique Trees	345
10.1.1	Cluster Graphs	346
10.1.2	Clique Trees	346
10.2	Message Passing: Sum Product	348
10.2.1	Variable Elimination in a Clique Tree	349
10.2.2	Clique Tree Calibration	355
10.2.3	A Calibrated Clique Tree as a Distribution	361
10.3	Message Passing: Belief Update	364
10.3.1	Message Passing with Division	364
10.3.2	Equivalence of Sum-Product and Belief Update Messages	368
10.3.3	Answering Queries	369
10.4	Constructing a Clique Tree	372
10.4.1	Clique Trees from Variable Elimination	372
10.4.2	Clique Trees from Chordal Graphs	374
10.5	Summary	376
10.6	Relevant Literature	377
10.7	Exercises	378
11	<i>Inference as Optimization</i>	381
11.1	Introduction	381
11.1.1	Exact Inference Revisited ★	382
11.1.2	The Energy Functional	384
11.1.3	Optimizing the Energy Functional	386
11.2	Exact Inference as Optimization	386
11.2.1	Fixed-Point Characterization	388
11.2.2	Inference as Optimization	390
11.3	Propagation-Based Approximation	391
11.3.1	A Simple Example	391
11.3.2	Cluster-Graph Belief Propagation	396
11.3.3	Properties of Cluster-Graph Belief Propagation	399
11.3.4	Analyzing Convergence ★	401
11.3.5	Constructing Cluster Graphs	404

11.3.6	Variational Analysis	411	
11.3.7	Other Entropy Approximations ★	414	
11.3.8	Discussion	428	
11.4	Propagation with Approximate Messages ★	430	
11.4.1	Factorized Messages	431	
11.4.2	Approximate Message Computation	433	
11.4.3	Inference with Approximate Messages	436	
11.4.4	Expectation Propagation	442	
11.4.5	Variational Analysis	445	
11.4.6	Discussion	448	
11.5	Structured Variational Approximations	448	
11.5.1	The Mean Field Approximation	449	
11.5.2	Structured Approximations	456	
11.5.3	Local Variational Methods ★	469	
11.6	Summary and Discussion	473	
11.7	Relevant Literature	475	
11.8	Exercises	477	
12	<i>Particle-Based Approximate Inference</i>	487	
12.1	Forward Sampling	488	
12.1.1	Sampling from a Bayesian Network	488	
12.1.2	Analysis of Error	490	
12.1.3	Conditional Probability Queries	491	
12.2	Likelihood Weighting and Importance Sampling	492	
12.2.1	Likelihood Weighting: Intuition	492	
12.2.2	Importance Sampling	494	
12.2.3	Importance Sampling for Bayesian Networks	498	
12.2.4	Importance Sampling Revisited	504	
12.3	Markov Chain Monte Carlo Methods	505	
12.3.1	Gibbs Sampling Algorithm	505	
12.3.2	Markov Chains	507	
12.3.3	Gibbs Sampling Revisited	512	
12.3.4	A Broader Class of Markov Chains ★	515	
12.3.5	Using a Markov Chain	518	
12.4	Collapsed Particles	526	
12.4.1	Collapsed Likelihood Weighting ★	527	
12.4.2	Collapsed MCMC	531	
12.5	Deterministic Search Methods ★	536	
12.6	Summary	540	
12.7	Relevant Literature	541	
12.8	Exercises	544	
13	<i>MAP Inference</i>	551	
13.1	Overview	551	
13.1.1	Computational Complexity	551	

13.1.2	Overview of Solution Methods	552
13.2	Variable Elimination for (Marginal) MAP	554
13.2.1	Max-Product Variable Elimination	554
13.2.2	Finding the Most Probable Assignment	556
13.2.3	Variable Elimination for Marginal MAP ★	559
13.3	Max-Product in Clique Trees	562
13.3.1	Computing Max-Marginals	562
13.3.2	Message Passing as Reparameterization	564
13.3.3	Decoding Max-Marginals	565
13.4	Max-Product Belief Propagation in Loopy Cluster Graphs	567
13.4.1	Standard Max-Product Message Passing	567
13.4.2	Max-Product BP with Counting Numbers ★	572
13.4.3	Discussion	575
13.5	MAP as a Linear Optimization Problem ★	577
13.5.1	The Integer Program Formulation	577
13.5.2	Linear Programming Relaxation	579
13.5.3	Low-Temperature Limits	581
13.6	Using Graph Cuts for MAP	588
13.6.1	Inference Using Graph Cuts	588
13.6.2	Nonbinary Variables	592
13.7	Local Search Algorithms ★	595
13.8	Summary	597
13.9	Relevant Literature	598
13.10	Exercises	601
14	<i>Inference in Hybrid Networks</i>	605
14.1	Introduction	605
14.1.1	Challenges	605
14.1.2	Discretization	606
14.1.3	Overview	607
14.2	Variable Elimination in Gaussian Networks	608
14.2.1	Canonical Forms	609
14.2.2	Sum-Product Algorithms	611
14.2.3	Gaussian Belief Propagation	612
14.3	Hybrid Networks	615
14.3.1	The Difficulties	615
14.3.2	Factor Operations for Hybrid Gaussian Networks	618
14.3.3	EP for CLG Networks	621
14.3.4	An “Exact” CLG Algorithm ★	626
14.4	Nonlinear Dependencies	630
14.4.1	Linearization	631
14.4.2	Expectation Propagation with Gaussian Approximation	637
14.5	Particle-Based Approximation Methods	642
14.5.1	Sampling in Continuous Spaces	642
14.5.2	Forward Sampling in Bayesian Networks	643

14.5.3	MCMC Methods	644
14.5.4	Collapsed Particles	645
14.5.5	Nonparametric Message Passing	646
14.6	Summary and Discussion	646
14.7	Relevant Literature	647
14.8	Exercises	649
15	<i>Inference in Temporal Models</i>	651
15.1	Inference Tasks	652
15.2	Exact Inference	653
15.2.1	Filtering in State-Observation Models	653
15.2.2	Filtering as Clique Tree Propagation	654
15.2.3	Clique Tree Inference in DBNs	655
15.2.4	Entanglement	656
15.3	Approximate Inference	661
15.3.1	Key Ideas	661
15.3.2	Factored Belief State Methods	663
15.3.3	Particle Filtering	665
15.3.4	Deterministic Search Techniques	675
15.4	Hybrid DBNs	675
15.4.1	Continuous Models	676
15.4.2	Hybrid Models	683
15.5	Summary	688
15.6	Relevant Literature	690
15.7	Exercises	692
III	Learning	695
16	<i>Learning Graphical Models: Overview</i>	697
16.1	Motivation	697
16.2	Goals of Learning	698
16.2.1	Density Estimation	698
16.2.2	Specific Prediction Tasks	700
16.2.3	Knowledge Discovery	701
16.3	Learning as Optimization	702
16.3.1	Empirical Risk and Overfitting	703
16.3.2	Discriminative versus Generative Training	709
16.4	Learning Tasks	711
16.4.1	Model Constraints	712
16.4.2	Data Observability	712
16.4.3	Taxonomy of Learning Tasks	714
16.5	Relevant Literature	715
17	<i>Parameter Estimation</i>	717
17.1	Maximum Likelihood Estimation	717

17.1.1	The Thumbtack Example	717
17.1.2	The Maximum Likelihood Principle	720
17.2	MLE for Bayesian Networks	722
17.2.1	A Simple Example	723
17.2.2	Global Likelihood Decomposition	724
17.2.3	Table-CPDs	725
17.2.4	Gaussian Bayesian Networks ★	728
17.2.5	Maximum Likelihood Estimation as M-Projection ★	731
17.3	Bayesian Parameter Estimation	733
17.3.1	The Thumbtack Example Revisited	733
17.3.2	Priors and Posteriors	737
17.4	Bayesian Parameter Estimation in Bayesian Networks	741
17.4.1	Parameter Independence and Global Decomposition	742
17.4.2	Local Decomposition	746
17.4.3	Priors for Bayesian Network Learning	748
17.4.4	MAP Estimation ★	751
17.5	Learning Models with Shared Parameters	754
17.5.1	Global Parameter Sharing	755
17.5.2	Local Parameter Sharing	760
17.5.3	Bayesian Inference with Shared Parameters	762
17.5.4	Hierarchical Priors ★	763
17.6	Generalization Analysis ★	769
17.6.1	Asymptotic Analysis	769
17.6.2	PAC-Bounds	770
17.7	Summary	776
17.8	Relevant Literature	777
17.9	Exercises	778
18	Structure Learning in Bayesian Networks	783
18.1	Introduction	783
18.1.1	Problem Definition	783
18.1.2	Overview of Methods	785
18.2	Constraint-Based Approaches	786
18.2.1	General Framework	786
18.2.2	Independence Tests	787
18.3	Structure Scores	790
18.3.1	Likelihood Scores	791
18.3.2	Bayesian Score	794
18.3.3	Marginal Likelihood for a Single Variable	797
18.3.4	Bayesian Score for Bayesian Networks	799
18.3.5	Understanding the Bayesian Score	801
18.3.6	Priors	804
18.3.7	Score Equivalence ★	807
18.4	Structure Search	807
18.4.1	Learning Tree-Structured Networks	808

18.4.2	Known Order	809	
18.4.3	General Graphs	811	
18.4.4	Learning with Equivalence Classes ★	821	
18.5	Bayesian Model Averaging ★	824	
18.5.1	Basic Theory	824	
18.5.2	Model Averaging Given an Order	826	
18.5.3	The General Case	828	
18.6	Learning Models with Additional Structure	832	
18.6.1	Learning with Local Structure	833	
18.6.2	Learning Template Models	837	
18.7	Summary and Discussion	838	
18.8	Relevant Literature	840	
18.9	Exercises	843	
19	<i>Partially Observed Data</i>	849	
19.1	Foundations	849	
19.1.1	Likelihood of Data and Observation Models	849	
19.1.2	Decoupling of Observation Mechanism	853	
19.1.3	The Likelihood Function	856	
19.1.4	Identifiability	860	
19.2	Parameter Estimation	862	
19.2.1	Gradient Ascent	863	
19.2.2	Expectation Maximization (EM)	868	
19.2.3	Comparison: Gradient Ascent versus EM	887	
19.2.4	Approximate Inference ★	893	
19.3	Bayesian Learning with Incomplete Data ★	897	
19.3.1	Overview	897	
19.3.2	MCMC Sampling	899	
19.3.3	Variational Bayesian Learning	904	
19.4	Structure Learning	908	
19.4.1	Scoring Structures	909	
19.4.2	Structure Search	917	
19.4.3	Structural EM	920	
19.5	Learning Models with Hidden Variables	925	
19.5.1	Information Content of Hidden Variables	926	
19.5.2	Determining the Cardinality	928	
19.5.3	Introducing Hidden Variables	930	
19.6	Summary	933	
19.7	Relevant Literature	934	
19.8	Exercises	935	
20	<i>Learning Undirected Models</i>	943	
20.1	Overview	943	
20.2	The Likelihood Function	944	
20.2.1	An Example	944	

20.2.2	Form of the Likelihood Function	946
20.2.3	Properties of the Likelihood Function	947
20.3	Maximum (Conditional) Likelihood Parameter Estimation	949
20.3.1	Maximum Likelihood Estimation	949
20.3.2	Conditionally Trained Models	950
20.3.3	Learning with Missing Data	954
20.3.4	Maximum Entropy and Maximum Likelihood ★	956
20.4	Parameter Priors and Regularization	958
20.4.1	Local Priors	958
20.4.2	Global Priors	961
20.5	Learning with Approximate Inference	961
20.5.1	Belief Propagation	962
20.5.2	MAP-Based Learning ★	967
20.6	Alternative Objectives	969
20.6.1	Pseudolikelihood and Its Generalizations	970
20.6.2	Contrastive Optimization Criteria	974
20.7	Structure Learning	978
20.7.1	Structure Learning Using Independence Tests	979
20.7.2	Score-Based Learning: Hypothesis Spaces	981
20.7.3	Objective Functions	982
20.7.4	Optimization Task	985
20.7.5	Evaluating Changes to the Model	992
20.8	Summary	996
20.9	Relevant Literature	998
20.10	Exercises	1001

IV Actions and Decisions 1007

21 Causality 1009

21.1	Motivation and Overview	1009
21.1.1	Conditioning and Intervention	1009
21.1.2	Correlation and Causation	1012
21.2	Causal Models	1014
21.3	Structural Causal Identifiability	1017
21.3.1	Query Simplification Rules	1017
21.3.2	Iterated Query Simplification	1020
21.4	Mechanisms and Response Variables ★	1026
21.5	Partial Identifiability in Functional Causal Models ★	1031
21.6	Counterfactual Queries ★	1034
21.6.1	Twinned Networks	1034
21.6.2	Bounds on Counterfactual Queries	1037
21.7	Learning Causal Models	1040
21.7.1	Learning Causal Models without Confounding Factors	1041
21.7.2	Learning from Interventional Data	1044

21.7.3	Dealing with Latent Variables ★	1048
21.7.4	Learning Functional Causal Models ★	1051
21.8	Summary	1053
21.9	Relevant Literature	1054
21.10	Exercises	1055
22	<i>Utilities and Decisions</i>	1059
22.1	Foundations: Maximizing Expected Utility	1059
22.1.1	Decision Making Under Uncertainty	1059
22.1.2	Theoretical Justification ★	1062
22.2	Utility Curves	1064
22.2.1	Utility of Money	1065
22.2.2	Attitudes Toward Risk	1066
22.2.3	Rationality	1067
22.3	Utility Elicitation	1068
22.3.1	Utility Elicitation Procedures	1068
22.3.2	Utility of Human Life	1069
22.4	Utilities of Complex Outcomes	1071
22.4.1	Preference and Utility Independence ★	1071
22.4.2	Additive Independence Properties	1074
22.5	Summary	1081
22.6	Relevant Literature	1082
22.7	Exercises	1084
23	<i>Structured Decision Problems</i>	1085
23.1	Decision Trees	1085
23.1.1	Representation	1085
23.1.2	Backward Induction Algorithm	1087
23.2	Influence Diagrams	1088
23.2.1	Basic Representation	1089
23.2.2	Decision Rules	1090
23.2.3	Time and Recall	1092
23.2.4	Semantics and Optimality Criterion	1093
23.3	Backward Induction in Influence Diagrams	1095
23.3.1	Decision Trees for Influence Diagrams	1096
23.3.2	Sum-Max-Sum Rule	1098
23.4	Computing Expected Utilities	1100
23.4.1	Simple Variable Elimination	1100
23.4.2	Multiple Utility Variables: Simple Approaches	1102
23.4.3	Generalized Variable Elimination ★	1103
23.5	Optimization in Influence Diagrams	1107
23.5.1	Optimizing a Single Decision Rule	1107
23.5.2	Iterated Optimization Algorithm	1108
23.5.3	Strategic Relevance and Global Optimality ★	1110
23.6	Ignoring Irrelevant Information ★	1119

23.7	Value of Information	1121
23.7.1	Single Observations	1122
23.7.2	Multiple Observations	1124
23.8	Summary	1126
23.9	Relevant Literature	1127
23.10	Exercises	1130
24	Epilogue	1133
A	Background Material	1137
A.1	Information Theory	1137
A.1.1	Compression and Entropy	1137
A.1.2	Conditional Entropy and Information	1139
A.1.3	Relative Entropy and Distances Between Distributions	1140
A.2	Convergence Bounds	1143
A.2.1	Central Limit Theorem	1144
A.2.2	Convergence Bounds	1145
A.3	Algorithms and Algorithmic Complexity	1146
A.3.1	Basic Graph Algorithms	1146
A.3.2	Analysis of Algorithmic Complexity	1147
A.3.3	Dynamic Programming	1149
A.3.4	Complexity Theory	1150
A.4	Combinatorial Optimization and Search	1154
A.4.1	Optimization Problems	1154
A.4.2	Local Search	1154
A.4.3	Branch and Bound Search	1160
A.5	Continuous Optimization	1161
A.5.1	Characterizing Optima of a Continuous Function	1161
A.5.2	Gradient Ascent Methods	1163
A.5.3	Constrained Optimization	1167
A.5.4	Convex Duality	1171
	Bibliography	1173
	Notation Index	1211
	Subject Index	1215

Acknowledgments

This book owes a considerable debt of gratitude to the many people who contributed to its creation, and to those who have influenced our work and our thinking over the years.

First and foremost, we want to thank our students, who, by asking the right questions, and forcing us to formulate clear and precise answers, were directly responsible for the inception of this book and for any clarity of presentation.

We have been fortunate to share the same mentors, who have had a significant impact on our development as researchers and as teachers: Joe Halpern, Stuart Russell. Much of our core views on probabilistic models have been influenced by Judea Pearl. Judea through his persuasive writing and vivid presentations inspired us, and many other researchers of our generation, to plunge into research in this field.

There are many people whose conversations with us have helped us in thinking through some of the more difficult concepts in the book: Nando de Freitas, Gal Elidan, Dan Geiger, Amir Globerson, Uri Lerner, Chris Meek, David Sontag, Yair Weiss, and Ramin Zabih. Others, in conversations and collaborations over the year, have also influenced our thinking and the presentation of the material: Pieter Abbeel, Jeff Bilmes, Craig Boutilier, Moises Goldszmidt, Carlos Guestrin, David Heckerman, Eric Horvitz, Tommi Jaakkola, Michael Jordan, Kevin Murphy, Andrew Ng, Ben Taskar, and Sebastian Thrun.

We especially want to acknowledge Gal Elidan for constant encouragement, valuable feedback, and logistic support at many critical junctions, throughout the long years of writing this book.

Over the course of the years of work on this book, many people have contributed to it by providing insights, engaging in enlightening discussions, and giving valuable feedback. It is impossible to individually acknowledge all of the people who made such contributions. However, we specifically wish to express our gratitude to those people who read large parts of the book and gave detailed feedback: Rahul Biswas, James Cussens, James Diebel, Yoni Donner, Tal El-Hay, Gal Elidan, Stanislav Funiak, Amir Globerson, Russ Greiner, Carlos Guestrin, Tim Heilman, Jeremy Heitz, Maureen Hillenmeyer, Ariel Jaimovich, Tommy Kaplan, Jonathan Laserson, Ken Levine, Brian Milch, Kevin Murphy, Ben Packer, Ronald Parr, Dana Pe'er, and Christian Shelton.

We are deeply grateful to the following people, who contributed specific text and/or figures, mostly to the case studies and concept boxes without which this book would be far less interesting: Gal Elidan, to chapter 11, chapter 18, and chapter 19; Stephen Gould, to chapter 4 and chapter 13; Vladimir Jojic, to chapter 12; Jonathan Laserson, to chapter 19; Uri Lerner, to chapter 14; Andrew McCallum and Charles Sutton, to chapter 4; Brian Milch, to chapter 6; Kevin

Murphy, to chapter 15; and Benjamin Packer, to many of the exercises used throughout the book. In addition, we are very grateful to Amir Globerson, David Sontag and Yair Weiss whose insights on chapter 13 played a key role in the development of the material in that chapter.

Special thanks are due to Bob Prior at MIT Press who convinced us to go ahead with this project and was constantly supportive, enthusiastic and patient in the face of the recurring delays and missed deadlines. We thank Greg McNamee, our copy editor, and Mary Reilly, our artist, for their help in improving this book considerably. We thank Chris Manning, for allowing us to use his \LaTeX macros for typesetting this book, and for providing useful advice on how to use them. And we thank Miles Davis for invaluable technical support.

We also wish to thank the many colleagues who used drafts of this book in teaching provided enthusiastic feedback that encouraged us to continue this project at times where it seemed unending. Sebastian Thrun deserves a special note of thanks, for forcing us to set a deadline for completion of this book and to stick to it.

We also want to thank the past and present members of the DAGS group at Stanford, and the Computational Biology group at the Hebrew University, many of whom also contributed ideas, insights, and useful comments. We specifically want to thank them for bearing with us while we devoted far too much of our time to working on this book.

Finally, noone deserves our thanks more than our long-suffering families — Natalie Anna Koller Avida, Maya Rika Koller Avida, and Dan Avida; Lior, Roy, and Yael Friedman — for their continued love, support, and patience, as they watched us work evenings and weekends to complete this book. We could never have done this without you.

List of Figures

1.1	Different perspectives on probabilistic graphical models	4
1.2	A reader's guide to the structure and dependencies in this book	10
2.1	Example of a joint distribution $P(\textit{Intelligence}, \textit{Grade})$	22
2.2	Example PDF of three Gaussian distributions	29
2.3	An example of a partially directed graph \mathcal{K}	35
2.4	Induced graphs and their upward closure	35
2.5	An example of a polytree	38
3.1	Simple Bayesian networks for the student example	48
3.2	The Bayesian network graph for a naive Bayes model	50
3.3	The Bayesian Network graph for the Student example	52
3.4	Student Bayesian network $\mathcal{B}^{\textit{student}}$ with CPDs	53
3.5	The four possible two-edge trails	70
3.6	A simple example for the d-separation algorithm	76
3.7	Skeletons and v-structures in a network	77
3.8	Three minimal I-maps for $P_{\mathcal{B}^{\textit{student}}}$, induced by different orderings	80
3.9	Network for the OneLetter example	82
3.10	Attempted Bayesian network models for the Misconception example	83
3.11	Simple example of compelled edges in an equivalence class.	88
3.12	Rules for orienting edges in PDAG	89
3.13	More complex example of compelled edges in an equivalence class	90
3.14	A Bayesian network with qualitative influences	97
3.15	A simple network for a burglary alarm domain	98
3.16	Illustration of the concept of a self-contained set	101
4.1	Factors for the Misconception example	104
4.2	Joint distribution for the Misconception example	105
4.3	An example of factor product	107
4.4	The cliques in two simple Markov networks	109
4.5	An example of factor reduction	111
4.6	Markov networks for the factors in an extended Student example	112

4.7	An attempt at an I-map for a nonpositive distribution P	122
4.8	Different factor graphs for the same Markov network	123
4.9	Energy functions for the Misconception example	124
4.10	Alternative but equivalent energy functions	128
4.11	Canonical energy function for the Misconception example	130
4.12	Example of alternative definition of d-separation based on Markov networks	137
4.13	Minimal I-map Bayesian networks for a nonchordal Markov network	138
4.14	Different linear-chain graphical models	143
4.15	A chain graph \mathcal{K} and its moralized version	149
4.16	Example for definition of c-separation in a chain graph	150
5.1	Example of a network with a deterministic CPD	160
5.2	A slightly more complex example with deterministic CPDs	161
5.3	The Student example augmented with a <i>Job</i> variable	162
5.4	A tree-CPD for $P(J \mid A, S, L)$	163
5.5	The OneLetter example of a multiplexer dependency	165
5.6	tree-CPD for a rule-based CPD	169
5.7	Example of removal of spurious edges	173
5.8	Two reduced CPDs for the OneLetter example	174
5.9	Decomposition of the noisy-or model for <i>Letter</i>	176
5.10	The behavior of the noisy-or model	177
5.11	The behavior of the sigmoid CPD	180
5.12	Example of the multinomial logistic CPD	181
5.13	Independence of causal influence	182
5.14	Generalized linear model for a thermostat	191
5.15	Example of encapsulated CPDs for a computer system model	193
6.1	A highly simplified DBN for monitoring a vehicle	203
6.2	HMM as a DBN	203
6.3	Two classes of DBNs constructed from HMMs	205
6.4	A simple 4-state HMM	208
6.5	One possible world for the University example	215
6.6	Plate model for a set of coin tosses sampled from a single coin	217
6.7	Plate models and ground Bayesian networks for a simplified Student example	219
6.8	Illustration of probabilistic interactions in the University domain	220
6.9	Examples of dependency graphs	227
7.1	Examples of 2-dimensional Gaussians	249
8.1	Example of M- and I-projections into the family of Gaussian distributions	275
8.2	Example of M- and I-projections for a discrete distribution	276
8.3	Relationship between parameters, distributions, and expected sufficient statistics	279
9.1	Network used to prove \mathcal{NP} -hardness of exact inference	289
9.2	Computing $P(D)$ by summing out the joint distribution	294
9.3	The first transformation on the sum of figure 9.2	295

9.4	The second transformation on the sum of figure 9.2	295
9.5	The third transformation on the sum of figure 9.2	295
9.6	The fourth transformation on the sum of figure 9.2	295
9.7	Example of factor marginalization	297
9.8	The Extended-Student Bayesian network	300
9.9	Understanding intermediate factors in variable elimination	303
9.10	Variable elimination as graph transformation in the Student example	308
9.11	Induced graph and clique tree for the Student example	309
9.12	Networks where conditioning performs unnecessary computation	321
9.13	Induced graph for the Student example using both conditioning and elimination	323
9.14	Different decompositions for a noisy-or CPD	326
9.15	Example Bayesian network with rule-based structure	329
9.16	Conditioning in a network with CSI	334
10.1	Cluster tree for the VE execution in table 9.1	346
10.2	Simplified clique tree \mathcal{T} for the Extended Student network	349
10.3	Message propagations with different root cliques in the Student clique tree	350
10.4	An abstract clique tree that is not chain-structured	352
10.5	Two steps in a downward pass in the Student network	356
10.6	Final beliefs for the Misconception example	362
10.7	An example of factor division	365
10.8	A modified Student BN with an unambitious student	373
10.9	A clique tree for the modified Student BN of figure 10.8	373
10.10	Example of clique tree construction algorithm	375
11.1	An example of a cluster graph versus a clique tree	391
11.2	An example run of loopy belief propagation	392
11.3	Two examples of generalized cluster graph for an MRF	393
11.4	An example of a 4×4 two-dimensional grid network	398
11.5	An example of generalized cluster graph for a 3×3 grid network	399
11.6	A generalized cluster graph for the 3×3 grid when viewed as pairwise MRF	405
11.7	Examples of generalized cluster graphs for network with potentials $\{A, B, C\}$, $\{B, C, D\}$, $\{B, D, F\}$, $\{B, E\}$ and $\{D, E\}$	406
11.8	Examples of generalized cluster graphs for networks with potentials $\{A, B, C\}$, $\{B, C, D\}$, and $\{A, C, D\}$	407
11.9	An example of simple region graph	420
11.10	The region graph corresponding to the Bethe cluster graph of figure 11.7a	421
11.11	The messages participating in different region graph computations	425
11.12	A cluster for a 4×4 grid network	430
11.13	Effect of different message factorizations on the beliefs in the receiving factor	431
11.14	Example of propagation in cluster tree with factorized messages	433
11.15	Markov network used to demonstrate approximate message passing	438
11.16	An example of a multimodal mean field energy functional landscape	456
11.17	Two structures for variational approximation of a 4×4 grid network	457
11.18	A diamond network and three possible approximating structures	462

11.19	Simplification of approximating structure in cluster mean field	468
11.20	Illustration of the variational bound $-\ln(x) \geq -\lambda x + \ln(\lambda) + 1$	469
12.1	The Student network $\mathcal{B}^{student}$ revisited	488
12.2	The mutilated network $\mathcal{B}_{I=i^1, G=g^2}^{student}$ used for likelihood weighting	499
12.3	The Grasshopper Markov chain	507
12.4	A simple Markov chain	509
12.5	A Bayesian network with four students, two courses, and five grades	514
12.6	Visualization of a Markov chain with low conductance	520
12.7	Networks illustrating collapsed importance sampling	528
13.1	Example of the max-marginalization factor operation for variable B	555
13.2	A network where a marginal MAP query requires exponential time	561
13.3	The max-marginals for the Misconception example	564
13.4	Two induced subgraphs derived from figure 11.3a	570
13.5	Example graph construction for applying min-cut to the binary MAP problem	590
14.1	Gaussian MRF illustrating convergence properties of Gaussian belief propagation	615
14.2	CLG network used to demonstrate hardness of inference	615
14.3	Joint marginal distribution $p(X_1, X_2)$ for a network as in figure 14.2	616
14.4	Summing and collapsing a Gaussian mixture	619
14.5	Example of unnormalizable potentials in a CLG clique tree	623
14.6	A simple CLG and possible clique trees with different correctness properties	624
14.7	Different Gaussian approximation methods for a nonlinear dependency	636
15.1	Clique tree for HMM	654
15.2	Different clique trees for the Car DBN of figure 6.1	659
15.3	Nonpersistent 2-TBN and different possible clique trees	660
15.4	Performance of likelihood weighting over time	667
15.5	Illustration of the particle filtering algorithm	669
15.6	Likelihood weighting and particle filtering over time	670
15.7	Three collapsing strategies for CLG DBNs, and their EP perspective	687
16.1	The effect of ignoring hidden variables	714
17.1	A simple thumbtack tossing experiment	718
17.2	The likelihood function for the sequence of tosses H, T, T, H, H	718
17.3	Meta-network for IID samples of a random variable	734
17.4	Examples of Beta distributions for different choices of hyperparameters	736
17.5	The effect of the Beta prior on our posterior estimates	741
17.6	The effect of different priors on smoothing our parameter estimates	742
17.7	Meta-network for IID samples from $X \rightarrow Y$ with global parameter independence	743
17.8	Meta-network for IID samples from $X \rightarrow Y$ with local parameter independence	746
17.9	Two plate models for the University example, with explicit parameter variables	758
17.10	Example meta-network for a model with shared parameters	763
17.11	Independent and hierarchical priors	765

18.1	Marginal training likelihood versus expected likelihood on underlying distribution	796
18.2	Maximal likelihood score versus marginal likelihood for the data $\langle H, T, T, H, H \rangle$.	797
18.3	The effect of correlation on the Bayesian score	801
18.4	The Bayesian scores of three structures for the ICU-Alarm domain	802
18.5	Example of a search problem requiring edge deletion	813
18.6	Example of a search problem requiring edge reversal	814
18.7	Performance of structure and parameter learning for instances from ICU-Alarm network	820
18.8	MCMC structure search using 500 instances from ICU-Alarm network	830
18.9	MCMC structure search using 1,000 instances from ICU-Alarm network	831
18.10	MCMC order search using 1,000 instances from ICU-Alarm network	833
18.11	A simple module network	847
19.1	Observation models in two variants of the thumbtack example	851
19.2	An example satisfying MAR but not MCAR	853
19.3	A visualization of a multimodal likelihood function with incomplete data	857
19.4	The meta-network for parameter estimation for $X \rightarrow Y$	858
19.5	Contour plots for the likelihood function for $X \rightarrow Y$	858
19.6	A simple network used to illustrate learning algorithms for missing data	864
19.7	The naive Bayes clustering model	875
19.8	The hill-climbing process performed by the EM algorithm	882
19.9	Plate model for Bayesian clustering	902
19.10	Nondecomposability of structure scores in the case of missing data	918
19.11	An example of a network with a hierarchy of hidden variables	931
19.12	An example of a network with overlapping hidden variables	931
20.1	Log-likelihood surface for the Markov network $A-B-C$	945
20.2	A highly connected CRF that allows simple inference when conditioned	952
20.3	Laplacian distribution ($\beta = 1$) and Gaussian distribution ($\sigma^2 = 1$)	959
21.1	Mutilated Student networks representing interventions	1015
21.2	Causal network for Simpson's paradox	1016
21.3	Models where $P(Y \mid do(X))$ is identifiable	1025
21.4	Models where $P(Y \mid do(X))$ is not identifiable	1025
21.5	A simple functional causal model for a clinical trial	1030
21.6	Twinned counterfactual network with an intervention	1036
21.7	Models corresponding to the equivalence class of the Student network	1043
21.8	Example PAG and members of its equivalence class	1050
21.9	Learned causal network for exercise 21.12	1057
22.1	Example curve for the utility of money	1066
22.2	Utility curve and its consequences to an agent's attitude toward risk	1067
23.1	Decision trees for the Entrepreneur example	1086
23.2	Influence diagram \mathcal{I}_F for the basic Entrepreneur example	1089
23.3	Influence diagram $\mathcal{I}_{F,C}$ for Entrepreneur example with market survey	1091

23.4	Decision tree for the influence diagram $\mathcal{I}_{F,C}$ in the Entrepreneur example	1096
23.5	Iterated optimization versus variable elimination	1099
23.6	An influence diagram with multiple utility variables	1101
23.7	Influence diagrams, augmented to test for s-reachability	1112
23.8	Influence diagrams and their relevance graphs	1114
23.9	Clique tree for the imperfect-recall influence diagram of figure 23.5.	1116
23.10	More complex influence diagram \mathcal{I}_S for the Student scenario	1120
23.11	Example for computing value of information using an influence diagram	1123
A.1	Illustration of asymptotic complexity	1149
A.2	Illustration of line search with Brent's method	1165
A.3	Two examples of the convergence problem with line search	1166

List of Algorithms

3.1	Algorithm for finding nodes reachable from X given Z via active trails	75
3.2	Procedure to build a minimal I-map given an ordering	80
3.3	Recovering the undirected skeleton for a distribution P that has a P-map	85
3.4	Marking immoralities in the construction of a perfect map	86
3.5	Finding the class PDAG characterizing the P-map of a distribution P	89
5.1	Computing d-separation in the presence of deterministic CPDs	160
5.2	Computing d-separation in the presence of context-specific CPDs	173
9.1	Sum-product variable elimination algorithm	298
9.2	Using Sum-Product-VE for computing conditional probabilities	304
9.3	Maximum cardinality search for constructing an elimination ordering	312
9.4	Greedy search for constructing an elimination ordering	314
9.5	Conditioning algorithm	317
9.6	Rule splitting algorithm	332
9.7	Sum-product variable elimination for sets of rules	333
10.1	Upward pass of variable elimination in clique tree	353
10.2	Calibration using sum-product message passing in a clique tree	357
10.3	Calibration using belief propagation in clique tree	367
10.4	Out-of-clique inference in clique tree	371
11.1	Calibration using sum-product belief propagation in a cluster graph	397
11.2	Convergent message passing for Bethe cluster graph with convex counting numbers	418
11.3	Algorithm to construct a saturated region graph	423
11.4	Projecting a factor set to produce a set of marginals over a given set of scopes	434
11.5	Modified version of BU-Message that incorporates message projection	441
11.6	Message passing step in the expectation propagation algorithm	443
11.7	The Mean-Field approximation algorithm	455
12.1	Forward Sampling in a Bayesian network	489
12.2	Likelihood-weighted particle generation	493
12.3	Likelihood weighting with a data-dependent stopping rule	502
12.4	Generating a Gibbs chain trajectory	506
12.5	Generating a Markov chain trajectory	509
13.1	Variable elimination algorithm for MAP	557

13.2	Max-product message computation for MAP	562
13.3	Calibration using max-product BP in a Bethe-structured cluster graph	573
13.4	Graph-cut algorithm for MAP in pairwise binary MRFs with submodular potentials	591
13.5	Alpha-expansion algorithm	593
13.6	Efficient min-sum message passing for untruncated 1-norm energies	603
14.1	Expectation propagation message passing for CLG networks	622
15.1	Filtering in a DBN using a template clique tree	657
15.2	Likelihood-weighted particle generation for a 2-TBN	666
15.3	Likelihood weighting for filtering in DBNs	666
15.4	Particle filtering for DBNs	670
18.1	Data perturbation search	817
19.1	Computing the gradient in a network with table-CPDs	867
19.2	Expectation-maximization algorithm for BN with table-CPDs	873
19.3	The structural EM algorithm for structure learning	922
19.4	The incremental EM algorithm for network with table-CPDs	939
19.5	Proposal distribution for collapsed Metropolis-Hastings over data completions	941
19.6	Proposal distribution over partitions in the Dirichlet process prior	942
20.1	Greedy score-based structure search algorithm for log-linear models	986
23.1	Finding the MEU strategy in a decision tree	1088
23.2	Generalized variable elimination for joint factors in influence diagrams	1105
23.3	Iterated optimization for influence diagrams with acyclic relevance graphs	1116
A.1	Topological sort of a graph	1146
A.2	Maximum weight spanning tree in an undirected graph	1147
A.3	Recursive algorithm for computing Fibonacci numbers	1150
A.4	Dynamic programming algorithm for computing Fibonacci numbers	1150
A.5	Greedy local search algorithm with search operators	1155
A.6	Local search with tabu list	1157
A.7	Beam search	1158
A.8	Greedy hill-climbing search with random restarts	1159
A.9	Branch and bound algorithm	1161
A.10	Simple gradient ascent algorithm	1164
A.11	Conjugate gradient ascent	1167

List of Boxes

Box 3.A	Concept: The Naive Bayes Model	50
Box 3.B	Case Study: The Genetics Example	58
Figure 3.B.1	Modeling Genetic Inheritance	58
Box 3.C	Skill: Knowledge Engineering	64
Box 3.D	Case Study: Medical Diagnosis Systems	67
Box 4.A	Concept: Pairwise Markov Networks	110
Figure 4.A.1	A pairwise Markov network (MRF) structured as a grid.	110
Box 4.B	Case Study: Markov Networks for Computer Vision	112
Figure 4.B.1	Two examples of image segmentation results	114
Box 4.C	Concept: Ising Models and Boltzmann Machines	126
Box 4.D	Concept: Metric MRFs	127
Box 4.E	Case Study: CRFs for Text Analysis	145
Figure 4.E.1	Two models for text analysis based on a linear chain CRF	147
Box 5.A	Case Study: Context-Specificity in Diagnostic Networks	166
Figure 5.A.1	Context-specific independencies for diagnostic networks	167
Box 5.B	Concept: Multinets and Similarity Networks	170
Box 5.C	Concept: BN2O Networks	177
Figure 5.C.1	A two-layer noisy-or network	178
Box 5.D	Case Study: Noisy Rule Models for Medical Diagnosis	183
Box 5.E	Case Study: Robot Motion and Sensors	187
Figure 5.E.1	Probabilistic model for robot localization track	188
Box 6.A	Case Study: HMMs and Phylo-HMMs for Gene Finding	206
Box 6.B	Case Study: HMMs for Speech Recognition	209
Figure 6.B.1	A phoneme-level HMM for a fairly complex phoneme.	210
Box 6.C	Case Study: Collective Classification of Web Pages	231
Box 6.D	Case Study: Object Uncertainty and Citation Matching	238
Figure 6.D.1	Two template models for citation-matching	239
Box 9.A	Concept: The Network Polynomial	304
Box 9.B	Concept: Polytrees	313
Box 9.C	Case Study: Variable Elimination Orderings	315
Figure 9.C.1	Comparison of algorithms for selecting variable elimination ordering	316

Box 9.D	Case Study: Inference with Local Structure	335
Box 10.A	Skill: Efficient Implementation of Factor Manipulation Algorithms	358
Algorithm 10.A.1	Efficient implementation of a factor product operation.	359
Box 11.A	Case Study: Turbocodes and loopy belief propagation	393
Figure 11.A.1	Two examples of codes	394
Box 11.B	Skill: Making loopy belief propagation work in practice	407
Box 11.C	Case Study: BP in practice	409
Figure 11.C.1	Example of behavior of BP in practice on an 11×11 Ising grid	410
Box 12.A	Skill: Sampling from a Discrete Distribution	489
Box 12.B	Skill: MCMC in Practice	522
Box 12.C	Case Study: The bugs System	525
Figure 12.C.1	Example of bugs model specification	525
Box 12.D	Concept: Correspondence and Data Association	532
Figure 12.D.1	Results of a correspondence algorithm for 3D human body scans	535
Box 13.A	Concept: Tree-Reweighted Belief Propagation	576
Box 13.B	Case Study: Energy Minimization in Computer Vision	593
Figure 13.B.1	MAP inference for stereo reconstruction	594
Box 15.A	Case Study: Tracking, Localization, and Mapping	679
Figure 15.A.1	Illustration of Kalman filtering for tracking	679
Figure 15.A.2	Sample trajectory of particle filtering for robot localization	681
Figure 15.A.3	Kalman filters for the SLAM problem	682
Figure 15.A.4	Collapsed particle filtering for SLAM	684
Box 16.A	Skill: Design and Evaluation of Learning Procedures	705
Algorithm 16.A.1	Algorithms for holdout and cross-validation tests	707
Box 16.B	Concept: PAC-bounds	708
Box 17.A	Concept: Naive Bayes Classifier	727
Box 17.B	Concept: Nonparametric Models	730
Box 17.C	Case Study: Learning the ICU-Alarm Network	749
Figure 17.C.1	The ICU-Alarm Bayesian network	750
Figure 17.C.2	Learning curve for parameter estimation for the ICU-Alarm network	751
Box 17.D	Concept: Representation Independence	752
Box 17.E	Concept: Bag-of-Word Models for Text Classification	766
Figure 17.E.1	Different plate models for text	768
Box 18.A	Skill: Practical Collection of Sufficient Statistics	819
Box 18.B	Concept: Dependency Networks	822
Box 18.C	Case Study: Bayesian Networks for Collaborative Filtering	823
Figure 18.C.1	Learned Bayesian network for collaborative filtering	823
Box 19.A	Case Study: Discovering User Clusters	877
Figure 19.A.1	Application of Bayesian clustering to collaborative filtering	878
Box 19.B	Case Study: EM in Practice	885
Figure 19.B.1	Convergence of EM run on the ICU Alarm network	885
Figure 19.B.2	Local maxima in likelihood surface	886
Box 19.C	Skill: Practical Considerations in Parameter Learning	888
Box 19.D	Case Study: EM for Robot Mapping	892
Figure 19.D.1	Sample results from EM-based 3D plane mapping	893

Box 19.E	Skill: Sampling from a Dirichlet distribution	900
Box 19.F	Concept: Laplace Approximation	909
Box 19.G	Case Study: Evaluating Structure Scores	915
Figure 19.G.1	Evaluation of structure scores for a naive Bayes clustering model	916
Box 20.A	Concept: Generative and Discriminative Models for Sequence Labeling	952
Figure 20.A.1	Different models for sequence labeling: HMM, MEMM, and CRF	953
Box 20.B	Case Study: CRFs for Protein Structure Prediction	968
Box 21.A	Case Study: Identifying the Effect of Smoking on Cancer	1021
Figure 21.A.1	Three candidate models for smoking and cancer.	1022
Figure 21.A.2	Determining causality between smoking and cancer.	1023
Box 21.B	Case Study: The Effect of Cholestyramine	1033
Box 21.C	Case Study: Persistence Networks for Diagnosis	1037
Box 21.D	Case Study: Learning Cellular Networks from Intervention Data	1046
Box 22.A	Case Study: Prenatal Diagnosis	1079
Figure 22.A.1	Typical utility function decomposition for prenatal diagnosis	1080
Box 22.B	Case Study: Utility Elicitation in Medical Diagnosis	1080
Box 23.A	Case Study: Decision Making for Prenatal Testing	1094
Box 23.B	Case Study: Coordination Graphs for Robot Soccer	1117
Box 23.C	Case Study: Decision Making for Troubleshooting	1125

1 Introduction

1.1 Motivation

Most tasks require a person or an automated system to *reason*: to take the available information and reach conclusions, both about what might be true in the world and about how to act. For example, a doctor needs to take information about a patient — his symptoms, test results, personal characteristics (gender, weight) — and reach conclusions about what diseases he may have and what course of treatment to undertake. A mobile robot needs to synthesize data from its sonars, cameras, and other sensors to conclude where in the environment it is and how to move so as to reach its goal without hitting anything. A speech-recognition system needs to take a noisy acoustic signal and infer the words spoken that gave rise to it.

In this book, we describe a general framework that can be used to allow a computer system to answer questions of this type. In principle, one could write a special-purpose computer program for every domain one encounters and every type of question that one may wish to answer. The resulting system, although possibly quite successful at its particular task, is often very brittle: If our application changes, significant changes may be required to the program. Moreover, this general approach is quite limiting, in that it is hard to extract lessons from one successful solution and apply it to one which is very different.

We focus on a different approach, based on the concept of a *declarative representation*. In this approach, we construct, within the computer, a *model* of the system about which we would like to reason. This model encodes our knowledge of how the system works in a computer-readable form. This representation can be manipulated by various algorithms that can answer questions based on the model. For example, a model for medical diagnosis might represent our knowledge about different diseases and how they relate to a variety of symptoms and test results. A reasoning algorithm can take this model, as well as observations relating to a particular patient, and answer questions relating to the patient's diagnosis. **The key property of a declarative representation is the separation of knowledge and reasoning. The representation has its own clear semantics, separate from the algorithms that one can apply to it. Thus, we can develop a general suite of algorithms that apply any model within a broad class, whether in the domain of medical diagnosis or speech recognition. Conversely, we can improve our model for a specific application domain without having to modify our reasoning algorithms constantly.**



declarative
representation
model

Declarative representations, or model-based methods, are a fundamental component in many fields, and models come in many flavors. Our focus in this book is on models for complex sys-

uncertainty

tems that involve a significant amount of *uncertainty*. Uncertainty appears to be an inescapable aspect of most real-world applications. It is a consequence of several factors. We are often uncertain about the true state of the system because our observations about it are partial: only some aspects of the world are observed; for example, the patient's true disease is often not directly observable, and his future prognosis is never observed. Our observations are also noisy — even those aspects that are observed are often observed with some error. The true state of the world is rarely determined with certainty by our limited observations, as most relationships are simply not deterministic, at least relative to our ability to model them. For example, there are few (if any) diseases where we have a clear, universally true relationship between the disease and its symptoms, and even fewer such relationships between the disease and its prognosis. Indeed, while it is not clear whether the universe (quantum mechanics aside) is deterministic when modeled at a sufficiently fine level of granularity, it is quite clear that it is not deterministic relative to our current understanding of it. To summarize, uncertainty arises because of limitations in our ability to observe the world, limitations in our ability to model it, and possibly even because of innate nondeterminism.

Because of this ubiquitous and fundamental uncertainty about the true state of world, we need to allow our reasoning system to consider different possibilities. One approach is simply to consider any state of the world that is possible. Unfortunately, it is only rarely the case that we can completely eliminate a state as being impossible given our observations. In our medical diagnosis example, there is usually a huge number of diseases that are *possible* given a particular set of observations. Most of them, however, are highly unlikely. If we simply list all of the possibilities, our answers will often be vacuous of meaningful content (e.g., “the patient can have any of the following 573 diseases”). **Thus, to obtain meaningful conclusions, we need to reason not just about what is possible, but also about what is *probable*.**



probability theory

The calculus of *probability theory* (see section 2.1) provides us with a formal framework for considering multiple possible outcomes and their likelihood. It defines a set of mutually exclusive and exhaustive possibilities, and associates each of them with a *probability* — a number between 0 and 1, so that the total probability of all possibilities is 1. This framework allows us to consider options that are unlikely, yet not impossible, without reducing our conclusions to content-free lists of every possibility.



Furthermore, one finds that probabilistic models are very liberating. Where in a more rigid formalism we might find it necessary to enumerate every possibility, here we can often sweep a multitude of annoying exceptions and special cases under the “probabilistic rug,” by introducing outcomes that roughly correspond to “something unusual happens.” In fact, as we discussed, this type of approximation is often inevitable, as we can only rarely (if ever) provide a deterministic specification of the behavior of a complex system. Probabilistic models allow us to make this fact explicit, and therefore often provide a model which is more faithful to reality.

1.2 Structured Probabilistic Models

This book describes a general-purpose framework for constructing and using probabilistic models of complex systems. We begin by providing some intuition for the principles underlying this framework, and for the models it encompasses. This section requires some knowledge of

basic concepts in probability theory; a reader unfamiliar with these concepts might wish to read section 2.1 first.

random variable

Complex systems are characterized by the presence of multiple interrelated aspects, many of which relate to the reasoning task. For example, in our medical diagnosis application, there are multiple possible diseases that the patient might have, dozens or hundreds of symptoms and diagnostic tests, personal characteristics that often form predisposing factors for disease, and many more matters to consider. These domains can be characterized in terms of a set of *random variables*, where the value of each variable defines an important property of the world. For example, a particular disease, such as *Flu*, may be one variable in our domain, which takes on two values, for example, *present* or *absent*; a symptom, such as *Fever*, may be a variable in our domain, one that perhaps takes on continuous values. The set of possible variables and their values is an important design decision, and it depends strongly on the questions we may wish to answer about the domain.

joint probability distribution

Our task is to reason probabilistically about the values of one or more of the variables, possibly given observations about some others. In order to do so using principled probabilistic reasoning, we need to construct a *joint distribution* over the space of possible assignments to some set of random variables \mathcal{X} . This type of model allows us to answer a broad range of interesting queries. For example, we can make the observation that a variable X_i takes on the specific value x_i , and ask, in the resulting *posterior distribution*, what the probability distribution is over values of another variable X_j .

posterior distribution

Example 1.1

Consider a very simple medical diagnosis setting, where we focus on two diseases — flu and hayfever; these are not mutually exclusive, as a patient can have either, both, or none. Thus, we might have two binary-valued random variables, Flu and Hayfever. We also have a 4-valued random variable Season, which is correlated both with flu and hayfever. We may also have two symptoms, Congestion and Muscle Pain, each of which is also binary-valued. Overall, our probability space has $2 \times 2 \times 4 \times 2 \times 2 = 64$ values, corresponding to the possible assignments to these five variables. Given a joint distribution over this space, we can, for example, ask questions such as how likely the patient is to have the flu given that it is fall, and that she has sinus congestion but no muscle pain; as a probability expression, this query would be denoted

$$P(\text{Flu} = \text{true} \mid \text{Season} = \text{fall}, \text{Congestion} = \text{true}, \text{Muscle Pain} = \text{false}).$$

■

1.2.1 Probabilistic Graphical Models

Specifying a joint distribution over 64 possible values, as in example 1.1, already seems fairly daunting. When we consider the fact that a typical medical- diagnosis problem has dozens or even hundreds of relevant attributes, the problem appears completely intractable. This book describes the framework of probabilistic graphical models, which provides a mechanism for exploiting structure in complex distributions to describe them compactly, and in a way that allows them to be constructed and utilized effectively.

Probabilistic graphical models use a graph-based representation as the basis for compactly encoding a complex distribution over a high-dimensional space. In this graphical representation, illustrated in figure 1.1, the nodes (or ovals) correspond to the variables in our domain, and the edges correspond to direct probabilistic interactions between them. For example, figure 1.1a (top)

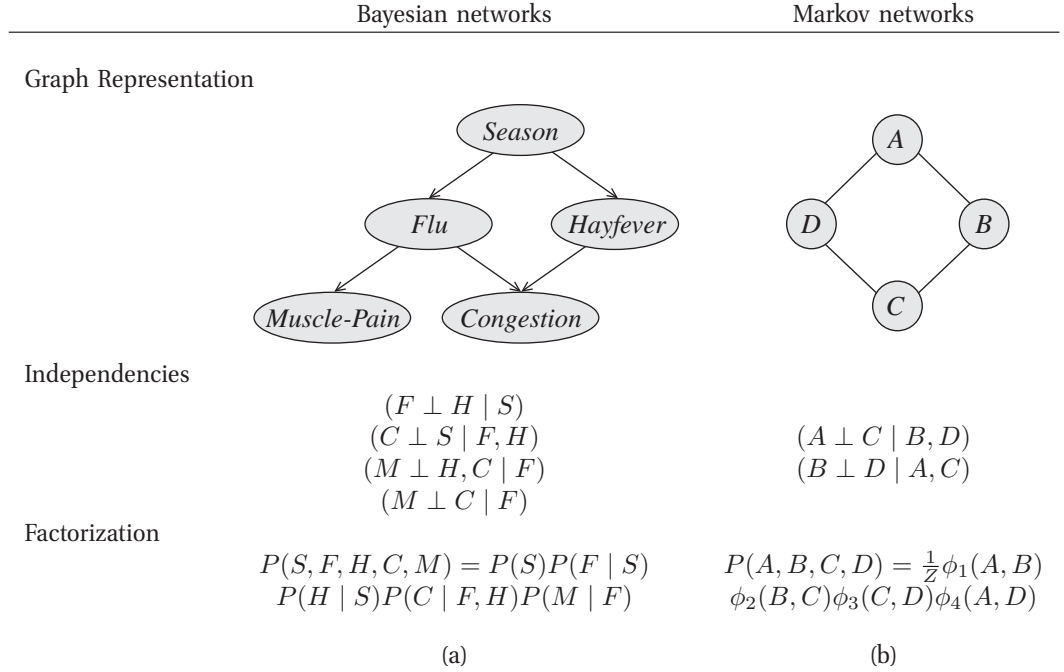


Figure 1.1 Different perspectives on probabilistic graphical models: top — the graphical representation; middle — the independencies induced by the graph structure; bottom — the factorization induced by the graph structure. (a) A sample Bayesian network. (b) A sample Markov network.

illustrates one possible graph structure for our flu example. In this graph, we see that there is no direct interaction between *Muscle Pain* and *Season*, but both interact directly with *Flu*.

There is a dual perspective that one can use to interpret the structure of this graph. From one perspective, the graph is a compact representation of a set of *independencies* that hold in the distribution; these properties take the form \mathbf{X} is independent of \mathbf{Y} given \mathbf{Z} , denoted $(\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z})$, for some subsets of variables $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$. For example, our “target” distribution P for the preceding example — the distribution encoding our beliefs about this particular situation — may satisfy the conditional independence $(\text{Congestion} \perp \text{Season} \mid \text{Flu}, \text{Hayfever})$. This statement asserts that

$$P(\text{Congestion} \mid \text{Flu}, \text{Hayfever}, \text{Season}) = P(\text{Congestion} \mid \text{Flu}, \text{Hayfever});$$

that is, if we are interested in the distribution over the patient having congestion, and we know whether he has the flu and whether he has hayfever, the season is no longer informative. Note that this assertion does *not* imply that *Season* is independent of *Congestion*; only that all of the information we may obtain from the season on the chances of having congestion we already obtain by knowing whether the patient has the flu and has hayfever. Figure 1.1a (middle) shows the set of independence assumptions associated with the graph in figure 1.1a (top).

factor

The other perspective is that the graph defines a skeleton for compactly representing a high-dimensional distribution: Rather than encode the probability of every possible assignment to all of the variables in our domain, we can “break up” the distribution into smaller *factors*, each over a much smaller space of possibilities. We can then define the overall joint distribution as a product of these factors. For example, figure 1.1(a-bottom) shows the factorization of the distribution associated with the graph in figure 1.1 (top). It asserts, for example, that the probability of the event “spring, no flu, hayfever, sinus congestion, muscle pain” can be obtained by multiplying five numbers: $P(\text{Season} = \text{spring})$, $P(\text{Flu} = \text{false} \mid \text{Season} = \text{spring})$, $P(\text{Hayfever} = \text{true} \mid \text{Season} = \text{spring})$, $P(\text{Congestion} = \text{true} \mid \text{Hayfever} = \text{true}, \text{Flu} = \text{false})$, and $P(\text{Muscle Pain} = \text{true} \mid \text{Flu} = \text{false})$. This parameterization is significantly more compact, requiring only $3 + 4 + 4 + 4 + 2 = 17$ nonredundant parameters, as opposed to 63 nonredundant parameters for the original joint distribution (the 64th parameter is fully determined by the others, as the sum over all entries in the joint distribution must sum to 1). The graph structure defines the factorization of a distribution P associated with it — the set of factors and the variables that they encompass.



It turns out that these two perspectives — the graph as a representation of a set of independencies, and the graph as a skeleton for factorizing a distribution — are, in a deep sense, equivalent. The independence properties of the distribution are precisely what allow it to be represented compactly in a factorized form. Conversely, a particular factorization of the distribution guarantees that certain independencies hold.

Bayesian network

Markov network

We describe two families of graphical representations of distributions. One, called *Bayesian networks*, uses a directed graph (where the edges have a source and a target), as shown in figure 1.1a (top). The second, called *Markov networks*, uses an undirected graph, as illustrated in figure 1.1b (top). It too can be viewed as defining a set of independence assertions (figure 1.1b [middle] or as encoding a compact factorization of the distribution (figure 1.1b [bottom])). Both representations provide the duality of independencies and factorization, but they differ in the set of independencies they can encode and in the factorization of the distribution that they induce.

1.2.2 Representation, Inference, Learning

The graphical language exploits structure that appears present in many distributions that we want to encode in practice: the property that variables tend to interact *directly* only with very few others. Distributions that exhibit this type of structure can generally be encoded naturally and compactly using a graphical model.

This framework has many advantages. First, it often allows the distribution to be written down tractably, even in cases where the explicit representation of the joint distribution is astronomically large. Importantly, the type of representation provided by this framework is *transparent*, in that a human expert can understand and evaluate its semantics and properties. This property is important for constructing models that provide an accurate reflection of our understanding of a domain. Models that are opaque can easily give rise to unexplained, and even undesirable, answers.

inference

Second, as we show, the same structure often also allows the distribution to be used effectively for *inference* — answering queries using the distribution as our model of the world. In particular, we provide algorithms for computing the posterior probability of some variables given evidence

on others. For example, we might observe that it is spring and the patient has muscle pain, and we wish to know how likely he is to have the flu, a query that can formally be written as $P(\text{Flu} = \text{true} \mid \text{Season} = \text{spring}, \text{Muscle Pain} = \text{true})$. These inference algorithms work directly on the graph structure and are generally orders of magnitude faster than manipulating the joint distribution explicitly.

data-driven
approach

Third, this framework facilitates the effective construction of these models, whether by a human expert or automatically, by *learning* from data a model that provides a good approximation to our past experience. For example, we may have a set of patient records from a doctor's office and wish to learn a probabilistic model encoding a distribution consistent with our aggregate experience. Probabilistic graphical models support a *data-driven approach* to model construction that is very effective in practice. In this approach, a human expert provides some rough guidelines on how to model a given domain. For example, the human usually specifies the attributes that the model should contain, often some of the main dependencies that it should encode, and perhaps other aspects. The details, however, are usually filled in automatically, by fitting the model to data. The models produced by this process are usually much better reflections of the domain than models that are purely hand-constructed. Moreover, they can sometimes reveal surprising connections between variables and provide novel insights about a domain.



These three components — representation, inference, and learning — are critical components in constructing an intelligent system. We need a declarative representation that is a reasonable encoding of our world model. We need to be able to use this representation effectively to answer a broad range of questions that are of interest. And we need to be able to acquire this distribution, combining expert knowledge and accumulated data. Probabilistic graphical models are one of a small handful of frameworks that support all three capabilities for a broad range of problems.

1.3 Overview and Roadmap

1.3.1 Overview of Chapters

The framework of probabilistic graphical models is quite broad, and it encompasses both a variety of different types of models and a range of methods relating to them. This book describes several types of models. For each one, we describe the three fundamental cornerstones: representation, inference, and learning.

We begin in part I, by describing the most basic type of graphical models, which are the focus of most of the book. These models encode distributions over a fixed set \mathcal{X} of random variables. We describe how graphs can be used to encode distributions over such spaces, and what the properties of such distributions are.

Specifically, in chapter 3, we describe the Bayesian network representation, based on directed graphs. We describe how a Bayesian network can encode a probability distribution. We also analyze the independence properties induced by the graph structure.

In chapter 4, we move to Markov networks, the other main category of probabilistic graphical models. Here also we describe the independencies defined by the graph and the induced factorization of the distribution. We also discuss the relationship between Markov networks and Bayesian networks, and briefly describe a framework that unifies both.

In chapter 5, we delve a little deeper into the representation of the parameters in probabilistic

models, focusing mostly on Bayesian networks, whose parameterization is more constrained. We describe representations that capture some of the finer-grained structure of the distribution, and show that, here also, capturing structure can provide significant gains.

In chapter 6, we turn to formalisms that extend the basic framework of probabilistic graphical models to settings where the set of variables is no longer rigidly circumscribed in advance. One such setting is a *temporal* one, where we wish to model a system whose state evolves over time, requiring us to consider distributions over entire trajectories. We describe a compact representation — a *dynamic Bayesian network* — that allows us to represent structured systems that evolve over time. We then describe a family of extensions that introduce various forms of higher level structure into the framework of probabilistic graphical models. Specifically, we focus on domains containing *objects* (whether concrete or abstract), characterized by attributes, and related to each other in various ways. Such domains can include repeated structure, since different objects of the same type share the same probabilistic model. These languages provide a significant extension to the expressive power of the standard graphical models.

In chapter 7, we take a deeper look at models that include continuous variables. Specifically, we explore the properties of the multivariate Gaussian distribution and the representation of such distributions as both directed and undirected graphical models. Although the class of Gaussian distributions is a limited one and not suitable for all applications, it turns out to play a critical role even when dealing with distributions that are not Gaussian.

In chapter 8, we take a deeper, more technical look at probabilistic models, defining a general framework called the *exponential family*, that encompasses a broad range of distributions. This chapter provides some basic concepts and tools that will turn out to play an important role in later development.

We then turn, in part II, to a discussion of the inference task. In chapter 9, we describe the basic ideas underlying exact inference in probabilistic graphical models. We first analyze the fundamental difficulty of the exact inference task, separately from any particular inference algorithm we might develop. We then present two basic algorithms for exact inference — variable elimination and conditioning — both of which are equally applicable to both directed and undirected models. Both of these algorithms can be viewed as operating over the graph structure defined by the probabilistic model. They build on basic concepts, such as graph properties and dynamic programming algorithms, to provide efficient solutions to the inference task. We also provide an analysis of their computational cost in terms of the graph structure, and we discuss where exact inference is feasible.

In chapter 10, we describe an alternative view of exact inference, leading to a somewhat different algorithm. The benefit of this alternative algorithm is twofold. First, it uses dynamic programming to avoid repeated computations in settings where we wish to answer more than a single query using the same network. Second, it defines a natural algorithm that uses message passing on a graph structure; this algorithm forms the basis for approximate inference algorithms developed in later chapters.

Because exact inference is computationally intractable for many models of interest, we then proceed to describe approximate inference algorithms, which trade off accuracy with computational cost. We present two main classes of such algorithms. In chapter 11, we describe a class of methods that can be viewed from two very different perspectives: On one hand, they are direct generalizations of the graph-based message-passing approach developed for the case of exact inference in chapter 10. On the other hand, they can be viewed as solving an optimization

problem: one where we approximate the distribution of interest using a simpler representation that allows for feasible inference. The equivalence of these views provides important insights and suggests a broad family of algorithms that one can apply to approximate inference.

In chapter 12, we describe a very different class of methods: *particle-based methods*, which approximate a complex joint distribution by considering samples from it (also known as particles). We describe several methods from this general family. These methods are generally based on core techniques from statistics, such as importance sampling and Markov-chain Monte Carlo methods. Once again, the connection to this general class of methods suggests multiple opportunities for new algorithms.

While the representation of probabilistic graphical models applies, to a great extent, to models including both discrete and continuous-valued random variables, inference in models involving continuous variables is significantly more challenging than the purely discrete case. In chapter 14, we consider the task of inference in continuous and *hybrid* (continuous/discrete) networks, and we discuss whether and how the exact and approximate inference methods developed in earlier chapters can be applied in this setting.

The representation that we discussed in chapter 6 allows a compact encoding of networks whose size can be unboundedly large. Such networks pose particular challenges to inference algorithms. In this chapter, we discuss some special-purpose methods that have been developed for the particular settings of networks that model dynamical systems.

We then turn, in part III, to the third of our main topics — learning probabilistic models from data. We begin in chapter 16 by reviewing some of the fundamental concepts underlying the general task of learning models from data. We then present the spectrum of learning problems that we address in this part of the book. These problems vary along two main axes: the extent to which we are given prior knowledge specifying the model, and whether the data from which we learn contain complete observations of all of the relevant variables. In contrast to the inference task, where the same algorithms apply equally to Bayesian networks and Markov networks, the learning task is quite different for these two classes of models. We begin with studying the learning task for Bayesian networks.

In chapter 17, we focus on the most basic learning task: learning parameters for a Bayesian network with a given structure, from fully observable data. Although this setting may appear somewhat restrictive, it turns out to form the basis for our entire development of Bayesian network learning. As we show, the factorization of the distribution, which was central both to representation and to inference, also plays a key role in making inference feasible.

We then move, in chapter 18, to the harder problem of learning both Bayesian network structure and the parameters, still from fully observed data. The learning algorithms we present trade off the accuracy with which the learned network represents the empirical distribution for the complexity of the resulting structure. As we show, the type of independence assumptions underlying the Bayesian network representation often hold, at least approximately, in real-world distributions. Thus, these learning algorithms often result in reasonably compact structures that capture much of the signal in the distribution.

In chapter 19, we address the Bayesian network learning task in a setting where we have access only to partial observations of the relevant variables (for example, when the available patient records have missing entries). This type of situation occurs often in real-world settings. Unfortunately, the resulting learning task is considerably harder, and the resulting algorithms are both more complex and less satisfactory in terms of their performance.

We conclude the discussion of learning in chapter 20 by considering the problem of learning Markov networks from data. It turns out that the learning tasks for Markov networks are significantly harder than the corresponding problem for Bayesian networks. We explain the difficulties and discuss the existing solutions.

Finally, in part IV, we turn to a different type of extension, where we consider the use of this framework for other forms of reasoning. Specifically, we consider cases where we can act, or intervene, in the world.

In chapter 21, we focus on the semantics of intervention and its relation to causality. We present the notion of a *causal model*, which allows us to answer not only queries of the form “if I observe X, what do I learn about Y,” but also *intervention queries*, of the form “if I manipulate X, what effect does it have on Y.”

We then turn to the task of *decision making* under uncertainty. Here, we must consider not only the distribution over different states of the world, but also the preferences of the agent regarding these outcomes. In chapter 22, we discuss the notion of *utility functions* and how they can encode an agent’s preferences about complex situations involving multiple variables. As we show, the same ideas that we used to provide compact representations of probability distribution can also be used for utility functions.

In chapter 23, we describe a unified representation for decision making, called *influence diagrams*. Influence diagrams extend Bayesian networks by introducing actions and utilities. We present algorithms that use influence diagrams for making decisions that optimize the agent’s expected utility. These algorithms utilize many of the same ideas that formed the basis for exact inference in Bayesian networks.

We conclude with a high-level synthesis of the techniques covered in this book, and with some guidance on how to use them in tackling a new problem.

1.3.2 Reader’s Guide

As we mentioned, the topics described in this book relate to multiple fields, and techniques from other disciplines — probability theory, computer science, information theory, optimization, statistics, and more — are used in various places throughout it. While it is impossible to present all of the relevant material within the scope of this book, we have attempted to make the book somewhat self-contained by providing a very brief review of the key concepts from these related disciplines in chapter 2.

Some of this material, specifically the review of probability theory and of graph-related concepts, is very basic yet central to most of the development in this book. Readers who are less familiar with these topics may wish to read these sections carefully, and even knowledgeable readers may wish to briefly review them to gain familiarity with the notations used. Other background material, covering such topics as information theory, optimization, and algorithmic concepts, can be found in the appendix.

The chapters in the book are structured as follows. The main text in each chapter provides the detailed technical development of the key ideas. Beyond the main text, most chapters contain boxes that contain interesting material that augments these ideas. These boxes come in three types: *Skill boxes* describe “hands-on” tricks and techniques, which, while often heuristic in nature, are important for getting the basic algorithms described in the text to work in practice. *Case study boxes* describe empirical case studies relating to the techniques described in the text.

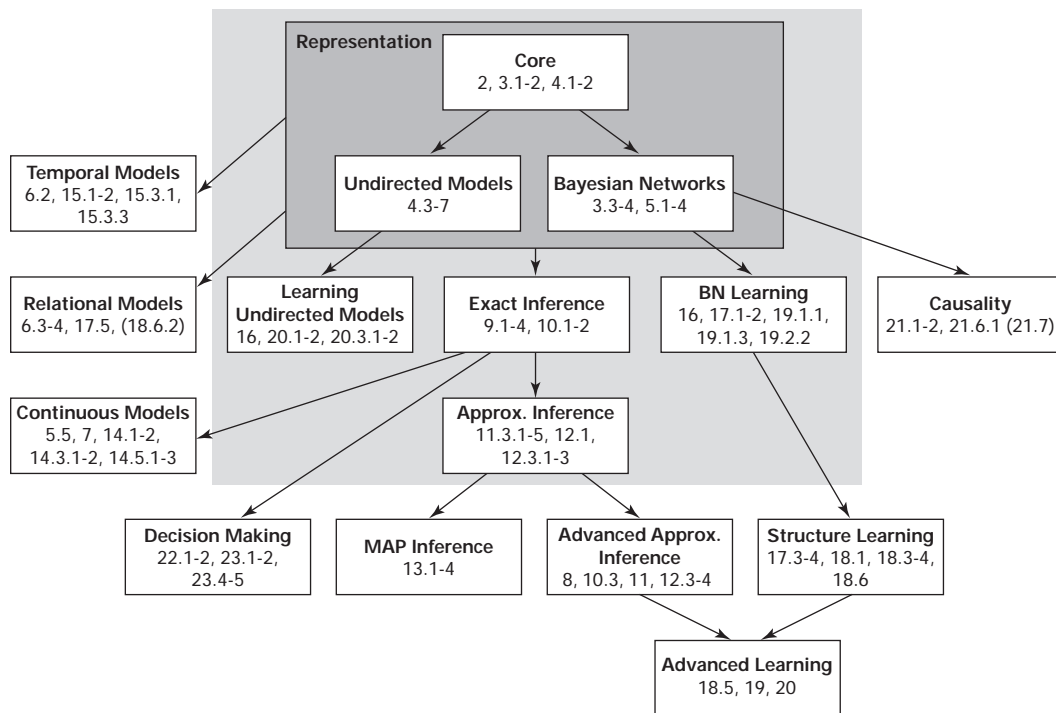


Figure 1.2 A reader's guide to the structure and dependencies in this book

These case studies include both empirical results on how the algorithms perform in practice and descriptions of applications of these algorithms to interesting domains, illustrating some of the issues encountered in practice. Finally, *concept boxes* present particular instantiations of the material described in the text, which have had significant impact in their own right.

This textbook is clearly too long to be used in its entirety in a one-semester class. Figure 1.2 tries to delineate some coherent subsets of the book that can be used for teaching and other purposes. The small, labeled boxes represent “units” of material on particular topics. Arrows between the boxes represent dependencies between these units. The first enclosing box (solid line) represents material that is fundamental to everything else, and that should be read by anyone using this book. One can then use the dependencies between the boxes to expand or reduce the depth of the coverage on any given topic. The material in the larger box (dashed line) forms a good basis for a one-semester (or even one-quarter) overview class. Some of the sections in the book are marked with an asterisk, denoting the fact that they contain more technically advanced material. In most cases, these sections are self-contained, and they can be skipped without harming the reader’s ability to understand the rest of the text.

We have attempted in this book to present a synthesis of ideas, most of which have been developed over many years by multiple researchers. To avoid futile attempts to divide up the credit precisely, we have omitted all bibliographical references from the technical presentation

in the chapters. Rather, each chapter ends with a section called “Relevant Literature,” which describes the historical evolution of the material in the chapter, acknowledges the papers and books that developed the key concepts, and provides some additional readings on material relevant to the chapter. We encourage the reader who is interested in a topic to follow up on some of these additional readings, since there are many interesting developments that we could not cover in this book.

Finally, each chapter includes a set of exercises that explore in additional depth some of the material described in the text and present some extensions to it. The exercises are annotated with an asterisk for exercises that are somewhat more difficult, and with two asterisks for ones that are truly challenging.

Additional material related to this book, including slides and figures, solutions to some of the exercises, and errata, can be found online at <http://pgm.stanford.edu>.

1.3.3 Connection to Other Disciplines

The ideas we describe in this book are connected to many fields. From probability theory, we inherit the basic concept of a probability distribution, as well as many of the operations we can use to manipulate it. From computer science, we exploit the key idea of using a graph as a data structure, as well as a variety of algorithms for manipulating graphs and other data structures. These algorithmic ideas and the ability to manipulate probability distributions using discrete data structures are some of the key elements that make the probabilistic manipulations tractable. Decision theory extends these basic ideas to the task of decision making under uncertainty and provides the formal foundation for this task.

From computer science, and specifically from artificial intelligence, these models inherit the idea of using a declarative representation of the world to separate procedural reasoning from our domain knowledge. This idea is of key importance to the generality of this framework and its applicability to such a broad range of tasks.

Various ideas from other disciplines also arise in this field. Statistics plays an important role both in certain aspects of the representation and in some of the work on learning models from data. Optimization plays a role in providing algorithms both for approximate inference and for learning models from data. Bayesian networks first arose, albeit in a restricted way, in the setting of modeling genetic inheritance in human family trees; in fact, restricted version of some of the exact inference algorithms we discuss were first developed in this context. Similarly, undirected graphical models first arose in physics as a model for systems of electrons, and some of the basic concepts that underlie recent work on approximate inference developed from that setting.

Information theory plays a dual role in its interaction with this field. Information-theoretic concepts such as entropy and information arise naturally in various settings in this framework, such as evaluating the quality of a learned model. Thus, tools from this discipline are a key component in our analytic toolkit. On the other side, the recent successes in coding theory, based on the relationship between inference in probabilistic models and the task of decoding messages sent over a noisy channel, have led to a resurgence of work on approximate inference in graphical models. The resulting developments have revolutionized both the development of error-correcting codes and the theory and practice of approximate message-passing algorithms in graphical models.

1.3.3.1 What Have We Gained?

Although the framework we describe here shares common elements with a broad range of other topics, it has a coherent common core: the use of structure to allow a compact representation, effective reasoning, and feasible learning of general-purpose, factored, probabilistic models. These elements provide us with a general infrastructure for reasoning and learning about complex domains.

As we discussed earlier, by using a declarative representation, we essentially separate out the description of the model for the particular application, and the general-purpose algorithms used for inference and learning. Thus, this framework provides a general algorithmic toolkit that can be applied to many different domains.

Indeed, probabilistic graphical models have made a significant impact on a broad spectrum of real-world applications. For example, these models have been used for medical and fault diagnosis, for modeling human genetic inheritance of disease, for segmenting and denoising images, for decoding messages sent over a noisy channel, for revealing genetic regulatory processes, for robot localization and mapping, and more. Throughout this book, we will describe how probabilistic graphical models were used to address these applications and what issues arise in the application of these models in practice.

In addition to practical applications, these models provide a formal framework for a variety of fundamental problems. For example, the notion of conditional independence and its explicit graph-based representation provide a clear formal semantics for irrelevance of information. This framework also provides a general methodology for handling data fusion — we can introduce *sensor variables* that are noisy versions of the true measured quantity, and use Bayesian conditioning to combine the different measurements. The use of a probabilistic model allows us to provide a formal measure for model quality, in terms of a numerical fit of the model to observed data; this measure underlies much of our work on learning models from data. The temporal models we define provide a formal framework for defining a general trend toward persistence of state over time, in a way that does not raise inconsistencies when change does occur.

In general, part of the rich development in this field is due to the close and continuous interaction between theory and practice. In this field, unlike many others, the distance between theory and practice is quite small, and there is a constant flow of ideas and problems between them. Problems or ideas arise in practical applications and are analyzed and subsequently developed in more theoretical papers. Algorithms for which no theoretical analysis exists are tried out in practice, and the profile of where they succeed and fail often provides the basis for subsequent analysis. This rich synergy leads to a continuous and vibrant development, and it is a key factor in the success of this area.

1.4 Historical Notes

The foundations of probability theory go back to the sixteenth century, when Gerolamo Cardano began a formal analysis of games of chance, followed by additional key developments by Pierre de Fermat and Blaise Pascal in the seventeenth century. The initial development involved only discrete probability spaces, and the analysis methods were purely combinatorial. The foundations of modern probability theory, with its measure-theoretic underpinnings, were laid by Andrey Kolmogorov in the 1930s.

Particularly central to the topics of this book is the so-called *Bayes theorem*, shown in the eighteenth century by the Reverend Thomas Bayes (Bayes 1763). This theorem allows us to use a model that tells us the conditional probability of event a given event b (say, a symptom given a disease) in order to compute the contrapositive: the conditional probability of event b given event a (the disease given the symptom). This type of reasoning is central to the use of graphical models, and it explains the choice of the name *Bayesian network*.

The notion of representing the interactions between variables in a multidimensional distribution using a graph structure originates in several communities, with very different motivations. In the area of statistical physics, this idea can be traced back to Gibbs (1902), who used an undirected graph to represent the distribution over a system of interacting particles. In the area of genetics, this idea dates back to the work on path analysis of Sewall Wright (Wright 1921, 1934). Wright proposed the use of a directed graph to study inheritance in natural species. This idea, although largely rejected by statisticians at the time, was subsequently adopted by economists and social scientists (Wold 1954; Blalock, Jr. 1971). In the field of statistics, the idea of analyzing interactions between variables was first proposed by Bartlett (1935), in the study of *contingency tables*, also known as *log-linear models*. This idea became more accepted by the statistics community in the 1960s and 70s (Vorobev 1962; Goodman 1970; Haberman 1974).

expert systems

In the field of computer science, probabilistic methods lie primarily in the realm of Artificial Intelligence (AI). The AI community first encountered these methods in the endeavor of building *expert systems*, computerized systems designed to perform difficult tasks, such as oil-well location or medical diagnosis, at an expert level. Researchers in this field quickly realized the need for methods that allow the integration of multiple pieces of evidence, and that provide support for making decisions under uncertainty. Some early systems (de Bombal et al. 1972; Gorry and Barnett 1968; Warner et al. 1961) used probabilistic methods, based on the very restricted *naive Bayes model*. This model restricts itself to a small set of possible hypotheses (e.g., diseases) and assumes that the different evidence variables (e.g., symptoms or test results) are independent given each hypothesis. These systems were surprisingly successful, performing (within their area of expertise) at a level comparable to or better than that of experts. For example, the system of de Bombal et al. (1972) averaged over 90 percent correct diagnoses of acute abdominal pain, whereas expert physicians were averaging around 65 percent.

Despite these successes, this approach fell into disfavor in the AI community, owing to a combination of several factors. One was the belief, prevalent at the time, that artificial intelligence should be based on similar methods to human intelligence, combined with a strong impression that people do not manipulate numbers when reasoning. A second issue was the belief that the strong independence assumptions made in the existing expert systems were fundamental to the approach. Thus, the lack of a flexible, scalable mechanism to represent interactions between variables in a distribution was a key factor in the rejection of the probabilistic framework.

The rejection of probabilistic methods was accompanied by the invention of a range of alternative formalisms for reasoning under uncertainty, and the construction of expert systems based on these formalisms (notably Prospector by Duda, Gaschnig, and Hart 1979 and Mycin by Buchanan and Shortliffe 1984). Most of these formalisms used the production rule framework, where each rule is augmented with some number(s) defining a measure of “confidence” in its validity. These frameworks largely lacked formal semantics, and many exhibited significant problems in key reasoning patterns. Other frameworks for handling uncertainty proposed at the time included fuzzy logic, possibility theory, and Dempster-Shafer belief functions. For a