Stochastic processes on graphs with cycles: geometric and variational approaches

by

Martin J. Wainwright

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

January, 2002
[June 2002]

© 2002 Massachusetts Institute of Technology
All Rights Reserved.

Signature of Author: _____
Department of Electrical Engineering and Computer Science
January 28, 2002

Certified by: _____
Alan S. Willsky
Professor of EECS
Thesis Supervisor

Certified by: _____
Tommi S. Jaakkola
Assistant Professor of EECS
Thesis Supervisor

Accepted by: _____
Arthur C. Smith
Professor of Electrical Engineering
Chair, Committee for Graduate Students

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
APR 16 2002
LIBRARIES

ARCHIVES

Stochastic processes on graphs with cycles: geometric and variational approaches

by Martin J. Wainwright

Submitted to the Department of Electrical Engineering
and Computer Science on January 28, 2002
in Partial Fulfillment of the Requirements for the Degree
of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Stochastic processes defined on graphs arise in a tremendous variety of fields, including statistical physics, signal processing, computer vision, artificial intelligence, and information theory. The formalism of graphical models provides a useful language with which to formulate fundamental problems common to all of these fields, including estimation, model fitting, and sampling. For graphs without cycles, known as trees, all of these problems are relatively well-understood, and can be solved efficiently with algorithms whose complexity scales in a tractable manner with problem size. In contrast, these same problems present considerable challenges in general graphs with cycles.

The focus of this thesis is the development and analysis of methods, both exact and approximate, for problems on graphs with cycles. Our contributions are in developing and analyzing techniques for estimation, as well as methods for computing upper and lower bounds on quantities of interest (e.g., marginal probabilities; partition functions). In order to do so, we make use of exponential representations of distributions, as well as insight from the associated information geometry and Legendre duality. Our results demonstrate the power of exponential representations for graphical models, as well as the utility of studying collections of modified problems defined on trees embedded within the original graph with cycles.

The specific contributions of this thesis include the following. We develop a method for performing exact estimation of Gaussian processes on graphs with cycles by solving a sequence of modified problems on embedded spanning trees. We introduce the tree-based reparameterization framework for approximate estimation of discrete processes. This perspective leads to a number of theoretical results on belief propagation and related algorithms, including characterizations of their fixed points and the associated approximation error. Next we extend the notion of reparameterization to a much broader class of methods for approximate inference, including Kikuchi methods, and present results on their fixed points and accuracy. Finally, we develop and analyze a novel class of upper bounds on the log partition function based on convex combinations of distributions in the exponential domain. In the special case of combining tree-structured distributions, the associated dual function gives an interesting perspective on the Bethe free energy.

Thesis Supervisors: Alan S. Willsky and Tommi S. Jaakkola
Title: Professors of Electrical Engineering and Computer Science

Notational Conventions

Symbol	Definition
General Notation	
$\lvert\cdot\rvert$	absolute value
$\lVert\cdot\rVert$	L^2 norm
∇	gradient operator
∇^2	Hessian operator
a_i	the ith component of the vector A
A_{ij}	element in the ith row and jth column of matrix A
e_k	indicator vector with 1 in the kth component and 0 everywhere else
$\mathbb{R}$	real numbers
$\mathbb{R}^N$	vector space of real-valued N-dimensional vectors
$[0,1]^N$	closed unit hypercube in $\mathbb{R}^N$
$(0,1)^N$	open unit hypercube in $\mathbb{R}^N$
$Ra(F)$	range of the mapping F
$F \circ G$	composition of mappings F and G
I	identity operator
$\mathbf{x}$	random vector
$\mathcal{X}^N$	sample space of N-dimensional random vector $\mathbf{x}$
$\mathbf{y}$	observation vector
$p(\mathbf{x})$	probability distribution on $\mathbf{x}$
$p(\mathbf{x} \mid \mathbf{y})$	conditional probability distribution of $\mathbf{x}$ given $\mathbf{y}$
$H(p)$	entropy of distribution p
$D(p \parallel q)$	Kullback-Leibler divergence between p and q
$\mathcal{N}(\mu, \Lambda)$	Gaussian distribution with mean μ and covariance Λ
$\mathcal{U}[a,b]$	uniform distribution on $[a,b]$
$\mathcal{L}$	Lagrangian of a constrained optimization problem

Symbol	Definition
Graphical models	
$\mathcal{G}$	undirected graph
$\mathcal{V}$	vertex or node set of graph
$\mathcal{E}$	edge set of graph
$\mathcal{C}$	graph clique
$\mathbf{C}$	set of all cliques of $\mathcal{G}$
$\widetilde{\mathcal{G}}$	triangulated version of $\mathcal{G}$
$\widetilde{\mathbf{C}}$	set of all clique of $\widetilde{\mathcal{G}}$
$\mathcal{S}$	separator set in a junction tree
$\mathbf{S}$	set of all separator sets
$\psi_{\mathcal{C}}$	compatibility function on clique $\mathcal{C}$
Z	partition function
N	number of nodes (i.e., $\|\mathcal{V}\|$)
m	number of discrete states
s, t	indices for nodes
(s,t)	edge between nodes s and t
j, k	indices for discrete states
$\mathcal{N}(s)$	neighbors of node s in $\mathcal{G}$
$\mathcal{T}$	embedded spanning tree of $\mathcal{G}$
$\mathcal{E}(\mathcal{T})$	edge set of $\mathcal{T}$
K_N	complete graph on N nodes

Symbol	Definition
Exponential families and information geometry	
θ	exponential parameter vector
$d(\theta)$	number of components in θ
$\phi_\mathcal{C}$	potential function
$\boldsymbol{\phi}$	collection of potential functions
$p(\mathbf{x};\theta)$	exponential distribution on $\mathbf{x}$ defined by θ
Φ	log partition function
Ψ	negative entropy function (dual to Φ)
η	mean parameters (dual variables)
Λ	Legendre mapping between θ and η
$\mathcal{M}_e$	e-flat manifold
$\mathcal{M}_m$	m-flat manifold
$D(\theta \parallel \theta^*)$	Kullback-Leibler divergence between $p(\mathbf{x};\theta)$ and $p(\mathbf{x};\theta^*)$
$\mathbb{E}_\theta[f]$	expectation of $f(\mathbf{x})$ under $p(\mathbf{x};\theta)$
$\mathrm{cov}_\theta\{f, g\}$	covariance of $f(\mathbf{x})$ and $g(\mathbf{x})$ under $p(\mathbf{x};\theta)$
$\mathrm{cum}_\theta\{f_1, \ldots, f_k\}$	k^{th}-order cumulant of $f_1(\mathbf{x}), \ldots f_k(\mathbf{x})$ under $p(\mathbf{x};\theta)$

Symbol	Definition
Tree-based reparameterization	
$\mathcal{T}^i$	embedded spanning tree
i	index for embedded spanning trees
L	total number of spanning trees used
$M_{st;k}$	belief propagation message from node s to t
$P_{s;j}$, $P_{st;jk}$	exact marginal probabilities
$T_{s;j}$, $T_{st;jk}$	approximate marginal probabilities
κ	arbitrary normalization constant
$p^i(\mathbf{x})$	tree-structured component of $p(\mathbf{x})$
$q^i(\mathbf{x})$	set of residual terms (i.e., $p(\mathbf{x})/p^i(\mathbf{x})$)
$\delta(x_s = j)$	indicator function for x_s to take value j
$\mathcal{A}$	set of composite indices $(s;j)$ and $(st;jk)$
$\mathcal{A}^i$	composite indices corresponding to $\mathcal{T}^i$
$\mathbb{C}$	constraint set for pseudomarginals
$\mathbb{C}^i$	constraint set based on tree $\mathcal{T}^i$
Θ	mapping from T to θ
$\mathcal{R}$	reparameterization operator
Π^i	projection operator onto tree $\mathcal{T}^i$
$\mathcal{I}^i$	injection operator from $\mathcal{T}^i$ to full set
$\mathcal{Q}^i$	combined reparameterization-identity mapping based on $\mathcal{T}^i$
$\{\theta^n\}$	sequence of TRP iterates
λ^n	step-size at iteration n
$i(n)$	spanning tree index at iteration n
$G(T;\theta)$	cost function (approximation to KL divergence)
$E_{s;j}$	log error $\log T_{s;j} - \log P_{s;j}$

Symbol	Definition
Advanced inference techniques	
$\mathbf{A}$	core structure
$\mathcal{G}(\mathbf{A})$	graph induced by the core structure
$Q_{\mathbf{A}}(\mathbf{x})$	approximating distribution defined by the core structure
$P_{\mathbf{A}}(\mathbf{x})$	components of original distribution over the core structure
$\mathbf{C}_{\max}(\mathbf{A})$	set of maximal cliques in $\mathbf{A}$
$\mathbf{C}_{\text{sep}}(\mathbf{A})$	set of separator sets associated with $\mathbf{A}$
$\mathbf{R}$	residual partition
Δ	particular residual element of $\mathbf{R}$
$Q_{\mathbf{A} \cup \Delta}(\mathbf{x})$	auxiliary distribution on augmented structure $\mathbf{A} \cup \Delta$
$P_{\mathbf{A} \cup \Delta}(\mathbf{x})$	components of original distribution on augmented structure $\mathbf{A} \cup \Delta$
$\widetilde{\mathbf{R}}$	augmented residual partition
$\widetilde{\Delta}$	elements of $\widetilde{\mathbf{R}}$
$\mathbb{M}$	marginalization operator
$\mathcal{G}_{\mathbf{A};\mathbf{R}}$	approximation to KL divergence based on $\mathbf{A}$ and $\mathbf{R}$
$\vec{\mathbf{Q}}$	collection of approximating distributions
M_{Υ}	core structure valued messages
$\bar{\theta}$	exponential parameter for target distribution
$\mathcal{A}(\mathbf{B})$	indices associated with elements of $\mathbf{B}$
$\boldsymbol{\phi}_{\mathbf{B}}$	$\{ \phi_\alpha \mid \alpha \in \mathcal{A}(\mathbf{B}) \}$
$\theta_{\mathbf{B}} \bullet \boldsymbol{\phi}_{\mathbf{B}}$	$\sum_{\alpha \in \mathbf{B}} \theta_\alpha \phi_\alpha$
$\Pi^{\mathbf{B}}$	projection operator of an exponential parameter onto $\mathbf{B}$
$\mathcal{I}$	injection operator into full set $\mathcal{A}$
$\theta_{\mathbf{A}}$	exponential parameter for $Q_{\mathbf{A}}$
$\theta_{\mathbf{A} \cup \Delta}$	exponential parameter for $Q_{\mathbf{A} \cup \Delta}$

Symbol	Definition
Convex upper bounds	
$\mathfrak{T}$	set of all spanning trees of $\mathcal{G}$
$\vec{\mu}$	probability distribution over trees $\mathcal{T}$
$\mu(\mathcal{T})$	probability of spanning tree $\mathcal{T}$
$\text{supp}(\vec{\mu})$	support of the distribution $\vec{\mu}$
μ_e	edge appearance probabilities $\Pr_{\vec{\mu}}\{e \in \mathcal{T}\}$
$\theta(\mathcal{T})$	exponential parameter vector structured according to tree $\mathcal{T}$
$\boldsymbol{\theta}$	collection of tree-structured exponential parameter vectors
$\mathbb{E}_{\vec{\mu}}[\boldsymbol{\theta}]$	convex combination $\sum_{\mathcal{T} \in \mathfrak{T}} \mu(\mathcal{T})\theta(\mathcal{T})$
$\mathcal{A}(\theta^*)$	set of feasible pairs $(\boldsymbol{\theta}; \vec{\mu})$ such that $\mathbb{E}_{\vec{\mu}}[\boldsymbol{\theta}] = \theta^*$
$\mathcal{Q}(\lambda; \vec{\mu}; \theta^*)$	Lagrangian dual function
λ, η	dual parameters
$\Pi^{\mathcal{T}}$	projection operator onto tree $\mathcal{T}$
$\mathbb{L}(\mathcal{G})$	set of tree-consistent mean parameters
$\mathbb{M}(\mathcal{G})$	set of globally consistent mean parameters
H_s	single-node entropy at x_s
I_{st}	mutual information between x_s and x_t
$\mathbb{T}(\mathcal{G})$	spanning tree polytope
$r(\cdot)$	rank function
$v(A)$	number of vertices touched by edges in $A \subset \mathcal{E}$
$c(A)$	number of connected components of $\mathcal{G}(A)$
$\mathcal{F}(\lambda; \boldsymbol{\mu_e}; \theta^*)$	function for optimal upper bounds
$\widehat{\lambda}(\boldsymbol{\mu_e})$	optimal set of mean parameters (as a function of $\boldsymbol{\mu_e}$)
$\widehat{\boldsymbol{\mu_e}}$	optimal set of edge appearance probabilities
$\boldsymbol{\nu}(\mathcal{T})$	edge incidence vector corresponding to spanning tree $\mathcal{T}$

Acknowledgments

During the course of my Ph.D., I have benefited from significant interactions with people both within MIT and throughout the greater academic community. I am grateful to be able to acknowledge these people and their contributions here.

First of all, I have been fortunate enough to have enjoyed the support and encouragement of not one but two thesis supervisors: Alan Willsky and Tommi Jaakkola. I doubt that anyone who interacts significantly with Alan Willsky can fail to be impressed (if not infected) by his tremendous intellectual energy and enthusiasm. I first met Alan as a student in his MIT graduate course on stochastic processes (6.432). This course greatly fueled my interest in problems of a stochastic nature, primarily due to the extraordinary clarity of his teaching, as well as his evident passion for the material. The following year, I was fortunate enough to have the opportunity to join Alan's Stochastic Systems Group (SSG) as a Ph.D. candidate. As an advisor, Alan has the impressive capacity of being able to provide the guidance and direction that graduate students (as beginning researchers) require, while simultaneously allowing them free rein to explore their developing interests. Alan also demonstrates a remarkable talent for rapidly distilling the essence of a problem, and suggesting ways of attacking it.

After my first year at MIT and as my interest in graphical models grew, I started to interact with Tommi Jaakkola more frequently. During many subsequent hours immersed in research discussions with him, I have benefited from the powerful combination of intuition and rigor in his thinking. As an advisor, he offers a rare mixture of qualities that I have appreciated: although always supportive and encouraging of my work, he is at the same time extremely exacting in assessing when our understanding of a problem is complete. His incisive questions and counterexamples have sent me back to the drawing boards more than once.

I also benefited from helpful interactions with the other members of my thesis committee. Prof. Sanjoy Mitter, after listening to one of my ideas, invariably offered sage advice as well as insightful pointers to related research. His work in organizing the 6.291 seminar in Fall 2001 was instrumental in bringing together many people within and outside LIDS, all of whom were interested in the connections between statistical physics, coding theory, and convex analysis. Prof. David Karger was a valuable resource as a resident expert on graph theory, combinatorial optimization, and randomized algorithms. In particular, I am grateful to him for directing me to the literature on the spanning tree polytope (and more generally, on matroid theory), which plays an important role in Chapter 7 of this thesis.

Earlier in my graduate career, I was fortunate to spend a summer working with Eero Simoncelli at New York University. The work that we initiated that summer

formed the basis of a productive collaboration (and friendship) that continues today. Our joint work with Alan on natural image models provided significant motivation for me to investigate graphical models, especially those with cycles, more deeply. Eero also provided me with the elderly but trusty IBM laptop on which much of this thesis was written.

I would also like to thank Prof. G. David Forney Jr., who has gone far out of his way to support my work. I am particularly indebted to him for organizing and inviting me to attend the Trieste meeting (May 2001), which gave insight into the exciting interplay between statistical physics, coding theory, and graphical models. He has also been very helpful in pointing out relevant work in the coding literature, isolating unclear aspects of my presentation, and suggesting how to clarify them.

I have enjoyed working together with Erik Sudderth; the results of Chapter 4 represent part of the fruits of this collaboration. Erik was also helpful in proofreading various parts of this thesis. Michael Schneider has been invaluable as the local expert on various topics, ranging from numerical linear algebra to public transport. Dewey Tucker has been unreasonably willing to listen to my half-baked proofs with a healthy skepticism. (If only his diet were half as healthy.) I thank Andrew Kim for his patience in providing technical assistance. Andy Tsai (PK) kept me company during countless late-night work sessions, and should be commended when he finally "does the right thing". I thank John Richards for his pellucid exposition of the James-Stein estimator, as well as for introducing me to Bug-Eyed Earl. Alex Ihler's no-nonsense approach to technical support as well as his refreshing frankness will not be forgotten. Last but never least in the SSG is Taylore Kelly, a remarkable person who is the glue holding the group together. I am thankful for her wacky sense of humor, fruit delivery service, and too many other things to list here. I have enjoyed many discussions, research and otherwise, with my late-night workmate Constantine Caramanis. I also thank him for lending his keen grammatical sense to the task of proofreading. Thank you to Yee Whye Teh and Max Welling for their generosity in sharing code for junction trees on grids. I have also benefited from interactions and discussions with numerous other people, including Andrea Montanari, Michael Jordan, David MacKay, Nati Srebro, Sekhar Tatikonda, Yair Weiss, and Jonathan Yedidia.

Finally, I would like to thank my parents, John and Patricia Wainwright, for their love and support throughout the years.

Contents

Abstract 3

Acknowledgments 11

List of Figures 19

1 Introduction 23
- 1.1 Research areas related to graphical models . . . 24
  - 1.1.1 Estimation or inference . . . 24
  - 1.1.2 Model selection . . . 25
  - 1.1.3 Sampling . . . 25
- 1.2 Principal methods . . . 26
- 1.3 Main problems and contributions . . . 26
  - 1.3.1 Inference in graphs with cycles . . . 27
  - 1.3.2 Exact inference for Gaussian processes . . . 28
  - 1.3.3 Approximate inference for discrete-valued processes . . . 29
  - 1.3.4 Upper and lower bounds . . . 31
- 1.4 Thesis overview . . . 33

2 Background 37
- 2.1 Graphical models . . . 37
  - 2.1.1 Basics of graph theory . . . 38
  - 2.1.2 Basics of graphical models . . . 41
  - 2.1.3 State estimation or inference . . . 43
  - 2.1.4 Exact inference in trees . . . 44
  - 2.1.5 Junction tree representation . . . 44
- 2.2 Exponential families and information geometry . . . 47
  - 2.2.1 Exponential representations . . . 48
  - 2.2.2 Properties of Φ . . . 52
  - 2.2.3 Riemannian geometry of exponential families . . . 53
  - 2.2.4 Legendre transform and dual variables . . . 55

2.2.5 Geometric consequences for graphical models 57
2.2.6 Kullback-Leibler divergence and Fisher information 58
2.2.7 I-projections onto flat manifolds 60
2.2.8 Geometry of I-projection 63
2.3 Variational methods and mean field 65
2.3.1 Mean field as a variational technique 66
2.3.2 Stationarity conditions for mean field 68

3 Perturbations and Bounds **71**
3.1 Introduction 71
3.1.1 Use of exponential representations 71
3.2 Perturbations and sensitivity analysis 73
3.2.1 Expansions of the expectation $\mathbb{E}_{\theta^*}[f]$ 73
3.2.2 Expansions for $\log \mathbb{E}_{\theta^*}[f]$ 76
3.3 Bounds on expectations 78
3.3.1 Relation to previous work 78
3.3.2 Basic bounds based on a single approximating point 79
3.3.3 Bounds based on multiple approximating distributions 81
3.4 Extension to the basic bounds 85
3.4.1 Tighter single point bounds 86
3.4.2 Tighter multiple point bounds 87
3.5 Results on bounding the log partition function 87
3.5.1 Unoptimized bounds 88
3.5.2 Bounds with optimal mean field vector 91
3.6 Discussion 92

4 Embedded trees algorithm for Gaussian processes **93**
4.1 Introduction 93
4.2 Estimation of Gaussian processes 94
4.2.1 Prior model and observations 94
4.2.2 Linear-Gaussian estimation 95
4.2.3 Gauss-Markov processes and sparse inverse covariance 95
4.2.4 Estimation techniques 96
4.3 Embedded trees algorithm 97
4.3.1 Embedded trees and matrix splitting 97
4.3.2 Recursions for computing the conditional mean 98
4.3.3 Convergence analysis 100
4.3.4 Calculation of error covariances 103
4.3.5 Results 104
4.4 Discussion 105

5 Tree-based reparameterization for approximate estimation **107**
5.1 Introduction 107

5.2 Estimation in graphical models 110
5.2.1 Exact estimation on trees as reparameterization 111
5.2.2 Belief propagation for graphs with cycles 112
5.3 Tree-based reparameterization framework 113
5.3.1 Exponential families of distributions 115
5.3.2 Basic operators 116
5.3.3 Tree-based reparameterization updates 117
5.3.4 Belief propagation as reparameterization 119
5.3.5 Empirical comparisons of BP versus TRP 121
5.4 Analysis of fixed points and convergence 124
5.4.1 Approximation to the Kullback-Leibler divergence 125
5.4.2 Tree reparameterization as a successive projection technique . . . 126
5.4.3 Sufficient conditions for convergence for two spanning trees . . . 129
5.4.4 Geometry and invariance of TRP updates 130
5.4.5 Implications for continuous processes 133
5.4.6 When does TRP/BP yield exact marginals? 134
5.5 Analysis of the approximation error 139
5.5.1 Exact expression 140
5.5.2 Error bounds 141
5.5.3 Illustrative examples of bounds 143
5.6 Discussion 146

6 Exploiting higher-order structure for approximate estimation 149
6.1 Introduction 149
6.1.1 Variational formulation 150
6.1.2 Related work 151
6.1.3 Overview of the chapter 151
6.2 Elements of the approximations 152
6.2.1 Basic definitions and notation 152
6.2.2 Core structures and distributions 153
6.2.3 Residual partition 155
6.2.4 Auxiliary distributions 156
6.2.5 Marginalization operators 158
6.3 Approximations to the Kullback-Leibler divergence 159
6.3.1 Disjoint and non-disjoint residual partitions 160
6.3.2 Approximation for a disjoint partition 161
6.3.3 Approximation for a non-disjoint partition 163
6.3.4 Properties of the approximation 165
6.3.5 Illustrative examples 167
6.4 Properties of optimal points 172
6.4.1 Existence of local minima 172
6.4.2 Invariance of optimal points 173

6.4.3 Generalized message-passing for minimization 175
6.4.4 Largest globally consistent substructure 179
6.5 Characterization of the error 184
6.5.1 Reformulation in exponential parameters 184
6.5.2 Exact error expression 185
6.5.3 Exact expression for larger substructures 186
6.5.4 Bounds on the error 186
6.6 Empirical simulations 187
6.6.1 When to use an approximation with more complex structure? . . 188
6.6.2 Choice of core structure 188
6.7 Discussion 190

7 Upper bounds based on convex combinations **193**
7.1 Introduction 193
7.1.1 Set-up 195
7.1.2 Basic form of bounds 196
7.2 Dual formulation with fixed $\bar{\mu}$ 198
7.2.1 Explicit form of dual function 199
7.2.2 Characterization of optimal points 201
7.2.3 Decomposition of entropy terms 203
7.2.4 Spanning tree polytope 204
7.3 Jointly optimal upper bounds 207
7.3.1 Optimal upper bounds on $\Phi(\theta^*)$ 207
7.3.2 Alternative proof 209
7.3.3 Characterization of joint optima 211
7.3.4 Relation to Bethe free energy 214
7.4 Algorithms and simulation results 214
7.4.1 Inner minimization over λ 214
7.4.2 Outer maximization over $\boldsymbol{\mu}_e$ 215
7.4.3 Empirical simulations 217
7.5 Discussion 220

8 Contributions and Suggestions **223**
8.1 High-level view 223
8.2 Suggestions for future research 225
8.2.1 Exact inference for Gaussian processes 225
8.2.2 Approximate inference for discrete processes 226
8.2.3 Bounds 229
8.3 Possible implications for related fields 230
8.3.1 Network information theory 230
8.3.2 Analysis of iterative decoding 231
8.3.3 Application to large deviations analysis 232

A Algorithms for optimal estimation on trees **235**
A.1 Partial ordering in scale 235
A.2 Basic notation 235
A.3 Markov decomposition 236
A.4 Upward sweep 237
A.5 Downward sweep 238

B Proofs for Chapter 3 **241**
B.1 Proof of Proposition 3.3.1 241
B.2 Proof of Proposition 3.3.2 241
B.3 Proof of Proposition 3.4.1 242
B.4 Proof of Proposition 3.4.2 242

C Proofs for Chapter 5 **245**
C.1 Proof of Proposition 5.3.1 245
C.2 Proof of Proposition 5.4.1 246
C.3 Proof of Theorem 5.4.1 248
C.4 Proof of Theorem 5.4.2 249
C.5 Proof of Proposition 5.4.2 253

D Proofs for Chapter 7 **255**
D.1 Proof of Proposition 7.2.2 255
D.2 Proof of Proposition 7.3.1 256

Bibliography **258**

List of Figures

2.1 Node and edge induced subgraphs. 38
2.2 Forests and spanning trees 40
2.3 Graph cliques of size one through four 40
2.4 Illustration of triangulation 40
2.5 Cut vertices and bridges 41
2.6 Graph separation and conditional independence 42
2.7 Necessity of running intersection for probabilistic consistency 45
2.8 Junction tree for 3×3 grid 47
2.9 Differential manifold of log distributions $\log p(\mathbf{x}; \theta)$ 54
2.10 Geometry of graph-structured distributions 58
2.11 KL divergence as a Bregman distance 59
2.12 Geometry of I-projection onto an e-flat manifold 64
2.13 Graphical consequence of mean field 67

3.1 Error in zeroth-order approximation on a single cycle 76
3.2 Convex combination of exponential parameters 83
3.3 Effect of refining the partition for unoptimized point 90
3.4 Effect of refining the partition for mean field solution 91

4.1 Gauss-Markov processes and sparse inverse covariance 96
4.2 Illustration of embedded spanning trees 98
4.3 Illustration of tree-cutting operation for Gaussian processes 99
4.4 Convergence rates for ET 105

5.1 Simple example of a graphical model 112
5.2 Illustration of spanning trees on grid 113
5.3 Graphical illustration of TRP updates 114
5.4 Message-free version of belief propagation 120
5.5 Convergence plots of TRP versus BP 123
5.6 Convergence percentages of TRP versus BP 124
5.7 Pythagorean condition of successive iterates 127
5.8 Tree-based consistency satisfied by fixed points 128

5.9 Geometry of tree-reparameterization updates 132
5.10 Degeneracy of compatibility functions for a 4-state process. 136
5.11 Analysis of exactness on single cycle 137
5.12 Nearly exact problem for TRP/BP 138
5.13 Behavior of TRP and bounds for different clique potentials 144
5.14 Dependence of bounds on spanning tree choice 145

6.1 Illustration of triangulated and non-triangulated graphs $\widetilde{\mathcal{G}}(\mathbf{A})$ 154
6.2 Core and residual structures for 3×3 grid. 156
6.3 Augmented subgraph and its triangulation for 3×3 grid 158
6.4 Non-disjointness of augmented residual sets 161
6.5 4-plaque Kikuchi approximation on a 3×3 grid 164
6.6 Exactness of $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ approximation for a 2-square graph 168
6.7 Kikuchi approximation on 2-square graph 169
6.8 Necessity of disjointness of augmented residual sets 170
6.9 Non-exactness of $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ for a simple 5-node graph 171
6.10 Core and augmented structures for BP 178
6.11 Flow of message-passing 180
6.12 Largest globally consistent substructure for BP 181
6.13 Largest globally consistent substructures for a Kikuchi approximation . 183
6.14 Error bounds for TRP/BP compared to a spanning tree approximation 189
6.15 Effect of tree choice in spanning tree approximation 190

7.1 Convex combination of distributions 197
7.2 Illustration of optimal $\widehat{\theta}(\mathcal{T}_i)$ on a single loop 203
7.3 Graph with a bridge or single-edge cutset 206
7.4 Non-critical edge subsets for a single loop 207
7.5 Illustration of optimal edge appearance probabilities on single loop . . . 209
7.6 Geometry of optimal $\widehat{\mu_e}$ in $\mathbb{T}(\mathcal{G})$ 213
7.7 Upper and lower bounds for grids of various sizes 218
7.8 Upper and lower bounds for complete graph K_9 220

A.1 Definitions of quantities on trees 236

List of Tables

5.1 Convergence behavior of TRP versus BP 122

Chapter 1

Introduction

A fundamental problem in applied probability theory is that of constructing, representing and manipulating a *global* probability distribution that is based on relatively *local* constraints. This issue arises in a tremendous variety of fields. For example, in statistical image processing or computer vision [e.g., 65,73,126,171], one relevant set of random variables are the grey-scale values of the image pixels. Of course, since images are locally smooth, neighboring pixels are likely to share similar intensity values. This fact imposes a large set of local constraints on the grey-scale values. In order to form a model suitable for applications like image coding or denoising, it is necessary to combine these local constraints so as to form a global distribution on images. Similar issues arise in building models of natural language [e.g., 138] or speech signals [e.g., 144]. In channel coding [e.g., 71, 167], reliable transmission of a binary signal over a noisy channel requires a redundant representation or code. Linear codes can be defined by requiring that certain subsets of the bits have even parity (i.e., their sum is zero in modulo two arithmetic). Each of these parity-checks typically involves only a relatively small fraction of the transmitted bits. The problem of decoding or estimating the transmitted codeword, however, requires a global distribution on all possible codewords. Finally, in statistical mechanics [e.g., 135,165], the behavior of many physical phenomena (e.g., gases, crystals, magnets) is well-described by positing local interactions among a large set of quantities (e.g., particles or magnets) viewed as random variables. Of interest to the physicist, however, are global properties of the system as a whole (e.g., phase transitions, magnetization).

The development of methods to attack problems of this nature has varied from field to field. Statistical physicists, dating back to Boltzmann and Gibbs [e.g., 75], made the first inroads. For example, Ising [90] in 1925, seeking to qualitatively understand phase transitions in ferromagnetic materials, introduced the model that now bears his name. In coding theory, Gallager [69, 70] in the early 1960s proposed and analyzed low-density parity check codes. Although they received relatively little attention at the time, they have since become the subject of considerable research [e.g., 37, 124, 129, 148, 149]. Onwards from the 1970s, statisticians and probability theorists have studied the relations among Markov fields, contingency tables, and log-linear models [e.g., 21, 48, 49, 52, 76, 79, 122, 160]. Markov random field models and the Gibbs sampler were introduced to image processing in the late 1970s and early 1980s [e.g., 73, 84, 112, 177].

Pearl [137] spearheaded the use of probabilistic network models in artificial intelligence, and also studied the formal semantics of both directed and undirected networks.

Since this pioneering work, it has become clear that the approaches of these different fields — though ostensibly disparate — can be unified by the formalism of *graphical models*. Graphical models provide a powerful yet flexible framework for representing and manipulating probability distributions defined by local constraints [49, 67, 101, 104, 121]. Indeed, models in a wide variety of fields, including the Ising model of statistical physics [90], graphs associated with compound codes (e.g., turbo codes [18, 130], and low-density parity check codes [70, 124, 149]), and various models for image processing and computer vision [e.g., 21, 73, 112, 126, 171, 177] can all be viewed as particular cases of a graphical model.

At the core of any graphical model is a graph — that is, a collection of nodes joined by certain edges. Nodes in the graph represent random variables, whereas the edge structure encodes particular statistical relations among these variables. These models derive their power from fundamental correspondences between graph-theoretic ideas, and concepts in probability theory [104, 121]. A special case of such a correspondence will be known by any reader familiar with (discrete-time) Markov processes. The defining feature of such processes is that the random variables in the past and future are conditionally independent given the present state. In graphical terms, samples of the Markov process can be viewed as living at nodes of a linear chain. The graphical property corresponding to conditional independence is that removing any single node will break the chain into two components (past and future). For graphs with more structure than a chain, there exists a correspondingly more general set of Markov properties. The well-known Hammersley-Clifford theorem [38, 79] is a precise specification of the general correspondence between Markov properties and graph structure.

■ 1.1 Research areas related to graphical models

Graphical models, while providing a unifying framework, are by no means a panacea. Indeed, it could be argued that these models pose more problems than they solve. Undoubtedly, however, graphical models provide a convenient language with which to formulate precisely a number of problems common to many fields. In this section, we provide a high-level overview of a subset of these problems.

■ 1.1.1 Estimation or inference

In many applications, it is desirable to estimate or make inferences about a collection $\mathbf{x} = \{x_s\}$ of random variables, based on a set of noisy observations $\mathbf{y} = \{y_s\}$. A Bayesian approach to this problem entails combining any prior information about $\mathbf{x}$ with the new information introduced by the observations. In the context of this thesis, the prior information about $\mathbf{x}$ is represented by a distribution specified by a graphical model.

For example, in image processing, each y_s could correspond to a noise-corrupted ob-

servation of the grey-scale intensity x_s at image location s. Any statistical dependency among the grey-scale values $\{x_s\}$ — that is, the prior information — is specified by a particular graphical model. Denoising an image refers to the procedure of using the noisy observations $\mathbf{y}$ so as to infer the true grey-scale values $\mathbf{x}$. The resulting estimate can be thought of as a "denoised" image. A similar task arises in channel coding: here the elements of $\mathbf{y}$ correspond to the received bits, which may have been corrupted by transmission through the channel. We use these received bits to estimate the transmitted codeword $\mathbf{x}$, where the structure of the code (i.e., the set of permissible codewords) is represented by a graphical model.

■ 1.1.2 Model selection

A related problem is that of model fitting. Suppose that we are given a set of samples $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$, drawn independently from some unknown distribution. Presumably these samples provide some information about the structure of the underlying distribution. The problem of model selection, then, is to make use of these samples so as to infer or fit an appropriate model for the underlying distribution. Any procedure for model selection depends, of course, on the criterion of model fidelity that is specified.

As an example drawn from image processing, each $\mathbf{x}^{(i)}$ might correspond to a sample of a particular image texture (e.g., wood or grass). On the basis of these samples, we want to select a model that captures the statistical structure of the given texture.

■ 1.1.3 Sampling

Given a distribution defined by a graphical model, an important problem is how to draw random samples from this distribution. Although this sampling problem might appear straightforward at first blush, it is, in general, an exceedingly difficult problem for large-dimensional problems. The tutorial paper by MacKay [128] gives helpful insight into the nature of the difficulties; see also Ripley [150].

Returning to our image processing example, suppose that we have specified a model for a particular texture — for example, wood. The ability to draw samples would allow us to assess whether or not the model captures the visually salient features of wood. If indeed the model were realistic, then an efficient sampling procedure would allow us to synthesize patches of wood texture.

Of course, these research areas are all interconnected. Indeed, the process of model selection typically entails performing inference as a subroutine. Moreover, any procedure for drawing random samples from a distribution forms the basis for a Monte Carlo method [28,150] for performing (approximate) inference. The bulk of this thesis focuses on estimation and inference; due to these interconnections, however, our results have implications for other research areas as well.

1.2 Principal methods

In this section, we provide a broad overview of the principal methods used in this thesis. In particular, our analysis draws primarily from the following three bodies of theory:

- exponential families and information geometry
- convex analysis and duality
- variational theory and methods

The set of distributions defined by a graphical model can be formulated as an exponential family. These families and their associated geometry have been studied extensively in applied probability theory and statistics [e.g., 5,7,13,33,43,45,82]. Exponential families have a rich geometric structure, in which the Fisher information matrix plays the role of a Riemannian metric [145]. Indeed, an exponential family constitutes a differential manifold of distributions, for which the exponential variables constitute a particular parameterization. A distinguishing feature of manifolds formed by exponential families is the existence of a second set of parameters, which are coupled to the exponential variables. From this dual parameterization arises a considerable amount of additional geometric structure, in which the Kullback-Leibler divergence assumes a central role. This body of theory is known collectively as *information geometry*.

At a broad level, convex analysis [e.g., 59, 86, 151] is the study of convex sets and functions. Ideas and techniques from convex analysis play important roles in various fields, from statistical physics [135] to information theory [41]. Especially important is the notion of convex duality, of which there are various related forms (e.g., Fenchel, Legendre, Lagrangian). Convex duality not only provides conceptual and geometric insight, but also has important practical consequences for developing optimization algorithms.

Variational formulations, along with the associated body of theory and methods, are integral to many disciplines of science and engineering [e.g., 92, 153, 179]. At the heart of such methods is the idea of specifying a quantity of interest in a variational fashion — that is, as the minimizing (or maximizing) argument of an optimization problem. A variational formulation makes it possible to study or approximate the quantity of interest by studying or approximating the corresponding optimization problem.

As will become clear later in the thesis, there exist deep connections between these three areas. For example, exponential families arise most naturally as maximum entropy distributions [179] subject to linear constraints — that is, in a variational fashion. Moreover, the two sets of parameters for an exponential family are coupled by a particular form of convex duality, namely the Legendre transform [151]. Convex analysis is also intimately linked to many variational methods [see, e.g., 59].

1.3 Main problems and contributions

In this section, we discuss the problems that are addressed by this thesis, as well as the nature of our specific contributions. The main problems tackled in this thesis are the

following:

- inference on graphs with cycles:
  (a) exact inference for Gaussian processes
  (b) approximate inference for discrete processes
- computable bounds on probabilistic quantities (e.g., the so-called partition function;[1] marginal distributions)

Before proceeding to an in-depth discussion of these problems, we pause to discuss the unifying theme of this thesis. An important subclass of graphs are those without cycles, which are known as trees. One fact highlighted by our work is that graphs with cycles are fundamentally different than trees. As we will see, for trees, all three of the problems described in Section 1.1 are relatively well-understood, and can be solved by very fast algorithms. In contrast, these same problems are intractable for general graphs with cycles.

At a very high level, all the work described in this thesis is based on the following simple observation: *embedded within any graph with cycles are a large number of trees.* Given a problem on a graph with cycles, it is tempting, therefore, to consider modified problems defined on trees. As demonstrated by our results, the remarkable fact is that studying this simpler set of modified tree problems can lead to considerable insight about the original problem on the graph with cycles. Although the work described here focuses primarily on embedded trees, it should be clear that similar ideas can be applied to triangulated subgraphs with more complex structure than trees (e.g., graphs of higher treewidth[2]) embedded within the original graph.

We now turn to discussion of the main problems addressed in this thesis.

■ 1.3.1 Inference in graphs with cycles

As noted above, a fundamental fact is that the complexity of inference depends very strongly on graph structure. A simple case, one which may be familiar to many readers, should help to illuminate the role of graph structure in inference. Suppose that we wish to estimate a discrete-time Markov process $\mathbf{x} = \{x_t \mid t = 0, \ldots, N-1\}$, based on an associated set of noisy observations $\mathbf{y} = \{y_t\}$ where each y_t is a measurement of the corresponding x_t. For this Markov chain problem, there exist well-known and very efficient algorithms for carrying out standard estimation tasks [e.g., 100, 109, 144, 146]. For example, in one version of the so-called *smoothing problem*, we want to compute, for each time $t = 0, \ldots, N-1$, the marginal distribution of x_t conditioned on the full set $\mathbf{y}$ of observations. Any efficient algorithm for this task has a recursive form, typically

[1] As we will see in the sequel, the partition function plays an important role in graphical models.

[2] An ordinary tree is a graph of treewidth one; roughly speaking, graphs of higher treewidth correspond to trees on clusters of nodes from the original graph. See [17, 162] for further discussion of hypergraphs and treewidth.

involving a forward and backward sweep. For example, in the Gauss-Markov case, the forward sweep corresponds to the Kalman filter [100, 109, 110], whereas one version of the backward sweep corresponds to the Rauch-Tung-Striebel smoother [146]. Going through the derivation reveals that Markov properties of the chain — namely, that past and future are conditionally independent given the present — are exploited heavily.

Interestingly, recursive algorithms for exact estimation, rather than being limited to chain-structured graphs, are more generally applicable to the class of acyclic graphs or trees. (Note that a simple chain is a special case of a tree). An important fact is that the nodes of any tree-structured graph can be put into a partial order by arbitrarily designating one node as the root, and then measuring the scale of other nodes in terms of their distance from the root. This partial ordering, in conjunction with Markov properties of a tree, permit the derivation of efficient recursive techniques for exact estimation on a tree [e.g., 35,137]. The most efficient implementation of such algorithms again have a two-pass form, in which the computation first sweeps from outer nodes towards the root node, and then from the root node outwards.

Graphs with cycles, on the other hand, are fundamentally different than acyclic graphs. In the presence of cycles, nodes cannot be partially ordered, so that it is no longer possible to exploit Markov properties of the graph to derive recursive algorithms. As we will discuss in Chapter 2, although there exist general-purpose algorithms for exact inference on graphs with cycles, they are all based on suitably modifying the graph so as to form a tree. Moreover, the complexity of these exact methods, in general, scales poorly with problem size.

It is therefore of considerable interest to develop efficient algorithms for exact or approximate inference on graphs with cycles. Although a great deal of work has been devoted to this area, there remain a variety of open problems. In the following sections, we discuss the open problems addressed in this thesis, first for Gaussian and then for discrete-valued processes.

■ 1.3.2 Exact inference for Gaussian processes

In the Gaussian case, exact inference refers to the computation of both the conditional means and error covariances at each node of the graph. The complexity of the brute force approach to this computation — namely, matrix inversion — scales cubically as a function of the number of nodes N. In many applications [e.g., 62, 126], the number of nodes may be on the order of 10^5 or 10^6, so that an $\mathcal{O}(N^3)$ cost is unacceptable.

Tree-structured Gaussian processes are especially attractive due to the tractability of inference. In particular, the computational complexity of a two-pass algorithm for exact inference on a tree is $\mathcal{O}(N)$ (see Chou et al. [35]). In order to leverage these fast algorithms for problems in signal or image processing, one strategy is to use a multiscale tree in order to model dependencies among a collection of random variables, representing a time series or 2-D random field, in an approximate fashion. The variables to be modeled are viewed as lying at the finest scale of the tree. In the context of image processing, these fine scale variables might correspond to grey-scale intensity

values at each pixel, whereas coarser scale variables might correspond to aggregate quantities (e.g., wavelet coefficients). Instead of modeling dependencies among the fine scale variables directly, the approach is to build a tree model on top of them, in which variables at higher levels of the tree capture dependencies among subsets of the fine scale variables. This general modeling philosophy, in conjunction with efficient techniques for stochastic realization of these multiscale tree models [e.g., 63, 88, 89], have been applied successfully to various problems [e.g., 47, 62, 87, 126].

It turns out that these tree-structured models tend to capture long-range dependencies well, but may not be as effective at modeling short-range interactions. To understand the source of this problem, consider again the example of image processing, in which fine scale variables correspond to grey-scale intensity values. Of course, intensity values at spatially adjacent pixels tend to be highly dependent. However, certain pairs of such pixels are mapped to pairs of tree nodes that are separated by a very large tree distance. A tree model will fail to capture the dependency between such a pair of variables, a deficiency which manifests itself with abrupt jumps (or boundary artifacts) in samples drawn from the approximate tree model [see, e.g., 89, 126].

A number of ad hoc methods [e.g., 88] have been proposed to deal with boundary artifacts, but none are entirely satisfactory. Indeed, the most natural solution is to add extra edges to the tree as necessary. With the addition of these edges, however, the new graph is not a tree (it has cycles!); as a consequence, efficient inference algorithms for trees [35] are no longer applicable. This fact necessitates the development of efficient algorithms for exact estimation of Gaussian processes on graphs with cycles.

There are a variety of methods for efficiently computing the conditional means of a Gaussian problem on a graph with cycles. The options include techniques from numerical linear algebra [54], as well as the so-called belief propagation algorithm [137], which will be discussed at more length in the following section. However, none of these methods compute the (correct) error covariances. This is a serious deficiency, since in many applications [e.g., 62, 126], these error statistics are as important as the means themselves.

In Chapter 4, we develop a new iterative algorithm for exact estimation of Gaussian processes on graphs with cycles. As a central engine, it exploits the existence of efficient algorithms [35] for solving any Gaussian estimation problem defined on a tree embedded within the original graph. For this reason, we call it the *embedded trees* (ET) algorithm. At each iteration, the next iterate is generated by solving an appropriately modified Gaussian estimation problem on a spanning tree of the graph. We will prove that if the sequence of tree problems is suitably constructed, then the sequence of iterates converges geometrically to the true means and error covariances on the graph with cycles.

■ 1.3.3 Approximate inference for discrete-valued processes

For discrete-valued Markov processes on graphs, one important inference problem is to compute marginal distributions at each node of the graph. It can be shown [39] that

this problem is NP-hard. As a result, techniques for approximate inference are the focus of a great deal of current research.

The *belief propagation* algorithm [137], also known as the *sum-product algorithm* in coding theory [e.g., 117, 130], is a well-known and widely studied method [e.g., 3, 130, 147, 173, 180] for approximate inference. This algorithm is used in a wide variety of fields, ranging from artificial intelligence and computer vision [e.g., 65, 68, 133] to coding theory, where it shows up as a highly successful iterative decoding method for turbo codes [18, 130] and low-density parity check codes [71, 124, 129, 149]. As a result, belief propagation has generated tremendous excitement in a number of communities.

Belief propagation (BP) is a technique for computing approximate marginal distributions at each node of the graph. It is an iterative algorithm, in which so-called messages are passed from node to node along edges of the graph. On a tree-structured graph, it is guaranteed to yield the correct marginals in a finite number of iterations. On a graph with cycles, in contrast, the algorithm may not converge, and even when it does, the resulting approximations are of variable accuracy. Accordingly, the behavior of BP in application to graphs with cycles has been the focus of a great deal of recent research [e.g., 2, 8, 147, 173, 180]. We provide a brief review of this work in Section 5.1 of Chapter 5. For now we highlight the recent results of Yedidia et al. [180], who provided a variational interpretation of BP. In particular, their analysis established that points to which BP can converge (i.e., fixed points) correspond to extremal points of the so-called Bethe free energy from statistical physics. Nonetheless, despite the advances of recent work, there remain a number of open questions associated with belief propagation, perhaps the most important of which being the nature of the approximation error.

This area is the focus of Chapter 5, in which we advocate a conceptual shift away from the traditional message-passing view of approximate inference (as in standard BP). In lieu, we develop the notion of *reparameterization*. Any graphical model is specified by a product of so-called compatibility functions defined over cliques of the graph; however, this representation is not necessarily unique. A reparameterization operation, then, corresponds to choosing a different set of compatibility functions for the factored representation of the distribution. On one hand, we will show that BP updates can be re-formulated as a very local form of such reparameterization; on the other hand, we will consider more general forms of reparameterization that entail performing exact computations on spanning trees of the graph. These more global operations will be called tree-based reparameterization (TRP) updates.

The perspective of TRP gives rise, in a very natural way, to a number of new theoretical insights. First of all, we give an intuitive characterization of fixed points: they must be consistent, in a suitable way to be defined, with respect to *every* acyclic substructure embedded within the original graph.[3] Secondly, we establish a fundamental property of TRP or BP updates: when viewed as a sequence of reparameterizations, they leave the original distribution on the graph with cycles unchanged. This invari-

[3]Spanning trees are maximal acyclic subgraphs.

ance has a number of consequences, of which the most important is the resulting insight into the approximation error — i.e., the difference between the TRP/BP approximate marginals, and the actual marginals. Results pertaining to this error have been obtained in certain special cases: single loops [173], and the graphs corresponding to turbo codes [147]. The TRP perspective allows us to give an exact expression of the approximation error for an arbitrary graph. This exact expression is the starting point for deriving error bounds. Interestingly, although our insights emerge in a natural way from the TRP perspective, most of them apply in an algorithm-independent manner to any constrained local minimum of the Bethe free energy, regardless of how it is obtained.

It is well-known that belief propagation tends to give poor results on certain kinds of graphs (e.g., those with many short cycles). It is therefore desirable to develop principled methods for improving the BP approximation. In Chapter 6, we present a framework for developing and analyzing such extensions. The basis of this framework is a decomposition of the graph with cycles into a core structure, over which exact computations can be performed, and a set of residual elements (e.g., edges and/or cliques) not captured by the core. We show that the notion of reparameterization, as developed in Chapter 5, extends in a natural way to all approximations in this class. As a consequence, most of our results on TRP have corresponding generalizations. We establish that fixed points are characterized by consistency conditions over certain embedded substructures. For example, in the case of Kikuchi approximations, we find that clique trees[4] embedded within the original graph play the same role that spanning trees do for the Bethe free energy of belief propagation. Moreover, we prove that the original distribution remains invariant under the reparameterization updates, and we also analyze the approximation error. An ancillary contribution of Chapter 6 is to unify two previously proposed extensions: the Kikuchi approximations of Yedidia et al. [180], and the expectation-propagation technique of Minka [131].

■ 1.3.4 Upper and lower bounds

It is often desirable to obtain upper and lower bounds on various quantities associated with a probability distribution, including marginal probabilities at particular nodes (or subsets of nodes), as well as the partition function. In the context of estimation, a set of upper and lower bounds on a particular marginal provides much stronger information than a mere approximation — namely, the guarantee that the desired marginal probability *must* lie within the specified window. Bounds on the partition function are important for a variety of problems, including model selection [105] and large deviations analysis [158]. Given a set of data points, the partition function has the interpretation as the likelihood of observing that particular set of data under the given model. Selecting a model according to the principle of maximum likelihood [113, 118], then, corresponds to choosing model parameters so as to maximize the partition function. The theory of large deviations [e.g., 53, 158] deals with the exponential rate at which the probability

[4] A clique tree for a graph is an acyclic graph in which the nodes consist of certain clusters of nodes (i.e., cliques) from the original graph.

of observing an unlikely event (a so-called large deviation: e.g., 900 or more heads in 1000 tosses of a fair coin) decays asymptotically as the number of samples tends to infinity. In this context, the (log) partition function is well-known to play the role of a rate function — that is, it specifies these exponential error rates.

Mean field theory [e.g., 105], as described in Section 2.3, provides a well-known lower bound on the partition function. This lower bound, in conjunction with the EM algorithm [55], forms the basis of an important method for approximate model fitting [105]. Strengthened versions of the mean field lower bound, derived by including higher order terms in a Taylor series, have been proposed by [123]. In comparison, upper bounds appear to be much more difficult to derive. For the case of binary-valued nodes with pairwise interactions, Jaakkola and Jordan [94] exploited ideas from convex analysis to derive a recursive node-elimination procedure for upper bounding the partition function.

In Chapter 3, we derive both lower and upper bounds on the expectation of an arbitrary function (say f). These lower bounds are closely related to standard mean field, in that they follow from exploiting the convexity of the log partition function — in our case, a partition function modified in a way dependent on f. We then derive a new set of upper bounds that are based on taking convex combinations of exponential parameters. We also develop a technique for strengthening an arbitrary bound, based on the idea of decomposing the function f in an additive manner. We prove that for both the lower and upper bounds developed in Chapter 3, this technique is guaranteed to yield (in general) strictly tighter bounds. The bounds developed in Chapter 3 play a fundamental role in our analysis of the error in approximate inference techniques, as described in Chapters 5 and 6.

The new class of upper bounds based on convex combinations are studied more extensively in Chapter 7. We consider, in particular, the set of convex combinations formed from all spanning trees embedded within a graph with cycles. A crucial fact here is that the number of such spanning trees is typically extremely large. (E.g., the complete graph K_N has N^{N-2} spanning trees [168].) Despite the apparent intractability of optimizing over such a huge number of trees, we show that exploiting ideas from Lagrangian duality leads to a drastic reduction in problem complexity. This simplification enables us to develop an efficient method for optimizing both the choice of exponential parameters as well as the choice of convex combination over all spanning trees. Moreover, this dual formulation of the problem gives a new and interesting perspective on the Bethe free energy.[5] In particular, our analysis leads to functions which, though closely related to the Bethe free energy, have the following attractive properties. First of all, they are strictly convex, so we are guaranteed a unique global minimum that can be found by standard methods from nonlinear programming [20]. Secondly, this global minimum yields an upper bound on the log partition function.

[5] As discussed in Section 1.3.3, the Bethe free energy plays an important role in the belief propagation algorithm for approximate estimation on graphs with cycles.

1.4 Thesis overview

In summary, the primary contributions of the thesis are as follows:

- a new iterative algorithm for exact estimation of Gaussian processes on graphs with cycles
- the tree-based reparameterization framework for approximate estimation of discrete-valued processes on graphs with cycles
- a unifying framework for the development and analysis of more advanced techniques for approximate inference
- a new class of upper bounds on the log partition function

The remainder of the thesis is organized, on a chapter by chapter basis, in the following manner:

Chapter 2: Background

This chapter sets out the background that underlies developments in the sequel. It begins with an overview of basic concepts in graph theory, followed by a self-contained but brief introduction to graphical models. We include a discussion of the junction tree representation of distributions [121,122], as well as the exact inference technique of the same name. We then introduce exponential families of distributions, and develop the associated theory of information geometry. The final section treats variational methods, with particular emphasis on mean field theory as an illustrative example.

Chapter 3: Perturbations and Bounds

This chapter illustrates the use of exponential representations in developing perturbation expansions and bounds on expectations of an arbitrary function (e.g., single-node marginal distributions). The perturbation expansions yield helpful information about the sensitivity of various quantities (e.g., marginal distributions) to changes in the model parameters. We then turn to the development of bounds on expectations of arbitrary functions. We show how to apply the lower bound from mean field theory to a tilted log partition function in order to obtain lower bounds on the expectation of an arbitrary function. We also derive a new class of upper bounds, based on the idea of taking convex combinations of exponential parameters. For the expectation of an arbitrary function, we develop a method for strengthening the bounds by performing an additive decomposition. We illustrate these bounds with some simple examples.

Chapter 4: Embedded trees algorithm for Gaussian processes

This chapter develops and analyzes the embedded trees (ET) algorithm for exact estimation of Gaussian processes defined on graphs with cycles. The ET algorithm generates a sequence of iterates (means and error covariances) by exactly solving a sequence of

modified problems defined on trees embedded within the graph. We prove that when the sequence of modified tree problems is appropriately chosen, the sequence of iterates converges to the correct mean and covariances for the original problem on the graph with cycles. The algorithm is illustrated in application to a problem on a nearest-neighbor grid. Theoretical extensions of this work as well as related empirical results can be found in [163].

Chapter 5: Tree-based reparameterization for approximate estimation

This chapter develops the tree-based reparameterization (TRP) framework for approximate inference on graphs with cycles. We show that belief propagation (BP) can be re-formulated as a special case of reparameterization, and establish that more global tree updates have superior convergence properties. We prove that fixed points of TRP updates satisfy the necessary conditions to be local minima of a cost function that is an approximation to the Kullback-Leibler divergence. Although this cost function is distinct from the Bethe free energy [180], the two functions coincide on the constraint set, which allows us to prove equivalence of TRP and BP fixed points. The TRP perspective leads to a new characterization of TRP/BP fixed points in terms of consistency over embedded acyclic subgraphs. We also establish that TRP and BP updates leave invariant the distribution on the graph with cycles, a result which has a number of important consequences. Finally, we use the fixed point characterization and invariance to analyze the approximation error. We first develop an exact expression for the error in an arbitrary graph with cycles. This expression, though conceptually interesting, is not tractable to compute in general. This difficulty motivates us to develop computable upper and lower bounds on the approximation error using the results from Chapters 3 and 7. We illustrate these bounds with some simple empirical examples.

Chapter 6: Exploiting higher-order structure for approximate estimation

This chapter provides a unified framework for developing and analyzing more advanced techniques for computing approximations to the marginals of a target distribution. Each approximation in this framework is specified by a cost function that depends on a set of so-called pseudomarginals. These pseudomarginals implicitly define a distribution, and the associated cost function constitutes an approximation to the Kullback-Leibler divergence between this distribution and the target distribution. We construct these approximations by decomposing the graph with cycles into a core structure, over which the pseudomarginals are updated by exact computations, and a set of residual terms (e.g., edges or cliques) not covered by the core structure. We demonstrate that various known approximations, including the Bethe free energy, Kikuchi approximations [180], and the proposal of Minka [131], are special cases of this framework. Moreover, we develop algorithms, analogous to the tree-based reparameterization updates of Chapter 5, for performing constrained minimization of the cost functions. The minimizing arguments constitute approximations to the actual marginals of the target distribution. Significantly, most of the theoretical results from Chapter 5 have natural generalizations

to all of the approximations in this framework. In particular, the ideas of reparameterization and invariance are generally applicable. We use these principles to characterize fixed points, and to analyze the approximation error.

Chapter 7: Upper bounds based on convex combinations

This chapter presents a new class of computable upper bounds on the log partition function that are applicable to an arbitrary undirected graphical model. The bounds are formed by taking a convex combination of tree-structured exponential parameters. The weight on each tree can be viewed as its probability under a distribution over all spanning trees of the graph. We consider the problem of optimizing these bounds with respect to both the tree-structured exponential parameters as well as the distribution over spanning trees. We show that a Lagrangian dual reformulation of the problem leads to substantial simplification. As a result, despite the extremely large number of spanning trees embedded in a general graph, we are able to develop an efficient algorithm for implicitly optimizing the bounds over *all* spanning trees. This dual reformulation also gives a new perspective on the Bethe free energy of approximate estimation. We illustrate the use of these bounds in application to random choices of distributions on various graphs. The methods developed in this chapter are broadly applicable. For instance, there are natural extensions to convex combinations of distributions structured according to clique trees, which in turn lead to a new perspective on Kikuchi free energies.

Chapter 8: Contributions and Suggestions

This chapter summarizes the contributions of the thesis, and points out a number of directions for future research. We also consider briefly the possible implications of the perspective and results of this thesis for related research areas, including network information theory, iterative decoding, and large deviations analysis.

Chapter 2

Background

This chapter outlines the background necessary for subsequent developments in this thesis. Graphical models provide a flexible framework for specifying globally consistent probability models based on local constraints. The primary focus of this thesis is problems that arise in using such models. We begin in Section 2.1 with an introduction to the basics of graphical models, and the relevant problems of inference and estimation. As a prelude to introducing graphical models, this section also contains a brief primer on graph theory. Section 2.2 introduces a particular representation of distributions defined by graphical models — namely, the exponential family. Associated with such families are a variety of elegant results, known collectively as information geometry. A third concept central to this thesis is that of a variational formulation. Accordingly, we devote Section 2.3 to an overview of variational methods, with a particular emphasis on mean field theory.

■ 2.1 Graphical models

Graphical models are a powerful framework for representing and manipulating probability distributions over sets of random variables. Indeed, stochastic processes defined on graphs arise in a variety of fields, including coding theory [71], statistical physics [15,31,135], artificial intelligence [137], computer vision [65], system theory [14] and statistical image processing [126]. The power of graphical models derives from the correspondence that they establish between the probabilistic concept of *conditional independence*, and the graph-theoretic notion of *node separation*.

We begin in Section 2.1.1 with a brief but self-contained introduction to the basics of graph theory. There are many books available to the reader interested in more background on graph theory [e.g., 16,22,25,26]. In Section 2.1.2, we turn to the basics of graphical models. More background on graphical models can be found in the books [67, 101,104,121]; another helpful source is the edited collection of papers [103]. Section 2.1.3 introduces the problem of estimation or inference in graphical models, which is central to many parts of this thesis. Section 2.1.4 briefly discusses exact inference algorithms for tree-structured graphs; more details can be found in Appendix A. In Section 2.1.5, we describe the notion of a junction tree, which is important both in a purely graph-theoretic context and for the purposes of inference in graphical models.

■ 2.1.1 Basics of graph theory

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of nodes or vertices $\mathcal{V} = \{1, \ldots, N\}$ that are joined by a set of edges $\mathcal{E}$. Edges in a graph can either be directed or undirected; this thesis will focus exclusively on undirected graphs. For an undirected graph, the notation (s, t) (or equivalently, (t, s)) denotes an undirected edge between nodes s and t in the vertex set. For any $s \in \mathcal{V}$, the set of *neighbors* of s in $\mathcal{G}$ is given by

$$\mathcal{N}(s) \triangleq \{ t \in \mathcal{V} \mid (s, t) \in \mathcal{E} \} \tag{2.1}$$

The *degree* of a node s, denoted $d(s)$, corresponds to the number of neighbors (i.e., the cardinality $|\mathcal{N}(s)|$ of the neighbor set).

A subgraph $\mathcal{H}$ of a graph $\mathcal{G}$ is formed by a particular subset of the vertices and edges of $\mathcal{G}$. It is often convenient to consider subgraphs induced by particular subsets of the vertex set, or by particular subsets of the edge set. First of all, given a subset S of the vertex set $\mathcal{V}$, the subgraph induced by S is given by $\mathcal{G}[S] = (S, \mathcal{E}[S])$ where $\mathcal{E}[S] = \{ (s, t) \in \mathcal{E} \mid s, t, \in S \}$. The graph $\mathcal{G}[S]$ is called a *node-induced subgraph.*

Similarly, given a subset $F \subset \mathcal{E}$ of the edge set, the subgraph induced by F is given by $\mathcal{G}(F) = (\mathcal{V}(F), F)$, where

$$\mathcal{V}(F) \triangleq \{ u \in \mathcal{V} \mid \exists\, v \in \mathcal{V} \text{ s.t } (u, v) \in F \}$$

This graph $\mathcal{G}(F)$ is called an *edge-induced* subgraph.

Examples of node and edge-induced subgraphs are given in Figure 2.1.

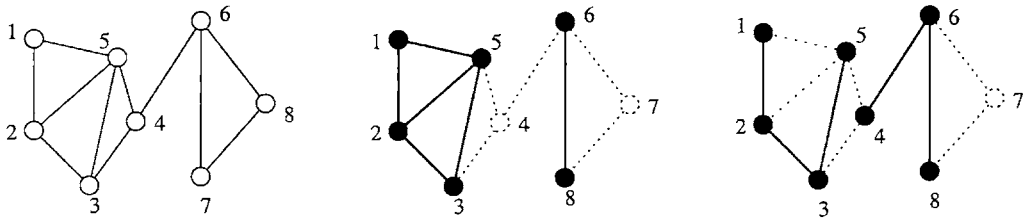


Figure 2.1. Illustration of node and edge-induced subgraphs. Vertices and edges in the subgraph are shown in dark circles and solid lines (respectively), while those not in the subgraph are shown with dotted open circles and dotted lines (respectively). (a) Graph $\mathcal{G}$ with cycles. (b) The node-induced subgraph $\mathcal{G}[S]$ for $S = \{1, 2, 3, 5, 6, 8\}$. (c) The edge-induced subgraph $\mathcal{G}(\mathcal{H})$ with $F = \{(1, 2), (2, 3), (3, 5), (4, 6), (6, 8)\}$.

A *path* is a graph $\mathcal{P}$ consisting of the vertex set $\mathcal{V}(\mathcal{P}) = \{s_0, s_1, \ldots, s_k\}$ and edge set $\mathcal{E}(\mathcal{P}) = \{(s_0, s_1), \ldots (s_{k-1}, s_k)\}$. The vertices s_0 and s_k are the end vertices of the path, and $l(\mathcal{P}) = k$ is the length of the path. We say that $\mathcal{P}$ is a path from s_0 to s_k. A *cycle* is a path from a node s back to itself formed of a sequence of distinct edges. I.e.,

a cycle consists of a sequence of distinct edges $\{ (s_1, s_2), (s_2, s_3), \ldots, (s_{k-1}, s_k) \}$ such that $s_1 = s_k$.

We say that a graph is *connected* if for each pair $\{s, t\}$ of distinct vertices, there is a path from s to t. A *component* of the graph is a maximal connected subgraph. The notation $c(\mathcal{G})$ denotes the number of (connected) components in the graph $\mathcal{G}$.

An important subclass of graph is those without cycles:

Definition 2.1.1 (Forests and trees). A *tree* $\mathcal{T}$ is a cycle-free graph consisting of a single connected component. A *forest* is formed by the union of a collection of trees. Given a graph $\mathcal{G}$, a *spanning tree* is an embedded tree (i.e., a tree-structured subgraph of $\mathcal{G}$) that reaches each vertex. See Figure 2.2. for illustration of these concepts.

Definition 2.1.2 (Cliques). A *clique* of a graph $\mathcal{G}$ is any fully connected subset of the vertex set $\mathcal{V}$. A clique is *maximal* if it is not properly contained within any other clique.

Figure 2.3 illustrates the structure of cliques of sizes one through four. Note that any single node is itself a clique, but not a maximal clique unless it has no neighbors. If we return to Figure 2.1(a), nodes $\{1, 2, 5\}$ form a 3-clique, but nodes $\{1, 2, 5, 3\}$ do *not* four a 4-clique, since node 1 is not connected (directly) to node 3.

Let $\mathbf{C} = \mathbf{C}(\mathcal{G})$ denote the set of all cliques in a graph $\mathcal{G}$. For instance, given a tree $\mathcal{T}$, the clique set $\mathbf{C}(\mathcal{T})$ consists of the union $\mathcal{V} \cup \mathcal{E}$ of the vertex set with the edge set. We use $\mathcal{C}$ to denote an arbitrary member of $\mathbf{C}$ (i.e., a particular clique of $\mathcal{G}$).

Given a subset of the clique set $\mathbf{C}$, it is natural to define the following generalization of an edge-induced subgraph:

Definition 2.1.3 (Clique-induced subgraphs). Given a subset $\mathbf{B} \subset \mathbf{C}$ of the clique set, let $\mathcal{G}(\mathbf{B})$ denote the subgraph of $\mathcal{G}$ induced by the cliques in $\mathbf{B}$. More precisely, $\mathcal{G}(\mathbf{B}) = (\mathcal{V}(\mathbf{B}); \mathcal{E}(\mathbf{B}))$ where

$$\mathcal{V}(\mathbf{B}) \triangleq \{ s \in \mathcal{V} \mid s \in \mathcal{C} \quad \text{for some } \mathcal{C} \in \mathbf{B} \} \tag{2.2a}$$

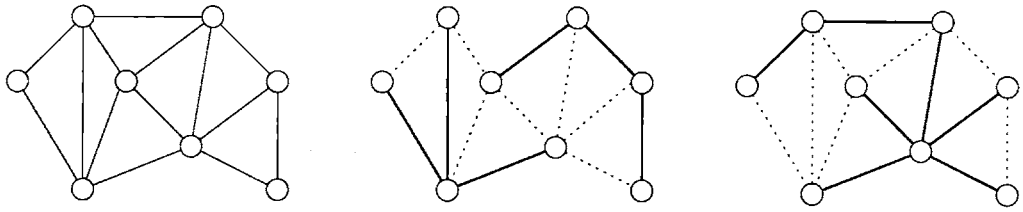
$$\mathcal{E}(\mathbf{B}) \triangleq \{ (s,t) \in \mathcal{E} \mid s, t \in \mathcal{C} \quad \text{for some } \mathcal{C} \in \mathbf{B} \} \tag{2.2b}$$

Note the clique set of $\mathcal{G}(\mathbf{B})$ can be strictly larger than $\mathbf{B}$. For example, if we consider a single loop $\mathcal{G}$ on three nodes with $\mathbf{B} = \{(1,2), (2,3), (1,3)\}$, then $\mathcal{G}(\mathbf{B}) = \mathcal{G}$, so that the clique set of $\mathcal{G}(\mathbf{B})$ includes the 3-clique $\{1, 2, 3\} \notin \mathbf{B}$.

An important subclass of graphs are those satisfying the following property:

Definition 2.1.4 (Triangulated). A graph $\mathcal{G}$ is *triangulated* if every cycle of length 4 or greater has a chord (i.e., an edge joining two vertices not adjacent in the cycle). See Figure 2.4 for illustrations of triangulated versus non-triangulated graphs.

This notion of triangulation will play a central role in the junction tree representation of graphical models, to be discussed in Section 2.1.5. Given a graph that is not triangulated, it is always possible to form a triangulated version $\widetilde{\mathcal{G}}$ by adding chords to



(a) (b) (c)

Figure 2.2. (a) Graph $\mathcal{G}$ with cycles. (b) A forest embedded within $\mathcal{G}$. (c) Embedded spanning tree that reaches each vertex of $\mathcal{G}$.

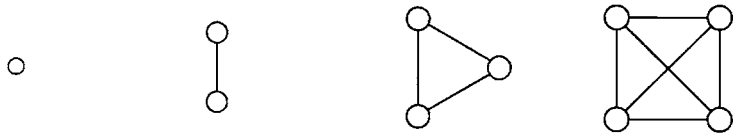
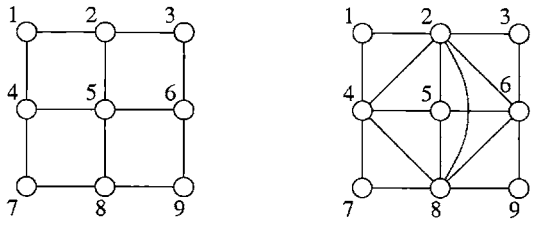


Figure 2.3: Graph cliques of size 1 through 4.



(a) (b)

Figure 2.4. Illustration of a non-triangulated versus triangulated graph. (a) This 3×3 grid is not triangulated; it has many four cycles (e.g., the cycle formed by nodes $1-2-5-4-1$) that lack a chord. (b) Here is one triangulated version of the 3×3 grid, formed by adding the extra edges $\{(2,4),(4,8),(2,6),(6,8),(2,8)\}$. The extra edge $(2,8)$ is added as a chord for the 4-cycle formed by nodes $2-4-8-6-2$.

cycles as necessary. However, this triangulated version need not be unique; that is, a given untriangulated graph $\mathcal{G}$ may have a number of possible triangulations.

It is useful to distinguish vertices (and edges), that if removed from the graph, increase the number of connected components:

Definition 2.1.5 (Cut vertices and bridges). A vertex is a *cut vertex* if its deletion from the graph increases the number of connected components. A *bridge* is an edge whose deletion increases the number of connected components. (See Figure 2.5).

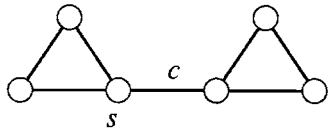


Figure 2.5: Vertex s is a cut vertex in the graph shown, whereas edge c is a bridge.

■ 2.1.2 Basics of graphical models

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a probabilistic graphical model is formed by associating with each node $s \in \mathcal{V}$ a random variable x_s taking values in the sample space $\mathcal{X}$. This sample space can be either a continuum (e.g., $\mathcal{X} = \mathbb{R}$), or the discrete alphabet $\mathcal{X} = \{0, \ldots, m-1\}$. In this latter discrete case, the underlying sample space $\mathcal{X}^N$ is the set of all N vectors $\mathbf{x} = \{x_s \mid s \in \mathcal{V}\}$ over m symbols, so that $|\mathcal{X}^N| = m^N$.

In a graphical model, the edges of the underlying graph represent probabilistic dependencies between variables, and come in two varieties — directed or undirected. Although the probabilistic interpretation of directed and undirected edges is different, any directed graph can be converted to an equivalent[1] undirected graph [see, e.g., 137]. In this thesis, we restrict our attention to undirected graphs.

The stochastic processes of interest are those which are Markov with respect to the underlying graph. To define this concept, let $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ be subsets of the vertex set $\mathcal{V}$. Let $\mathbf{x}_{\mathcal{A}|\mathcal{B}}$ be the random variables in $\mathcal{A}$ conditioned on those in $\mathcal{B}$. The set $\mathcal{B}$ separates $\mathcal{A}$ and $\mathcal{C}$ if in the modified graph with $\mathcal{B}$ removed, there are no paths between nodes in the sets $\mathcal{A}$ and $\mathcal{C}$ (see Figure 2.6).

Definition 2.1.6. A stochastic process $\mathbf{x}$ is *Markov* with respect to the graph $\mathcal{G}$ if $\mathbf{x}_{\mathcal{A}|\mathcal{B}}$ and $\mathbf{x}_{\mathcal{C}|\mathcal{B}}$ are conditionally independent whenever $\mathcal{B}$ separates $\mathcal{A}$ and $\mathcal{C}$.

This definition of Markovianity constitutes a generalization of the concept as applied to a discrete time series. Indeed, a time series sampled at discrete instants can be viewed

[1] However, it may no longer be possible to read directly certain conditional independencies from the undirected graph.

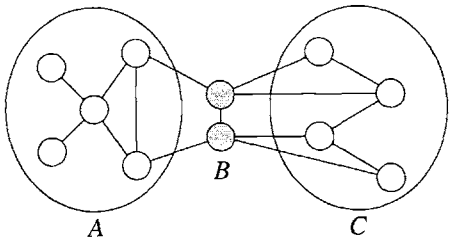


Figure 2.6. Illustration of the relation between conditional independence and graph separation. Here the set of nodes $\mathcal{B}$ separates $\mathcal{A}$ and $\mathcal{C}$, so that $\mathbf{x}_{\mathcal{A}|\mathcal{B}}$ and $\mathbf{x}_{\mathcal{C}|\mathcal{B}}$ are conditionally independent.

as a stochastic process defined on a chain. For such a graph, Definition 2.1.6 corresponds to the usual notion that the past and future are conditionally independent given the present.

A graph strongly constrains the distribution of a Markov process. Indeed, the *Hammersley-Clifford theorem* [21,79] guarantees that distributions of Markov processes over graphs can be expressed in factorized form as products of so-called *compatibility functions* defined over the cliques:

Theorem 2.1.1 (Hammersley-Clifford). Let $\mathcal{G}$ be a graph with a set of cliques $\mathbf{C}$. Suppose that a distribution[2] p over a discrete random vector $\mathbf{x}$ is formed as a normalized product of nonnegative functions over the cliques:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathbf{C}} \psi_C(\mathbf{x}) \tag{2.3}$$

where $\psi_C(\mathbf{x})$ is a compatibility function depending only on the subvector $\mathbf{x}_C = \{x_s \mid s \in C\}$; and $Z \triangleq \sum_{\mathbf{x}} \prod_{C \in \mathbf{C}} \psi_C(\mathbf{x})$ is the partition function. Then the underlying process $\mathbf{x}$ is Markov with respect to the graph. Conversely, the distribution p of any Markov random field over $\mathcal{G}$ that is strictly positive (i.e., $p(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathcal{X}^N$) can be represented in this factorized form.

Remarks: There a variety of proofs of this result [e.g., 21, 79]; see Clifford [38] for a historical overview. One of the most elegant proofs [79] uses the Möbius inversion formula [see, e.g., 28]. Note that this theorem generalizes the usual factorizations of Markov chains, for which the compatibility functions are formed by forward (or backward) transition functions defined on the edges (i.e., maximal cliques for a chain). See Lauritzen [121] for an example of a non-positive distribution (i.e., $p(\mathbf{x}) = 0$ for some $\mathbf{x} \in \mathcal{X}^N$) for which the converse is false.

[2]Strictly speaking, p is a probability mass function for discrete random variables; however, we will use distribution to mean the same thing.

■ 2.1.3 State estimation or inference

A problem that arises in many applications of interest is that of estimating the random vector $\mathbf{x} = \{x_s \mid s \in \mathcal{V}\}$ based on a set of noisy observations $\mathbf{y} = \{y_s \mid s \in \mathcal{V}\}$. For instance, in image processing or computer vision [65, 126], the vector $\mathbf{x}$ could represent an image defined on a grid, and $\mathbf{y}$ could represent a noisy or blurred version of this image. Similarly, in the context of channel coding [67, 71], the vector $\mathbf{x}$ would represent message bits, whereas $\mathbf{y}$ would correspond to the received bits.

In all cases, the goal is to *estimate* or to draw statistical *inferences* about the unobserved $\mathbf{x}$ based on the observations $\mathbf{y}$. The observation model can be formulated mathematically in the form of a conditional distribution. In particular, we assume that for each node $s \in \mathcal{V}$, the variable y_s is a noisy observation of x_s, specified by the conditional density $p(y_s|x_s)$. We assume that the observations $\mathbf{y}$ are conditionally independent given the hidden variables[3] $\mathbf{x}$, so that $p(\mathbf{y}|\mathbf{x}) = \prod_{s\in\mathcal{V}} p(y_s|x_s)$.

Of central interest for problems of estimation or inference is the posterior density $p(\mathbf{x}|\mathbf{y})$, which defines a variety of estimators:

1. The *maximum a posteriori* (MAP) estimate corresponds to the peak or mode of the posterior density — that is: $\widehat{\mathbf{x}}_{MAP} = \arg\max_{\mathbf{x}\in\mathcal{X}^N} p(\mathbf{x}|\mathbf{y})$

2. Also of interest are posterior marginals of a subset of variables. For instance, for a discrete process $\mathbf{x}$, the single node marginals are given by

$$p(x_s|\mathbf{y}) = \sum_{\mathbf{x}' \text{ s.t } x'_s = x_s} p(\mathbf{x}'|\mathbf{y}) \tag{2.4}$$

   Here the notation means summing over all configurations $\mathbf{x}' \in \mathcal{X}^N$ such that $x'_s = x_s$. For a continuous-valued process, this summation should be replaced by integration.

By combining the prior in equation (2.3) with the observation density via Bayes rule, we have:

$$p(\mathbf{x} \mid \mathbf{y}) = \frac{1}{Z} \prod_{C\in\mathbf{C}} \psi_C(\mathbf{x}) \prod_s p(y_s|x_s) \tag{2.5}$$

Note that each individual node forms a singleton clique, meaning that some of the factors in (2.3) may involve functions of each individual variable. As a consequence, the transformation from the prior distribution $p(\mathbf{x})$ of equation (2.3) to the conditional distribution $p(\mathbf{x} \mid \mathbf{y})$ of equation (2.5) is simply to modify the singleton factors of equation (2.3). As a result, from here onwards, we suppress explicit mention of measurements, since problems of estimation or inference for either $p(\mathbf{x})$ or $p(\mathbf{x} \mid \mathbf{y})$ are of identical structure and complexity.

[3]This assumption entails no loss of generality, since any observation that is a function of variables at multiple nodes can be merged into a clique potential that includes those nodes.

The computations of the MAP estimate or of the single-node marginals are both well-defined tasks. The focus of this thesis will be the latter task. Difficulties arise from different sources, depending on whether $\mathbf{x}$ is a discrete or continuous-valued process. For a continuous process, it may not be possible to evaluate analytically the necessary integrals. This difficulty is relevant even for small problems. For a discrete process, on the other hand, computing a marginal simply involves a discrete summation, which is a straightforward operation for small problems. Here the difficulty arises as the problem size grows. In particular, given a discrete-valued process on N nodes with $m \geq 2$ states, the number of terms in the summation of equation (2.4) explodes exponentially as m^{N-1}. Consequently, for sufficiently large graphs, it will be impossible to perform the discrete summation. A similar curse of dimensionality applies to the computation of the MAP estimate.

■ 2.1.4 Exact inference in trees

For a hidden Markov chain, there exist highly efficient algorithms for computing the MAP estimate, or the single-node marginals at each node. These algorithms exploit the Markov properties of a chain — namely, that the past and future are conditionally independent given the present — to perform the necessary computations in a recursive and hence efficient manner. For the linear-Gaussian problem, this formulation leads to the Rauch-Tung-Striebel smoother [146]. For a discrete-state hidden Markov chain, the resulting algorithm is known as the $\alpha - \beta$ algorithm in the speech processing literature [143].

Interestingly, these recursive algorithms can be generalized to trees, which are singly-connected graphs without cycles. (A chain is a special case of a tree.) An important property of trees is that their nodes can be assigned a partial ordering in terms of their depth in relation to an arbitrary node designated as the root. That is, the root is scale 0; the immediate descendants (i.e., children) of the root are scale 1; and so on down to the leaves (terminal nodes) of the tree. With this partial ordering, the most efficient implementation of a tree inference algorithm follows a two-pass form, first sweeping up from the leaves to the root, and then downwards from the root to the leaves. For a discrete process, the computational complexity of these algorithms is $\mathcal{O}(m^2 N)$. See Appendix A for more details about such tree algorithms.

■ 2.1.5 Junction tree representation

The set of cliques of a Markov chain are single nodes and pairs of adjacent nodes. In this case, the compatibility functions $\{\psi_C\}$ of equation (2.3) can always be written as a function of local marginal and conditional distributions. For example, the standard forward factorization of a Markov chain on three nodes is in terms of an initial distribution and transitions:

$$p(\mathbf{x}) = p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_2)$$

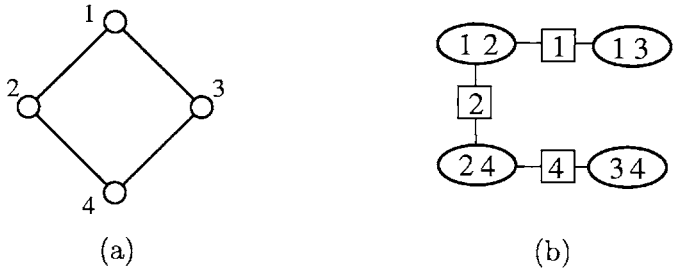
There is an alternative factorization that is symmetric with respect to the nodes — namely $p(\mathbf{x}) = [p(x_1, x_2)/p(x_1)p(x_2)]\ [p(x_2, x_3)/p(x_2)p(x_3)]p(x_1)p(x_2)p(x_3)$. More generally, the same kind of symmetric factorization holds for any tree-structured graph $\mathcal{T}$:

$$p(\mathbf{x}) = \prod_{s \in \mathcal{V}} p(x_s) \prod_{(s,t) \in \mathcal{E}} \frac{p(x_s, x_t)}{p(x_s)p(x_t)} \tag{2.6}$$

That is, for a tree, the compatibility functions of equation (2.3) can always be represented directly in terms of local marginal distributions: $\psi_s(x_s) = p(x_s)$ for each node $s \in \mathcal{V}$; and $\psi_{st}(x_s, x_t) = [p(x_s, x_t)/p(x_s)p(x_t)]$ for each edge $(s,t) \in \mathcal{E}$.

In contrast, for a graph with cycles, the compatibility functions do not, in general, have any direct correspondence with local marginal distributions on those same cliques.[4] However, such a correspondence does hold on a graph formed of suitably aggregated nodes, which is the subject of the *junction tree representation*. The basic idea is to cluster nodes within the original graph $\mathcal{G}$ so as to form a *clique tree* — that is, an acyclic graph whose nodes are formed by cliques of $\mathcal{G}$. We use the calligraphic $\mathcal{C}$ to refer to a given node of the clique tree (i.e., a given clique of $\mathcal{G}$).

Having formed a tree, it is tempting to simply apply a standard tree inference algorithm. However, the clique tree must satisfy an additional restriction so as to ensure consistency of probabilistic inference on the tree. To understand the source of this problem, consider the single loop on 4 nodes shown in Figure 2.7(a), as well as the clique tree (one of many possible) shown in Figure 2.7(b). Here ellipses represent nodes of the clique tree (i.e., cliques of the original graph), whereas the boxes represent *separator sets*, which correspond to intersections of nodes adjacent on the clique tree. Observe that node 3 occurs twice in the clique tree, once in each of the cliques $\{1, 3\}$



(a) (b)

Figure 2.7. A simple example showing the necessity of the running intersection property for probabilistic consistency. (a) Single loop on 4 nodes. (b) One possible clique tree for the graph in (a). This clique tree fails the running intersection property.

and $\{3, 4\}$. However, any tree inference algorithm applied to the clique tree of (b) will

[4]The simplest example to consider is the single loop on 4 nodes; here the pairwise compatibility functions can never correspond to P_{st}/P_sP_t.

not enforce the implicit constraint that the corresponding random variable x_3 in clique $\{1,3\}$ must match the x_3 in clique $\{3,4\}$. As a result, running a tree inference algorithm on the graph in (b) will not yield the correct results for the single loop of (a).

What is required is a mechanism for enforcing consistency among the different appearances of the same random variable. Note that the same problem does not arise for node 2, although it also appears in both of the two cliques $\{1,2\}$ and $\{2,4\}$. The difference is that node 2 also appears in all separator sets in the path between these two cliques, which provides a pipeline for transmitting and enforcing the associated consistency constraints. This motivates the following definition:

Definition 2.1.7. A clique tree has the *running intersection property* if for any two clique nodes $\mathcal{C}_1$ and $\mathcal{C}_2$, all nodes on the unique path joining them contain the intersection $\mathcal{C}_1 \cap \mathcal{C}_2$. A clique tree with this property is known as a *junction tree.*

For what type of graphs can one build junction trees? It is clear that no clique tree of the single loop in Figure 2.7(a) has the running intersection property. (Since the clique tree of Figure 2.7(b) does not satisfy running intersection, by a symmetry argument neither can any other clique tree.) An important result in graph theory establishes a correspondence between junction trees and triangulated graphs (see Definition 2.1.4).

Proposition 2.1.1. A graph $\mathcal{G}$ has a junction tree $\iff$ it is triangulated.

Proof. See Lauritzen [121]. □

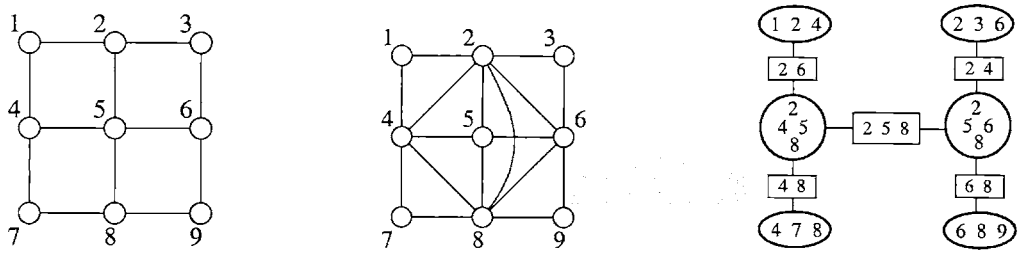
This proposition leads to a method for exact inference on arbitrary graphs:

Algorithm 2.1.1 (Junction tree).

1. Given a graph with cycles $\mathcal{G}$, triangulate it by adding edges as necessary.
2. Form a junction tree associated with the triangulated graph $\widetilde{\mathcal{G}}$.
3. Run a tree inference algorithm on the junction tree.

Although this procedure is sound in principle, its practical use is limited. For most applications of interest, the size of the cliques in the triangulated version $\widetilde{\mathcal{G}}$ grows with problem size. As a result, the state cardinality of the supernodes in the junction tree grows exponentially, meaning that applying tree algorithms rapidly becomes prohibitively complex. This explosion in the state cardinality is another demonstration of the intrinsic complexity of exact computations for graphs with cycles.

Example 2.1.1. To illustrate the junction tree procedure and its associated complexities, we consider the 3×3 grid shown in Figure 2.8(a). The first step is to form a triangulated version $\widetilde{\mathcal{G}}$, as shown in Figure 2.8(b). Note that the graph would not be



(a) (b) (c)

Figure 2.8. Illustration of junction tree procedure. (a) Original graph is a 3×3 grid. (b) Triangulated version of original graph. Note the two 4-cliques in the middle. (c) Corresponding junction tree for triangulated graph in (b), with maximal cliques depicted within ellipses, and separator sets within rectangles.

triangulated if the additional edge joining nodes 2 and 8 (shown in a dashed line) were not present. Without this edge, the 4-cycle $(2 - 4 - 8 - 6 - 2)$ would lack a chord. As a result of this additional edge, the junction tree has two 4-cliques in the middle, as shown in Figure 2.8(c). Consequently, running a tree inference algorithm on the junction tree involves dealing with variables with state cardinalities of m^4. This difficulty only worsens as the grid size grows.

Despite its limited practical use, the junction tree procedure provides conceptual insight into the inherent complexity of a given distribution on a graph. In particular, it gives rise to an alternative representation of the distribution, in terms of local marginal distributions on maximal cliques and separator sets. That is,

$$p(\mathbf{x}) = \frac{\prod_{C \in \mathbf{C}} p(\mathbf{x}_C)}{\prod_{S \in \mathbf{S}} p(\mathbf{x}_S)} \tag{2.7}$$

where $\mathbf{C}$ is the set of all maximal cliques of $\widetilde{\mathcal{G}}$, and $\mathbf{S}$ is the associated set of separators. Unlike the representation of equation (2.3), equation (2.7) provides a decomposition directly in terms of local marginal distributions. The price to be paid is that the decomposition involves functions defined over larger clusters of variables. Note that equation (2.6) is a particular case of this decomposition, where the maximal cliques are the edges of the ordinary tree, and the separator sets correspond to nodes with degree greater than one.

■ 2.2 Exponential families and information geometry

Exponential families of distributions and their associated geometry have been studied extensively in applied probability theory and statistics. Work in this area dates back to

Rao [145] in 1945, who developed the geometric role of the Fisher information matrix. Subsequent contributions were made by a variety of people, including Chentsov [32,33], Csiszár [42–45], Barndorff-Nielson [13] and Amari [5–7]. This section contains a brief introduction to this body of theory, which is often referred to as *information geometry*. We emphasize only those concepts necessary for our subsequent development; see the references above, or the edited collection of papers in [82] for further details. Although information geometry applies to any exponential family of distributions, we focus here on such distributions in the specific context of graphical models.

■ 2.2.1 Exponential representations

Equation (2.3) decomposes a graph distribution as a product of compatibility functions defined on the cliques. A related representation is the Gibbs form, in which a distribution is specified as the exponential of a sum of functions on the cliques. In the context of graphical models, an exponential family constitutes a collection of such Gibbs distributions:

$$p(\mathbf{x};\theta) = \exp\Big\{\sum_{\alpha} \theta_\alpha \phi_\alpha(\mathbf{x}) - \Phi(\theta)\Big\} \tag{2.8a}$$

$$\Phi(\theta) = \log\Big(\sum_{\mathbf{x}\in\mathcal{X}^N} \exp\{\sum_{\alpha} \theta_\alpha \phi_\alpha(\mathbf{x})\}\Big) \tag{2.8b}$$

The quantity Φ defined in equation (2.8b) is the *log partition function* that serves to normalize the distribution; when the sample space $\mathcal{X}^N$ is continuous, the summation defining Φ should be replaced by an integral.

Any exponential family is specified by a collection of potential functions $\{\phi_\alpha \mid \alpha \in \mathcal{A}\}$, where $\mathcal{A}$ is a finite index set. The domain of the *exponential parameter vector* θ is the set

$$\Theta \triangleq \{\theta \in \mathbb{R}^{|\mathcal{A}|} \mid \Phi(\theta) < \infty\}$$

In the discrete case, this imposes no restrictions (i.e., $\Theta = \mathbb{R}^{|\mathcal{A}|}$); in continuous examples, Θ can be a strict subset of $\mathbb{R}^{|\mathcal{A}|}$. In this thesis, we focus primarily on the discrete case.

Each parameter vector $\theta \in \Theta$ indexes a particular member $p(\mathbf{x};\theta)$ of the family, assuming that the set of clique potentials $\boldsymbol{\phi} = \{\phi_\alpha\}$ is fixed. With some abuse of notation, we will often use the parameter vector θ itself as a shorthand for the associated distribution.

Minimal representations

It is typical to define an exponential family with a collection of functions $\boldsymbol{\phi} = \{\phi_\alpha\}$ that are linearly independent. Indeed, if there were any linear dependencies, they could be eliminated without sacrificing any expressive power of the exponential model. This condition gives rise to a so-called *minimal representation* [e.g., 13], in which there is a

unique parameter vector θ associated with each distribution. In this case, the *dimension* of the exponential family, denoted by $d(\theta)$, is given by $|\mathcal{A}|$.

To illustrate these definitions, we consider some simple examples:

Example 2.2.1. Consider a scalar Gaussian random variable $x \sim \mathcal{N}(\mu, \sigma^2)$. Then its density has the exponential representation

$$p(x;\theta) = \exp\{\theta_1 x + \theta_2 x^2 - \Phi(\theta)\} \tag{2.9}$$

I.e. here we have $\phi_1(x) = x$ and $\phi_2(x) = x^2$. By completing the square, we obtain relations between the exponential parameters (θ_1, θ_2) and the mean and variance — namely, $\theta_2 = -1/[2\sigma^2]$ and $\theta_1 = \mu/\sigma^2$. Here the dimension of the family is $d(\theta) = 2$. Moreover, the domain of θ is the half plane

$$\Theta = \{(\theta_1, \theta_2) \in \mathbb{R}^2 \mid \theta_2 < 0\}$$

The restriction on θ_2 is required so that the associated integral defining the log partition function — namely, $\Phi(\theta) = \int_{-\infty}^{\infty} \exp\{\theta_1 x + \theta_2 x^2\} dx$ — is finite.

Example 2.2.2. Now consider a binary process (i.e., $\mathbf{x} \in \{0,1\}^N$) defined on a graph with pairwise maximal cliques. The standard (minimal) representation corresponds to the *Boltzmann machine* [e.g., 105], also known as the *Ising model* in statistical physics [15, 31]:

$$p(\mathbf{x};\theta) = \exp\Big\{\sum_{s\in\mathcal{V}} \theta_s x_s + \sum_{(s,t)\in\mathcal{E}} \theta_{st} x_s x_t - \Phi(\theta)\Big\} \tag{2.10}$$

where θ_{st} is the strength of edge (s,t), and θ_s is the node parameter for node s. In this case, $d(\theta) = |\mathcal{V}| + |\mathcal{E}| = N + |\mathcal{E}|$, and the domain Θ of θ is all of $\mathbb{R}^{d(\theta)}$.

Examples 2.2.1 and 2.2.2 illustrate that the sample space $\mathcal{X}$ is critical in assessing the linear independence of a set of functions $\{\phi_\alpha\}$, and hence the minimality of the representation. In Example 2.2.1, the functions x and x^2 are linearly independent over $\mathbb{R}$, so that equation (2.9) constitutes a minimal representation. In contrast, these same functions are *not* linearly independent over $\{0,1\}$. As a consequence, including x_s^2 terms in the Ising model of Example 2.2.2 would lead to an overcomplete representation.

Example 2.2.3. We now consider an extension of Example 2.2.2, with $\mathbf{x} \in \{0,1\}^N$. The Ising model corresponds to pairwise maximal cliques. To incorporate higher-order cliques (e.g., the 3-clique $\{s,t,u\}$), we add a multinomial of the form $x_s x_t x_u$, with corresponding exponential parameter θ_{stu}. Cliques of higher order are incorporated in a similar fashion, so that the minimal representation of the most general distribution (i.e., possibly on the complete graph) is of the form:

$$p(\mathbf{x};\theta) = \exp\Big\{\sum_{s=1}^{n} \theta_s x_s + \sum_{s<t} \theta_{st} x_s x_t + \sum_{s<t<u} \theta_{stu} x_s x_t x_u + \ldots \\ \ldots + \theta_{1\ldots N} x_1 x_2 \cdots x_N - \Phi(\theta)\Big\} \tag{2.11}$$

It can be verified that the set of functions $\{x_s\}_{s=1}^N \cup \{x_s x_t\}_{s<t} \cup \ldots \cup \{x_1 \cdots x_n\}$ are linearly dependent over $\{0,1\}^N$, and span the space of all real-valued functions on $\{0,1\}^N$. Hence the dimension of the family is given by:

$$d(\theta) = \binom{N}{1} + \binom{N}{2} + \ldots + \binom{N}{N} = 2^N - 1$$

Since any distribution on the binary vector $\mathbf{x} \in \{0,1\}^N$ has $2^N - 1$ degrees of freedom,[5] we see that any distribution can be represented in the form equation (2.11).

Of course, the Ising model of equation (2.10) arises as a particular case of equation (2.11), where we place the restriction that $\theta_J = 0$ for all subsets $J \subset \{1, \ldots, N\}$ of size $|J| > 2$. Indeed, a nested sequence of exponential families $\mathcal{F}_k$ can be defined by imposing restrictions of the form $\theta_J = 0$ for all $|J| > k$, for $k = 1, \ldots, N-1$. See Amari [6] for details on such nested families. In the context of graphical models, these restrictions correspond to a limit on the maximal clique size in the associated graph.

Examples 2.2.2 and 2.2.3 can be extended to minimal exponential representations of m-ary processes ($m > 2$) as well. In particular, the analog of the Ising model for an m-ary process is specified in terms of the functions

$$\mathcal{R}(s) \triangleq \{x_s^a \mid a = 1, \ldots, m-1\} \qquad \text{for } s \in \mathcal{V} \tag{2.12a}$$

$$\mathcal{R}(s,t) \triangleq \{x_s^a x_t^b \mid a, b = 1, \ldots, m-1\} \qquad \text{for } (s,t) \in \mathcal{E} \tag{2.12b}$$

The dimension of this exponential family is given by $d(\theta) = (m-1)\,N + (m-1)^2|\mathcal{E}|$. Incorporating higher order cliques entails adding higher degree multinomials to the clique functions of equation (2.12). This procedure, though conceptually straightforward, can lead to cumbersome notation. See Amari [6] for further details.

Overcomplete representations

In addition to such a minimal parameterization, parts of our analysis (especially Chapter 5) make use of an *overcomplete representation*, in which the $\{\phi_\alpha\}$ are linearly dependent. In this case, the lack of linear independence means that there exists an entire manifold of parameter vectors θ, each associated with the same distribution.

Example 2.2.4 (Overcomplete representation of binary process). An overcomplete representation of a binary process on a graph with pairwise cliques entails specifying a 2-vector for each node $s \in \mathcal{V}$, and a 2×2 matrix of values for each edge (s,t) in the graph. To do so, we choose our clique potentials as indicator functions: that is, the collection of functions $\{\delta(x_s = j) \mid j = 0,1\}$ for each node $s \in \mathcal{V}$, and $\{\delta(x_s = j)\delta(x_t = k) \mid j,k = 0,1\}$ for each edge $(s,t) \in \mathcal{E}$. Here, the indicator or delta

[5] Any distribution can be represented by a 2^N vector, and we lose one degree of freedom due to the normalization $\sum_{\mathbf{x}} p(\mathbf{x}) = 1$.

function $\delta(x_s = j)$ is equal to 1 when node x_s takes the state value j, and 0 otherwise. The corresponding representation would be of the form

$$p(\mathbf{x};\theta) = \exp\Big\{\sum_{s\in\mathcal{V}}\sum_{j=0}^{1}\theta_{s;j}\delta(x_s = j) + \sum_{(s,t)\in\mathcal{E}}\sum_{j,k=0}^{1}\theta_{st;jk}\delta(x_s = j)\delta(x_t = k) - \Phi(\theta)\Big\} \tag{2.13}$$

where θ is the corresponding overcomplete parameter vector.

It is straightforward to generalize this type of overcomplete representation in terms of indicator functions to m-ary processes.

Different types of binary potentials

Given a distribution over a binary vector defined by a graph with pairwise cliques, it will often be useful to specify potential types from one of the following classes:

(a) in a graph with *attractive* potentials, all pairs of neighboring random variables are more likely to take the same values than opposite values.

(b) conversely, in a graph with *repulsive* potentials, all neighboring variables are encouraged to take opposite values.

(c) a graph with *mixed or frustrated* potentials consists of a combination of attractive and repulsive potentials.

In the statistical physics literature [e.g., 15,31], these types of distributions are referred to as ferromagnetic, anti-ferromagnetic, and paramagnetic respectively.

The convention of this thesis will be that a binary random variable x_s takes values in $\{0,1\}$. In order to specify potential types, it is useful to consider a so-called *spin representation* in which a binary random variable u_s takes values in $\{-1,+1\}$. The term "spin" comes from the statistical physics literature [31]; for instance, one can think of u_s as giving the orientation (up or down) of a magnet at node s. We let

$$p(\mathbf{u};\omega) = \exp\{\sum_{s}\omega_s u_s + \sum_{(s,t)}\omega_{st}u_s u_t - \Phi(\omega)\} \tag{2.14}$$

be a minimal exponential representation corresponding to the spin vector $\mathbf{u}\in\{0,1\}^N$, where ω is the associated vector of exponential parameters. In this spin representation, the nature of the interaction between u_s and u_t is determined entirely by the sign of ω_{st}. In particular, the potential is attractive (respectively repulsive) if and only if ω_{st} is positive (respectively negative).[6]

[6]Note that the same statement does not hold for the exponential parameter θ_{st} in a $\{0,1\}$ representation (see, e.g., equation (2.10)). For this representation, if we disregard the single node parameters θ_s, setting $\theta_{st} > 0$ places higher weight on the configuration $(x_s, x_t) = (1,1)$, but equal weights on the remaining configurations $\{(0,0),\ (1,0),\ (0,1)\ \}$.

Thus, a spin representation is convenient for constructing random distributions on graphs with particular types of potentials. Moreover, any spin parameter ω specifies a unique exponential parameter θ. In particular, we substitute the relation $u_s = 2x_s - 1$, which converts from $\{0,1\}$ variables to spins, into equation (2.14), and then equate coefficients with the Ising model of equation (2.10). In this way, we obtain the following relations:

$$\theta_s = 2\left[\omega_s - \sum_{t \in \mathcal{N}(s)} \omega_{st}\right] \tag{2.15a}$$

$$\theta_{st} = 4\omega_{st} \tag{2.15b}$$

We now define for future reference a few ensembles of random potentials. In all cases, we set the node parameters $\omega_s = 0$ for all nodes $s \in \mathcal{V}$. Let $\mathcal{U}[a,b]$ denote the uniform distribution on the interval $[a,b]$. Given a particular *edge weight* $d > 0$, we then choose the edge parameters as follows:

(a) for the *uniform attractive ensemble* with edge weight $d > 0$, set $\omega_{st} \sim \mathcal{U}[0,d]$ independently for each edge $(s,t) \in \mathcal{E}$

(b) for the *uniform repulsive ensemble*, set $\omega_{st} \sim \mathcal{U}[-d,0]$ independently for each edge $(s,t) \in \mathcal{E}$

(c) for the *uniform mixed ensemble*, set $\omega_{st} \sim \mathcal{U}[-d,d]$ independently for each edge $(s,t) \in \mathcal{E}$

Given the (randomly-chosen) distribution $p(\mathbf{u};\omega)$ specified in terms of the spin parameter ω, we then convert to the distribution $p(\mathbf{x};\theta)$, where θ is obtained from ω via equation (2.15).

■ 2.2.2 Properties of Φ

In this section, we develop some important properties of the log partition function Φ defined in equation (2.8b), including its convexity. Given a distribution $p(\mathbf{x};\theta)$ and a function $f : \mathcal{X}^N \to \mathbb{R}$, we define the expectation of $f(\mathbf{x})$ with respect to $p(\mathbf{x};\theta)$ as follows:

$$\mathbb{E}_\theta[f(\mathbf{x})] = \sum_{\mathbf{x} \in \mathcal{X}^N} p(\mathbf{x};\theta)\, f(\mathbf{x}) \tag{2.16}$$

When the sample space $\mathcal{X}$ is continuous, this summation should be replaced by an integral.

With this notation, we can show that the function Φ is closely related to the cumulant generating function[7] associated with the random variables $\{\phi_\alpha(\mathbf{x})\}$. In particular,

[7]Another interpretation of Φ arises in statistical physics, where it is known as the *Helmholtz free energy* [31,135].

given a parameter vector $\theta \in \mathbb{R}^{d(\theta)}$ and another vector $\epsilon \in \mathbb{R}^{d(\theta)}$, we compute:

$$\log\Big(\mathbb{E}_\theta\big[\exp\{\sum_\alpha \epsilon_\alpha \phi_\alpha(\mathbf{x})\}\big]\Big) \;=\; \Phi(\theta+\epsilon) - \Phi(\theta) \tag{2.17}$$

The quantity on the left-hand side is the cumulant generating function (or the logarithm of the moment generating function) [80]. Equation (2.17) shows that this cumulant generating function is equal to the difference between the function Φ at two distinct values.

Using this relation, it can be shown that derivatives of Φ with respect to θ correspond to the cumulants of $\{\phi_\alpha(\mathbf{x})\}$. For example,

$$\frac{\partial \Phi}{\partial \theta_\alpha}(\theta) \;=\; \mathbb{E}_\theta[\phi_\alpha] \tag{2.18a}$$

$$\frac{\partial^2 \Phi}{\partial \theta_\alpha \partial \theta_\beta}(\theta) \;=\; \mathrm{cov}_\theta\{\phi_\alpha, \phi_\beta\} \;\triangleq\; \mathbb{E}_\theta\Big\{\big(\phi_\beta - \mathbb{E}_\theta[\phi_\alpha]\big)\big(\phi_\beta - \mathbb{E}_\theta[\phi_\beta]\big)\Big\} \tag{2.18b}$$

are the first and second order cumulants. In general, let $\mathrm{cum}_\theta\{\phi_{\alpha_1}, \ldots, \phi_{\alpha_k}\}$ denote the k^{th}-order cumulant of $\{\phi_{\alpha_1}, \ldots, \phi_{\alpha_k}\}$ under $p(\mathbf{x};\theta)$. Then higher order cumulants are defined recursively by successive differentiation of lower order cumulants; e.g., $\mathrm{cum}_\theta\{\phi_{\alpha_1}, \phi_{\alpha_2}, \phi_{\alpha_3}\} = \frac{\partial}{\partial \theta_{\alpha_3}}\big[\mathrm{cum}_\theta\{\phi_{\alpha_1}, \phi_{\alpha_2}\}\big]$.

The second order cumulant in equation (2.18b) reveals an important property of the log partition function:

Lemma 2.2.1. The function Φ is convex as a function of θ. The convexity is strict when the representation is minimal.

Proof. Note that the quantity in (2.18b) is an element of the Fisher information matrix $(-\mathbb{E}_\theta\{\frac{\partial^2 \log p(\mathbf{x};\theta)}{\partial \theta^2}\})$. Therefore, the Hessian $\nabla^2\Phi$ is positive semi-definite (strictly positive definite for a minimal representation), so that Φ is convex (respectively strictly convex). □

The convexity of Φ will play a central role in subsequent geometric developments.

■ 2.2.3 Riemannian geometry of exponential families

An important feature of exponential families of distributions is their geometric structure. In this section, we provide a very brief introduction to the differential geometry of these families. See Amari [5–7] and the edited collection of papers [82] for further details.

Consider an exponential representation in terms of the $d(\theta)$-dimensional parameter θ, assumed to be minimal. For each $\theta \in \Theta$, we have $p(\mathbf{x};\theta) > 0$ for all $\mathbf{x} \in \mathcal{X}^N$. Therefore, we can associate with each point $\theta \in \Theta$ a function — namely, the log distribution $\log p(\mathbf{x};\theta)$. Under suitable regularity conditions [13], this association defines a $d(\theta)$-dimensional differential manifold $\mathcal{M}$ of functions $\{\log p(\mathbf{x};\theta) \mid \theta \in \Theta\}$. When

the sample space $\mathcal{X}^N$ is discrete and hence finite, we can view $\mathcal{M}$ as embedded within $\mathbb{R}^{|\mathcal{X}^N|}$; otherwise, it is embedded within an infinite-dimensional function space. The mapping $\theta \mapsto \log p(\mathbf{x};\theta)$ is the co-ordinate mapping of the manifold, as illustrated in Figure 2.9.

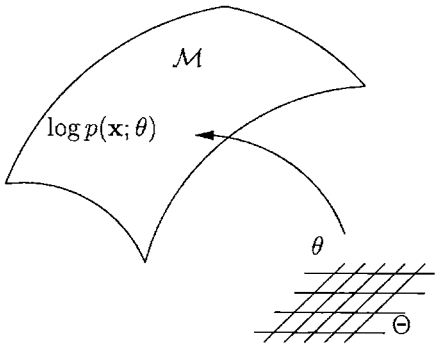


Figure 2.9. The exponential parameters θ serve as the co-ordinates for the $d(\theta)$-dimensional differential manifold of log distributions $\log p(\mathbf{x};\theta)$. Associated with each $\theta \in \Theta$ is a log distribution $\log p(\mathbf{x};\theta)$; the association $\theta \mapsto \log p(\mathbf{x};\theta)$ defines the co-ordinate mapping.

Given a line $\theta(t)$ in Θ, we can consider the curve in $\mathcal{M}$ defined by its image $\log p(\mathbf{x};\theta(t))$ under the co-ordinate mapping. The set of all tangent vectors to such curves at a particular value of θ defines the *tangent space* of $\mathcal{M}$ at the point $\log p(\mathbf{x};\theta)$. It can be seen that this tangent space is a $d(\theta)$-dimensional vector space. In particular, letting $\mathbf{e}_\alpha$ be a $d(\theta)$-vector of zeros with a single one in element α and zero elsewhere, consider the co-ordinate line $\theta(s;\alpha) = (1-s)\theta + s\mathbf{e}_\alpha$. By straightforward calculations, the tangent vector $\mathbf{t}_\alpha$ to the curve $\log p(\mathbf{x};\theta(s;\alpha))$ is given by

$$\mathbf{t}_\alpha = \frac{\partial}{\partial \theta_\alpha} \log p(\mathbf{x};\theta) \;=\; \phi_\alpha(\mathbf{x}) - \mathbb{E}_\theta[\phi_\alpha]\, \mathbf{1}(\mathbf{x}) \tag{2.19}$$

where $\mathbf{1}(\mathbf{x}) = 1$ for all $\mathbf{x} \in \mathcal{X}^N$. In computing this derivative, we have used equation (2.18a). It can be shown that the set $\{\mathbf{t}_\alpha \mid \alpha \in \mathcal{A}\}$ spans the tangent space at $\log p(\mathbf{x};\theta)$.

We now use $p(\mathbf{x};\theta)$ to define a weighted inner product on the tangent space. Of course, it suffices to specify the inner product for any pair $\{\mathbf{t}_\alpha, \mathbf{t}_\beta\}$, which we do as follows:

$$\langle \mathbf{t}_\alpha,\, \mathbf{t}_\beta \rangle_\theta = \mathbb{E}_\theta \Big[\frac{\partial}{\partial \theta_\alpha} \log p(\mathbf{x};\theta) \frac{\partial}{\partial \theta_\beta} \log p(\mathbf{x};\theta) \Big] = \mathrm{cov}_\theta\{\phi_\alpha,\, \phi_\beta\} \tag{2.20}$$

where we have used equations (2.18b) and (2.19) to see the equivalence to a covariance.

The quantities $g_{\alpha\beta}(\theta) \triangleq \text{cov}_\theta\{\phi_\alpha, \phi_\beta\}$ are elements of the *Fisher information matrix*, denoted by $G(\theta)$. For a minimal θ-representation, it can be seen that the Fisher information matrix is strictly positive definite for all θ. It therefore defines a Riemannian metric, with the squared distance between the distribution θ and an infinitesimally perturbed distribution $\theta + \Delta$ given by

$$[d(\theta, \theta + \Delta)]^2 = \sum_{\alpha,\beta} g_{\alpha\beta}(\theta)\Delta_\alpha\Delta_\beta = \Delta^T G(\theta)\Delta = \|\Delta\|^2_{G(\theta)} \tag{2.21}$$

The Fisher information matrix and the induced distance function of equation (2.21) also play important roles in other contexts, as we will explore in subsequent sections.

■ 2.2.4 Legendre transform and dual variables

The aspect of information geometry that sets it apart from classical Riemannian geometry is the existence of a dual parameterization, coupled to the exponential θ-parameterization. The coupling arises from convex duality associated with the log partition function Φ. The monograph of Rockafellar [151] provides a comprehensive treatment of convex duality; a more elementary and geometric treatment of duality can be found in Bertsekas [20]. In this section, we exploit the convexity of Φ to apply notions from convex analysis — in particular, the Legendre transform — from which we obtain a second set of parameters dual to the exponential θ-representation. In a later section, we use the Legendre duality to develop a geometric interpretation of the presence or absence of certain cliques in a graph-structured distribution.

The convexity of Φ allows us to apply the Legendre transform. Here we assume that the domain Θ of θ is either all of $\mathbb{R}^{|\mathcal{A}|}$, or some convex subset. The Legendre dual of Φ is defined as:

$$\Psi(\eta) = \sup_\theta \{\eta^T\theta - \Phi(\theta)\} \tag{2.22}$$

where η is a vector of the same dimension as the exponential parameter θ. Since the quantity to be maximized (i.e., $\eta^T\theta - \Phi(\theta)$) is strictly concave as a function of θ, the supremum in equation (2.22) is attained at some point $\widehat{\theta}$. Taking derivatives to find stationary points, and making use of equation (2.18a) yields the defining equation:

$$\eta_\alpha \equiv \eta_\alpha(\widehat{\theta}) = \mathbb{E}_{\widehat{\theta}}[\phi_\alpha] \tag{2.23}$$

Since they are obtained by taking expectations, these dual variables η are often referred to as the *mean parameters*. Substituting the relation in (2.23) back into equation (2.22) yields the relation

$$\Psi(\eta(\widehat{\theta})) = \sum_\alpha \widehat{\theta}_\alpha \mathbb{E}_{\widehat{\theta}}[\phi_\alpha] - \Phi(\widehat{\theta}) = \mathbb{E}_{\widehat{\theta}}[\log p(\mathbf{x}; \widehat{\theta})] \tag{2.24}$$

so that the Legendre dual Ψ is the negative entropy. Note that Ψ is itself a convex function, so that we can again apply the Legendre transform. It is not difficult to show

that applying the Legendre transform twice in this manner recovers the log partition function; that is,

$$\Phi(\theta) = \sup_{\eta} \{\theta^T \eta - \Psi(\eta)\} \tag{2.25}$$

The Legendre duality of Φ gives rise to a mapping $\Lambda : \theta \mapsto \eta$, defined explicitly by

$$[\Lambda(\theta)]_\alpha = \frac{\partial \Phi(\theta)}{\partial \theta_\alpha} = \mathbb{E}_\theta[\phi_\alpha] \tag{2.26}$$

For a minimal representation, Lemma 2.2.1 guarantees that Φ is strictly convex, in which case the mapping is one-to-one [151]. It is therefore invertible on its image, with the inverse map $\Lambda^{-1} : \eta \mapsto \theta$ defined by the corresponding relation

$$[\Lambda^{-1}(\eta)]_\alpha = \frac{\partial \Psi(\eta)}{\partial \eta_\alpha} \tag{2.27}$$

On the basis of these mappings, we can specify distributions either in terms of the exponential parameter θ, or the associated dual parameter η. Given a valid dual parameter η in a minimal representation, the quantity $p(\mathbf{x}; \eta)$ denotes the equivalent exponential distribution $p(\mathbf{x}; \Lambda^{-1}(\eta))$.

A few examples help to give intuition for the Legendre mapping:

Example 2.2.5 (Legendre transform for Gaussian). Let $\mathbf{x} \sim \mathcal{N}(0, P)$ be a zero-mean Gaussian random vector with covariance P. Then the density has an exponential representation of the form:

$$p(\mathbf{x}; \theta) = \exp\left\{\frac{1}{2}\sum_{s=1}^{N} \theta_{ss} x_s^2 + \sum_{s<t} \theta_{st} x_s x_t - \Phi(\theta)\right\} \tag{2.28}$$

Here θ specifies elements of the inverse covariance (i.e., $P_{st}^{-1} = -\theta_{st}$).[8] From equation (2.26), the dual variables are given by:

$$\begin{aligned} \eta_{ss} &= \mathbb{E}_\theta[x_s^2] = \mathrm{var}(x_s) \\ \eta_{st} &= \mathbb{E}_\theta[x_s x_t] = \mathrm{cov}(x_s, x_t) \end{aligned}$$

so that η specifies elements of the covariance matrix P. That is, the Legendre transform maps back and forth between the matrix inverse pair P and $-P^{-1}$.

Example 2.2.6. We now return to the Ising model (see Example 2.2.2), where the random vector $\mathbf{x} \in \{0, 1\}^N$ has a distribution of the form

$$p(\mathbf{x}; \theta) = \exp\Big\{\sum_{s \in \mathcal{V}} \theta_s x_s + \sum_{(s,t) \in \mathcal{E}} \theta_{st} x_s x_t - \Phi(\theta)\Big\}$$

[8] We require that θ belongs to the set for which $P^{-1}(\theta) > 0$.

In this case, the dual variables are given by the expectations:

$$\begin{aligned} \eta_s &= \mathbb{E}_\theta[x_s] \equiv p(x_s = 1; \theta) \\ \eta_{st} &= \mathbb{E}_\theta[x_s x_t] \equiv p(x_s = 1, x_t = 1; \theta) \end{aligned}$$

That is, the dual variables correspond to particular marginal probabilities at individual nodes, and pairs of nodes (s,t). Note that the dual variables fully specify the single node marginals, and pairwise marginals for $(s,t) \in \mathcal{E}$:

$$p(x_s;\theta) = [(1-\eta_s) \;\; \eta_s]^T; \qquad p(x_s,x_t;\theta) = \begin{bmatrix} (1+\eta_{st}-\eta_s-\eta_t) & \eta_t - \eta_{st} \\ \eta_s - \eta_{st} & \eta_{st} \end{bmatrix}$$

When the underlying graph $\mathcal{G}$ of the Ising model is a tree, these local marginals determine the full distribution $p(\mathbf{x};\theta)$ explicitly via the tree factorization given in equation (2.6). For a graph with cycles, such a local construction is not possible; however, whenever the dual variables $\{\eta_s, \eta_{st}\}$ belong to the range of the Legendre transform, then the invertibility of this transform guarantees that the dual variables still completely specify the distribution.

The Legendre mapping is also closely related to the Fisher information matrix. In particular, by differentiating equation (2.23) with respect to θ_β, we see that the Jacobian of the mapping $\Lambda : \theta \mapsto \eta$ is given by the Fisher information matrix $[G(\theta)]_{\alpha\beta} = \mathrm{cov}_\theta\{\phi_\alpha, \phi_\beta\}$. That is,

$$\eta(\theta + \Delta\theta) - \eta(\theta) \approx G(\theta)\Delta\theta \tag{2.29}$$

up to first order in the perturbation $\Delta\theta$. Similarly, the inverse Fisher information matrix G^{-1}, which is guaranteed to exist when Φ is strictly convex, corresponds to the Jacobian of the inverse mapping $\Lambda^{-1} : \eta \mapsto \theta$.

■ 2.2.5 Geometric consequences for graphical models

In the specific context of graphical models, the Legendre duality also leads to an interesting geometric interpretation of the presence or absence of given clique potentials. In particular, consider the constrained maximum entropy problem:

$$\begin{cases} \max_p H(p) \\ \sum_{\mathbf{x}} p(\mathbf{x})\phi_\alpha(\mathbf{x}) \leq \eta_\alpha \end{cases} \tag{2.30}$$

Geometrically, the maximization takes place over a polyhedral set, formed by the intersection of the probability simplex $\mathcal{P} = \{p(\mathbf{x}) \mid 0 \leq p(\mathbf{x}) \leq 1; \sum_{\mathbf{x}} p(\mathbf{x}) = 1\}$ with the hyperplane constraints $\{p(\mathbf{x}) \mid \sum_{\mathbf{x}} p(\mathbf{x})\phi_\alpha(\mathbf{x}) \leq \eta_\alpha\}$. It is well-known [41] that the solution to this problem assumes the familiar Gibbs form of equation (2.8), where the exponential parameter θ_α now corresponds to the Lagrange multiplier associated with

the constraint $\sum_{\mathbf{x}} p(\mathbf{x})\phi_\alpha(\mathbf{x}) \leq \eta_\alpha$. That is, it reflects the sensitivity of the problem to perturbations in the associated η_α constraint.

By the Karush-Kuhn-Tucker conditions [20], the Lagrange multiplier θ_α is zero whenever the η_α-constraint is inactive (i.e., not met with equality.) On this basis, the presence or absence of particular cliques in a graphical model can be related to hyperplane constraints. In particular, we can add a given clique potential ϕ_β by imposing a hyperplane constraint of the form $\sum_{\mathbf{x}} p(\mathbf{x})\phi_\beta(\mathbf{x}) \leq \eta_\beta$. Progressively lowering η_β so as to tighten the constraint will eventually ensure that the associated Lagrange multiplier is non-zero, meaning that the clique potential ϕ_β appears in the exponential representation with a non-zero weight θ_β. Conversely, we can remove a given clique from the graphical distribution by loosening the associated constraint. Eventually, the constraint will become inactive, so that the Lagrange multiplier θ_β is zero and the clique is effectively absent from the graph.

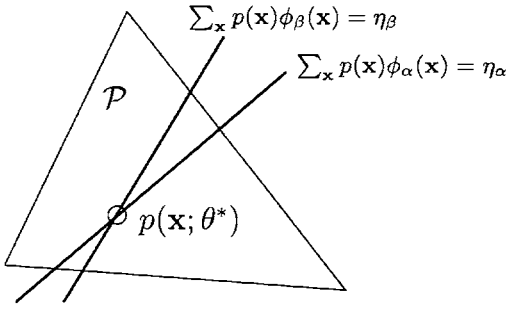


Figure 2.10. Geometry of graph-structured distributions. Distributions $p(\mathbf{x})$ are restricted to the simplex $\mathcal{P} = \{\, p(\mathbf{x}) \mid 0 \leq p(\mathbf{x}) \leq 1;\ \sum_{\mathbf{x}} p(\mathbf{x}) = 1 \,\}$, and lie on the intersections of hyperplane constraint sets $\{\, p(\mathbf{x}) \mid \sum_{\mathbf{x}} p(\mathbf{x})\phi_\alpha(\mathbf{x}) = \eta_\alpha \,\}$ imposed by the clique potentials $\{\phi_\alpha\}$.

■ 2.2.6 Kullback-Leibler divergence and Fisher information

The Kullback-Leibler divergence [118] can be viewed as a measure of the "distance" between two distributions. For a discrete random vector, its usual definition [see, e.g., 41] is $D(p \parallel q) = \sum_{\mathbf{x}\in\mathcal{X}^N} p(\mathbf{x})[\log p(\mathbf{x}) - \log q(\mathbf{x})]$. The definition shows that it is not a true distance, since (for example) it is not symmetric in p and q. However, it can be shown using Jensen's inequality that $D(p \parallel q) \geq 0$ for all p and q, with equality if and only if $p = q$.

With a minor abuse of notation[9], we let $D(\theta \parallel \theta^*)$ denote the KL divergence between two distributions in exponential form $p(\mathbf{x};\theta)$ and $p(\mathbf{x};\theta^*)$. The exponential parameter-

[9]Strictly speaking, the divergence applies to distributions $p(\mathbf{x};\theta)$ and $p(\mathbf{x};\theta^*)$, and not to the parameters θ and θ^* themselves.

ization leads to an alternative representation of this KL divergence:

$$\begin{aligned} D(\theta \,\|\, \theta^*) &= \sum_\alpha \mathbb{E}_\theta[\phi_\alpha]\big[\theta - \theta^*\big]_\alpha + \Phi(\theta^*) - \Phi(\theta) \\ &= \eta^T[\theta - \theta^*] + \Phi(\theta^*) - \Phi(\theta) \end{aligned} \tag{2.31}$$

where $\eta_\alpha = [\Lambda(\theta)]_\alpha$. That is, the pair (θ, η) are dually coupled via the Legendre transform.

Equation (2.31) shows that the KL divergence $D(\theta \,\|\, \theta^*)$ can be viewed as a Bregman distance, induced by the convex log partition function Φ. In particular, since $\frac{\partial \Phi(\theta)}{\partial \theta_\alpha} = \mathbb{E}_\theta[\phi_\alpha]$, the KL divergence $D(\theta \,\|\, \theta^*)$ is equivalent to the difference between $\Phi(\theta^*)$ and the first-order tangent approximation $\Phi(\theta) + \nabla^T\Phi(\theta)(\theta^* - \theta)$, as illustrated

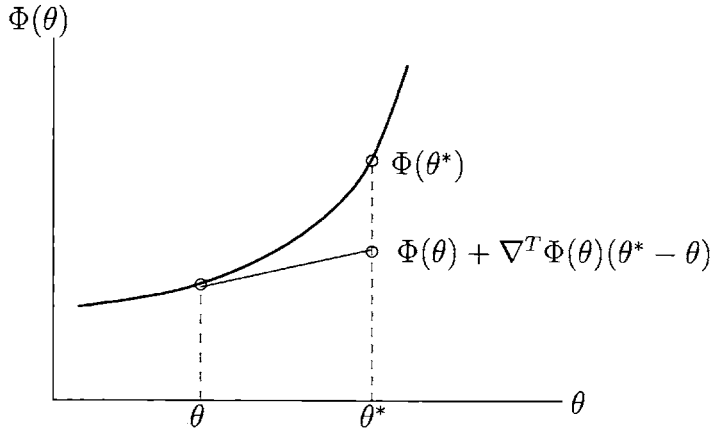


Figure 2.11. Kullback-Leibler divergence as a Bregman distance induced by the log partition function Φ. The KL divergence $D(\theta \,\|\, \theta^*)$ is equal to the difference between $\Phi(\theta^*)$ and the tangent approximation $\Phi(\theta) + \nabla^T\Phi(\theta)(\theta^* - \theta)$.

in Figure 2.11. Bregman distances are defined in precisely this manner; see Censor and Zenios [30] for more details on Bregman distances and their properties. For a minimal representation, the strict convexity of Φ guarantees that this tangent approximation is always an underestimate of $\Phi(\theta^*)$, so that the KL divergence is positive for $\theta \neq \theta^*$.[10]

It is also possible to re-write the KL divergence in terms of the dual variables η. In particular, from the Legendre duality between Φ and Ψ, we have for all dually coupled pairs (θ, η):

$$\eta^T\theta = \Phi(\theta) + \Psi(\eta)$$

Substituting this relation into equation (2.31), we obtain an alternative representation

[10]For overcomplete representations, it is possible to have distinct parameters $\theta \neq \theta^*$ that induce the same distribution, in which case $D(\theta \,\|\, \theta^*) = 0$.

of the KL divergence:

$$D(\theta \parallel \theta^*) = (\theta^*)^T[\eta^* - \eta] + \Psi(\eta) - \Psi(\eta^*) \tag{2.32}$$

Comparing equations (2.32) and (2.31), we see that the former is obtained from the latter by replacing Φ with its dual Ψ, and interchanging the roles of θ and θ^* (and their associated dual parameters η and η^*). Equation (2.32) gives rise to the notion of the dual of the KL divergence, as studied by Chentsov [32, 33].

The Kullback-Leibler divergence is very closely related to the Riemannian metric defined in equation (2.21). In particular, by Taylor series expansion of $\log p(\mathbf{x}; \theta)$, we obtain

$$D(\theta \parallel \theta^*) \approx \frac{1}{2}[\theta - \theta^*]^T G(\theta)[\theta - \theta^*] \; = \; \frac{1}{2}\|\theta - \theta^*\|^2_{G(\theta)} \tag{2.33}$$

where the approximate equality holds up to second order. In this sense, the squared distance induced by the Fisher information $G(\theta)$ is an approximation to the KL divergence. This notion will arise again in Section 2.2.8.

■ 2.2.7 I-projections onto flat manifolds

In this section, we define a pair of optimization problems canonical to information geometry. In particular, they entail projecting (where the KL divergence serves as the "distance") a given distribution onto certain types of "flat" manifolds. The dual parameterizations allow us to specify two types of flat manifold, depending on whether distributions are specified in terms of the exponential parameters θ, or the mean parameters η. This procedure of projecting onto a flat manifold, known as an *I-projection*, constitutes the basic building block for a variety of well-known optimization algorithms [e.g., 42, 134].

We begin with definitions of e and m-flat manifolds:

Definition 2.2.1. Given a linear subset of Θ, an *e-flat manifold* corresponds to its image under the coordinate mapping $\theta \mapsto p(\mathbf{x}; \theta)$. That is,

$$\mathcal{M}_e = \{\, p(\mathbf{x}; \theta) \mid A\theta = a \,\} \tag{2.34}$$

for some matrix A and vector a.

An *e-geodesic* is a 1-dimensional e-flat manifold — that is, a family of distributions specified by a line in the exponential coordinates:

$$\{\, p(\mathbf{x}; \theta(t)) \;\mid\; \theta(t) = (1-t)\,\theta_0 + t\,\theta_1, \;\; t \in \mathbb{R} \,\}$$

for some fixed θ_0 and θ_1.

With a minor abuse of notation, we shall often use $\theta \in \mathcal{M}_e$ to mean that θ belongs to the linear subset defining the e-flat manifold. To illustrate, we consider a few examples:

Example 2.2.7. For the Ising model (Example 2.2.2), an important type of e-flat manifold is induced by the linear subset $\mathcal{F}_0 = \{ \theta \mid \theta_{st} = 0 \ \forall \ (s,t) \in \mathcal{E} \}$. Any distribution in this family has the representation

$$p(\mathbf{x};\theta) = \exp\Big\{ \sum_{s=1}^{N} \theta_s x_s - \Phi(\theta) \Big\}$$

That is, these distributions are fully factorized, with no interactions between different components x_s of the random vector $\mathbf{x}$.

We specify an m-flat manifold in a similar fashion:

Definition 2.2.2. An *m-flat manifold* is the set of distributions corresponding to a linear subset of the dual variables:

$$\mathcal{M}_m = \{ p(\mathbf{x};\ \eta) \mid B\eta = b \} \tag{2.35}$$

Recall that $p(\mathbf{x};\eta)$ refers to the exponential distribution given by $p(\mathbf{x};\Lambda^{-1}(\eta))$.

We define an m-geodesic in a similar fashion to an e-geodesic: that is, as a 1-dimensional family of distributions specified by a line in the dual coordinates. Again, we shall often abuse notation by writing $\eta \in \mathcal{M}_m$ to mean that η belongs to the linear subset defining $\mathcal{M}_m$.

Note that an m-geodesic corresponds to the familiar type of mixture of distributions. That is, given the line $\eta(t) = (1-t)\eta_0 + t\eta_1$, the induced m-geodesic corresponds to mixtures of distributions in the form

$$p(\mathbf{x};\eta(t)) = (1-t)\, p(\mathbf{x};\eta_0) + t\, p(\mathbf{x};\eta_1)$$

We consider a few examples of m-flat manifolds:

Example 2.2.8. Consider a scalar Gaussian random variable with exponential representation $p(x;\theta) = \exp\{\theta_1 x + \theta_2 x^2 - \Phi(\theta)\}$. Here the dual parameter is given by $\eta_1 = \mathbb{E}_\theta[x] = \mu$, and $\eta_2 = \mathbb{E}_\theta[x^2] = \mu^2 + \sigma^2$, where μ and σ are the mean and standard deviation respectively. Thus, we see that the set of scalar Gaussian distributions with fixed mean corresponds to an m-flat manifold.

Example 2.2.9. Consider the Ising model (see Example 2.2.2). The mean parameters consist of the probabilities $\eta_s = p(x_s = 1;\theta)$ and $\eta_{st} = p(x_s = 1, x_t = 1;\theta)$. Thus, the set of all distributions with a specified set of single-node marginals

$$\{ p(\mathbf{x};\eta) \mid \eta_s = \widetilde{\eta}_s \}$$

forms an m-flat manifold.

The notions of e-flat and m-flat manifolds give rise to a pair of canonical optimization problems in information geometry. We begin by considering the problem of projecting onto an e-flat manifold.

Projection onto an e-flat manifold

For a fixed reference distribution θ^* and e-flat manifold $\mathcal{M}_e$, consider the constrained optimization problem:

$$\begin{cases} \min_\theta D(\theta^* \parallel \theta) \\ \text{s.t} \qquad \theta \in \mathcal{M}_e \end{cases} \tag{2.36}$$

From equation (2.31) and the convexity of Φ, we see that the KL divergence $D(\theta^* \parallel \theta)$ is a convex function of its second argument. Therefore, problem (2.36) is a convex optimization problem with linear constraints, so that it has a unique global optimum — say $\widehat{\theta} = \arg\min_{\theta \in \mathcal{M}_e} D(\theta^* \parallel \theta)$. Using equations (2.18a), and (2.31), we compute the gradient $\nabla_\theta D(\theta^* \parallel \theta) = \eta - \eta^*$. By the standard condition for a global minimum of a convex function over a linear manifold [20], we obtain:

$$[\eta^* - \widehat{\eta}]^T [\theta - \widehat{\theta}] = 0 \tag{2.37}$$

for all $\theta \in \mathcal{M}_e$. Equation (2.37) is the defining condition for $\widehat{\theta}$, which is known as the *I-projection* of the point θ^* onto $\mathcal{M}_e$.

Many e-flat manifolds of interest are obtained by zeroing a subset of the exponential parameters — that is:

$$\mathcal{F}_\mathcal{J} = \{\theta \mid \theta_\alpha = 0 \quad \forall\, \alpha \notin \mathcal{J}\}$$

The set of fully factorized distributions, described in Example 2.2.7, is an important case. The optimality condition of equation (2.37) has strong consequences for projections onto such manifolds. In particular, for any index $\beta \in \mathcal{J}$, we can form a perturbation $\Delta\theta = \mathbf{e}_\beta$ of all zeros except for a one in the β-entry. This perturbation $\Delta\theta$ lies in the e-flat manifold $\mathcal{F}_\mathcal{J}$, so that it must be orthogonal to $[\eta^* - \widehat{\eta}]$. Using equation (2.37), this implies that

$$\eta^*_\beta = \widehat{\eta}_\beta \qquad \forall\, \beta \in \mathcal{J} \tag{2.38}$$

That is, the dual parameters of the projection $\widehat{\eta}$ must agree with the dual parameters η^* of the original distribution for all indices β that are free to vary.

Example 2.2.10. Consider again the Ising model (Example 2.2.2), and consider the problem of projecting θ^* onto the set $\mathcal{F}_0 = \{\, \theta \mid \theta_{st} = 0 \;\; \forall \;\; (s,t) \in \mathcal{E} \,\}$ of fully factorized distributions (see Example 2.2.7). Then equation (2.38) ensures that the I-projection $\widehat{\theta}$ satisfies:

$$\mathbb{E}_{\widehat{\theta}}[x_s] = \widehat{\eta}_s = \eta^*_s = \mathbb{E}_{\theta^*}[x_s]$$

Since $\mathbf{x} \in \{0,1\}^N$ is a binary random variable, the dual variables are equivalent to node marginal probabilities. Therefore, the single node marginals of the I-projection $p(\mathbf{x}; \widehat{\theta})$ agree with those of $p(\mathbf{x}; \theta^*)$. This type of property holds for more general nested families of distributions, as described in Amari [6].

Projection onto a m-flat manifold

The problem of projecting onto an m-flat manifold $\mathcal{M}_m$ is effectively dual to the problem of projecting onto an e-flat manifold. In this case, the relevant optimization problem is

$$\begin{cases} \min_\theta D(\theta \parallel \theta^*) \\ \text{s.t} \qquad \theta \in \mathcal{M}_m \end{cases} \tag{2.39}$$

Note that in contrast to problem (2.36), here we optimize over the *first* argument of the KL divergence. This change should be understandable, in light of the relation between the KL divergence in (2.31), and its alternative formulation in equation (2.32).

On the basis of equation (2.32), it can be shown that problem (2.39) is convex and linearly constrained in the dual variables η, and so has a unique global optimum $\widehat{\eta}$. Again, straightforward computations yield the defining condition for this optimum:

$$[\theta^* - \widehat{\theta}]^T [\eta - \widehat{\eta}] = 0 \qquad \forall \eta \in \mathcal{M}_m \tag{2.40}$$

■ 2.2.8 Geometry of I-projection

Associated with equations (2.37) and (2.40) is an elegant geometric picture. Here we present this viewpoint for the I-projection onto an e-flat manifold, noting that a similar picture holds for I-projection onto an m-flat manifold. The Pythagorean results of this section (Theorems 2.2.1 and 2.2.2) date back to Kullback [118, 119]; see also Csiszár [42, 43].

To develop the geometry, note first of all that for any $\theta \in \mathcal{M}_e$, the vector $[\theta - \widehat{\theta}]$ can be viewed as the tangent vector to some e-geodesic lying within $\mathcal{M}_e$, as illustrated in Figure 2.12. Secondly, consider the m-geodesic joining the points θ^* and $\widehat{\theta}$. Although it is linear by definition in η-coordinates, it will be a curve in the θ-coordinates — namely:

$$\theta(t) = \Lambda^{-1}(\widehat{\eta} + t[\eta^* - \widehat{\eta}]) \tag{2.41}$$

This curved m-geodesic is illustrated in Figure 2.12. We calculate the tangent vector to the curve (2.41) at $\widehat{\eta}$ (i.e., at $t = 0$) as $G^{-1}(\widehat{\eta})\,[\eta^* - \widehat{\eta}]$, where we have recalled that the Jacobian of the inverse mapping Λ^{-1} is given by the inverse Fisher information G^{-1}.

Now consider the inner product, as defined by the Fisher information matrix $G(\widehat{\theta})$, between these two tangent vectors. In particular, using the fact that $G(\widehat{\theta})G^{-1}(\widehat{\eta}) = I$, we compute:

$$\langle [\theta - \widehat{\theta}],\, G^{-1}(\widehat{\eta})\,[\eta^* - \widehat{\eta}] \rangle_{G(\widehat{\theta})} \;=\; [\eta^* - \widehat{\eta}]^T [\theta - \widehat{\theta}] \tag{2.42}$$

which must vanish by equation (2.37). Therefore, the geometric consequence of equation (2.37) is that the m-geodesic joining θ^* and $\widehat{\theta}$ forms an orthogonal intersection with the e-flat manifold $\mathcal{M}_e$, as illustrated in Figure 2.12. Here orthogonality on the left side of equation (2.42) is measured using the Riemannian inner product $\langle\, ,\, \rangle_{G(\widehat{\theta})}$ induced by the Fisher information matrix (see Section 2.2.3), whereas the right side

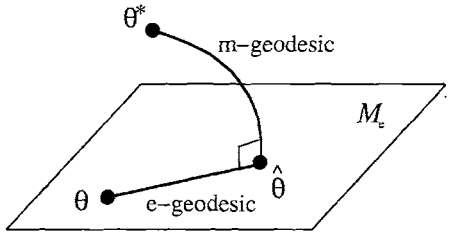


Figure 2.12. The point θ^* is projected onto the e-flat manifold $\mathcal{M}_e$ by following an m-geodesic. This yields the I-projection $\widehat{\theta}$. The tangent vector to the m-geodesic joining θ^* and $\widehat{\theta}$ is orthogonal to the manifold $\mathcal{M}_e$.

corresponds to Euclidean inner product of two vectors in $\mathbb{R}^{d(\theta)}$. Since the I-projection of θ^* onto an e-flat manifold is obtained by following an m-geodesic, it is often called an m-*projection.*

Recall from equation (2.33) that the KL divergence is closely related to the Riemannian metric induced by the Fisher information matrix. The geometric picture of I-projection allows us to further develop this relation by showing that, as with Hilbert space norms, the KL divergence satisfies a type of Pythagorean relation:

Theorem 2.2.1 (Pythagoras for m-projection). Let $\widehat{\theta}$ be the I-projection of a point θ^* onto an e-flat manifold $\mathcal{M}_e$. Then for all $\theta \in \mathcal{M}_e$, we have:

$$D(\theta^* \parallel \theta) = D(\theta^* \parallel \widehat{\theta}) + D(\widehat{\theta} \parallel \theta) \tag{2.43}$$

Proof. We provide the proof here, since it follows in a straightforward manner from our earlier development. We first use the form of the KL divergence in (2.31) to write:

$$D(\theta^* \parallel \widehat{\theta}) + D(\widehat{\theta} \parallel \theta) \quad = \quad [\eta^*]^T[\theta^* - \widehat{\theta}] + \widehat{\eta}^T[\widehat{\theta} - \theta] + \Phi(\theta^*) - \Phi(\theta)$$

We then use the optimality condition of equation (2.37) to rewrite the second term on the RHS as $[\eta^*]^T[\widehat{\theta} - \theta]$. Cancelling out the $[\eta^*]^T\widehat{\theta}$ terms then yields the result. □

Figure 2.12 again illustrates this geometry, where the points θ, $\widehat{\theta}$ and θ^* correspond to the vertices of a "right" triangle, with the segment between θ and θ^* corresponding to the hypotenuse. The "distances" between these three points are related by the Pythagorean relation[11] of equation (2.43).

We note that a geometric picture similar to that of Figure 2.12 also holds for the I-projection of θ^* (or alternatively, η^*) onto an m-flat manifold. The primary difference is that the picture holds in the dual coordinates η, rather than the exponential coordinates.

[11] In passing, we note that this Pythagorean relation holds more generally for projections onto linear sets, where the projection is defined by any Bregman distance. See [30] for further details on Bregman distances and their properties.

In this case, the projection $\widehat{\eta}$ is obtained by following a e-geodesic (curved in the η-coordinates) between η^* and $\widehat{\eta}$. For this reason, this operation is often called an *e-projection* (onto an m-flat manifold). Moreover, a Pythagorean relation analogous to that of equation (2.43) also holds:

Theorem 2.2.2 (Pythagoras for e-projection). Let $\widehat{\theta}$ be the I-projection of a point θ^* onto an m-flat manifold $\mathcal{M}_m$. Then for all $\theta \in \mathcal{M}_m$, we have:

$$D(\theta \parallel \theta^*) = D(\theta \parallel \widehat{\theta}) + D(\widehat{\theta} \parallel \theta^*)$$

Proof. The proof of this result is entirely analogous to that of Theorem 2.2.1. □

Various extensions to Theorems 2.2.1 and 2.2.2 are possible. For example, if we project onto a convex set of distributions (as opposed to a m-flat or linear manifold), then the equality of Theorem 2.2.2 is weakened to an inequality (i.e., from a Pythagorean result to the triangle inequality) [see 41].

Moreover, I-projections constitute the basic building blocks for a variety of well-known iterative algorithms. These algorithms can be divided into two broad classes: *successive projection algorithms*, and *alternating minimization algorithms*. Csiszár [43] established the convergence of the successive projection technique; in a later paper [45], he showed that the iterative scaling procedure [50] is a particular case of such an algorithm. Csiszár and Tusnády [42] present alternating minimization algorithms, and provide conditions for their convergence. The tutorial introduction by O'Sullivan [134] shows how many well-known algorithms (e.g., expectation-maximization [55], Blahut-Arimoto [9,24]) can be reformulated as particular cases of alternating minimization.

■ 2.3 Variational methods and mean field

The term *variational methods* refers to a variety of optimization problems, and associated techniques for their solution. Its origins lie in the calculus of variations [72], where the basic problem is finding the extremum of an integral involving an unknown function and its derivatives. Modern variational methods encompass a wider range of techniques, including the finite element method [157], dynamic programming [19], as well as the maximum entropy formalism [98, 99, 179]. Here we begin with a simple example to motivate the idea of a variational method; we then turn to an exposition of mean field methods. For more details, we refer the reader to the tutorial paper [92], which provides an introduction to variational methods with emphasis on their application to graphical models. The book by Rustagi [153] gives more technical details, with particular applications to problems in classical statistics.

To motivate the idea of a variational method, consider the following example, also discussed in [92]. Suppose that for a fixed vector $b \in \mathbb{R}^n$ and symmetric positive definite matrix $Q \in \mathbb{R}^{n \times n}$, we are interested in solving the linear equation $Qx = b$. Clearly, the solution is $\widehat{x} = Q^{-1}b$, which could be obtained by performing a brute force matrix inversion, and then forming a matrix-vector product. For large problems, this brute

force approach will be intractable. A variational formulation of the problem motivates a more efficient technique, and suggests natural approximations to the optimum. In particular, we consider the cost function $J(x) = \frac{1}{2}x^T Q x - b^T x$. Clearly, $J(x)$ is convex and bounded below, and so has a unique global minimum. Indeed, the minimizing argument of $J(x)$ is the desired optimum; that is, we can compute $\widehat{x}$ by minimizing $J(x)$. Moreover, to obtain approximations to the optimum, we need only perform a partial minimization of $J(x)$. The method of choice for such problems is the conjugate gradient method of numerical linear algebra [54]. It generates a sequence $\{x^k\}$, such that each x^k minimizes $J(x)$ over a $k-$dimensional subspace. Thus, the n^{th} iterate x^n will be equal (aside from possible numerical inaccuracies) to the optimum $\widehat{x}$; however, the iterations are typically terminated for some $k \ll n$, thereby yielding an approximation $x^k \approx \widehat{x}$.

■ 2.3.1 Mean field as a variational technique

We now describe particular subclass of variational methods known under the rubric of *mean field*. This term refers to a collection of techniques for obtaining approximations to distributions. While we take a variational approach to mean field, these methods can be motivated and derived from a variety of perspectives [e.g., 31, 135]. Our exposition shares the spirit of the tutorial introductions given in [92, 105]; it differs in details in that we make extensive use of exponential representation of equation (2.8).

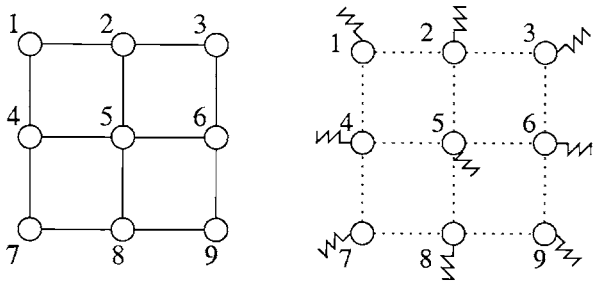
Let $p(\mathbf{x}; \theta^*)$ be the distribution of interest. We assume that this distribution is intractable, so approximating it is a natural problem. Consider the variational problem of minimizing the Kullback-Leibler divergence

$$D(\theta \parallel \theta^*) = \sum_{\alpha} \mathbb{E}_{\theta}[\phi_\alpha] [\theta - \theta^*]_\alpha + \Phi(\theta^*) - \Phi(\theta) \tag{2.44}$$

between $p(\mathbf{x}; \theta)$ and $p(\mathbf{x}; \theta^*)$. Of course, if we could perform an unconstrained minimization, we would trivially recover $\theta \equiv \theta^*$. However, since calculating the KL divergence in (2.44) entails taking expectations under $p(\mathbf{x}; \theta)$, it is necessary to restrict the minimization to a tractable family $\mathcal{F}$ of distributions. In particular, we form an approximation $p(\mathbf{x}; \widehat{\theta})$ by computing:

$$\widehat{\theta} = \arg\min_{\theta \in \mathcal{F}} D(\theta \parallel \theta^*) \tag{2.45}$$

That is, we compute the optimal approximation in some family, where optimality is measured by the KL divergence. It is important that this optimization problem does *not* correspond to an I-projection. Indeed, although we will see that $\mathcal{F}$ typically corresponds to an e-flat manifold, the optimization in equation (2.45) is over the first argument of the KL divergence, and not the second as it would be for a projection onto an e-flat manifold (see problem (2.36)). The fact that mean field is not an I-projection has important consequences, as we will discuss later.



(a) (b)

Figure 2.13. Illustration of the mean field approximation. (a) Original graph is a 3×3 grid. (b) Fully disconnected graph, corresponding to a naive mean field approximation. Wavy lines at each node represent adjustable input parameters.

The form of the KL divergence in (2.44) suggests an alternative interpretation of this optimization. By the convexity of the log partition function and equation (2.18a), we have

$$\Phi(\theta^*) \geq \Phi(\theta) + \sum_{\alpha} \mathbb{E}_\theta[\phi_\alpha]\,[\theta^* - \theta]_\alpha \tag{2.46}$$

for all θ. This lower bound also appears in the statistical physics literature, with slightly different notation, as the so-called Gibbs-Bogoliubov-Feynman inequality [see 31, 182]. As a consequence of equation (2.46), the optimization of (2.45) can be viewed as *maximizing* a lower bound on the (intractable) log partition function $\Phi(\theta^*)$. This interpretation is important in the application of mean field methods to parameter estimation via the EM algorithm [see 105].

The formulation in (2.45) encompasses a variety of mean field techniques, where a specific technique corresponds to a particular choice of e-flat manifold for the approximating family $\mathcal{F}$. For example, given the Ising model of equation (2.10), consider the family $\mathcal{F}_0 = \{\theta \mid \theta_{st} = 0 \ \forall\, (s,t) \in \mathcal{E}\}$ — that is, the e-flat manifold of fully factorized distributions (see Example 2.2.7)). Performing the minimization of equation (2.45) with this choice corresponds to finding the best fully factorized approximation. Doing so entails finding zero points of the gradient, and a particular iterative scheme for solving this fixed point equation give rise to the (naive) mean field equations.[12]

The graphical consequence of the naive mean field approximation is to decouple all nodes of the graph. Figure 2.13 illustrates this transformation: the original graph, shown in (a), is a 3×3 grid. The mean field distribution is fully factorized, and so has the

[12] In naive mean field, a fully factorized binary distribution is represented as $q(x) = \prod_s \mu_s^{x_s}(1-\mu_s)^{1-x_s}$, where the quantities $\{\mu_s\}$ are the mean field parameters. Taking gradients with respect to μ yields the usual mean field equations. Our exponential parameterization is related via $\theta_s = \log[\mu_s/(1-\mu_s)]$.

structure of the fully disconnected graph shown in (b). The mean field approximation introduces the additional variational parameters μ_s (or $\theta_s = \log[\mu_s/(1-\mu_s)]$), which can be viewed as adjustable inputs to each node. This is typical of a variational transformation: it simplifies the problem (i.e., removes edges) with the additional expense of introducing variational parameters to optimize.

The naive mean field approximation can be quite accurate in certain cases. An important example is a large and densely connected graphical model in which the pairwise couplings between variables are relatively weak. By the law of large numbers, the confluence of many effects on a given node converges to a "mean effect", so that the actual distribution is close to fully factorized. (See Jaakkola [92] for further discussion of such issues.) However, a fully factorized approximation is unable to capture multimodal behavior, and can often be a very poor approximation. For this reason, it is natural to use approximations with more structure, but that nonetheless remain tractable. Natural examples include factorized distributions formed by clustered nodes, as well as tree-structured distributions. Accordingly, different choices of $\mathcal{F}$ — corresponding to distributions with more structure than a fully factorized distribution — lead to more advanced mean field methods. For example, given a particular tree embedded within the original graph with edge set $\mathcal{E}_{\text{tree}} \subset \mathcal{E}$, we can set

$$\mathcal{F}_{\text{tree}} = \{ \theta \mid \theta_{st} = 0 \ \forall (s,t) \notin \mathcal{E}_{\text{tree}} \} \tag{2.47}$$

which corresponds to the e-flat manifold of distributions structured according to the tree. This general idea of obtaining approximations richer than a fully factorized distribution is known as *structured mean field*; such approaches were pioneered by Saul and Jordan [155], and have been investigated by a number of other researchers [e.g., 12, 74, 91, 95, 176].

■ 2.3.2 Stationarity conditions for mean field

As noted earlier, mean field optimization, as formulated in equation (2.45), does not fall within the purview of standard information geometry. In particular, although the family $\mathcal{F}$ is an e-flat manifold, the minimization does *not* take place over the second argument (which would correspond to an m-projection), but rather over the first argument. For this reason, mean field theory fails to share the geometry and optimality conditions of the m- or e-projections described in Section 2.2.7. For instance, solutions of mean field equations are not necessarily unique, and can exhibit undesirable properties such as "spontaneous symmetry breaking", in which the mean field solution is asymmetric despite complete symmetry of the actual distribution. See [92] for a simple but compelling example. Nonetheless, mean field solutions do have certain geometric properties, which we develop in this subsection for future reference.

We now derive an alternative set of stationary conditions for mean field in the general case. We first take derivatives of the KL divergence with respect to θ to obtain $\nabla_\theta D(\theta \,\|\, \theta^*) = G(\theta)\,[\theta - \theta^*]$ where $[G(\theta)]_{\alpha\beta} = \text{cov}_\theta\{\phi_\alpha, \phi_\beta\}$ is the Fisher information matrix evaluated at θ. Let $\mathcal{J}$ be a subset of the potential index set, such that the

approximating family has the form $\mathcal{F}_{\mathcal{J}} = \{ \theta \mid \theta_\alpha = 0 \quad \forall \alpha \notin \mathcal{J} \}$. Then the stationary conditions for a mean field solution $\widehat{\theta}$ are:

$$\left[G(\widehat{\theta}) \, (\widehat{\theta} - \theta^*) \right]_\alpha = \sum_\beta [G(\widehat{\theta})]_{\alpha\beta} \, [\widehat{\theta} - \theta^*]_\beta = 0 \qquad \forall \alpha \in \mathcal{J} \tag{2.48}$$

From equation (2.29), recall the role of the Fisher information matrix as the Jacobian of the mapping between θ and the dual variables $\eta_\alpha = \mathbb{E}_\theta[\phi_\alpha]$. By a Taylor series expansion, this ensures that

$$[\widehat{\eta} - \eta^*]_\alpha \approx 0 \qquad \forall \alpha \in \mathcal{J}$$

where the approximate equality holds up to first order in the perturbation $[\widehat{\theta} - \theta^*]$. That is, the mean field stationary conditions in (2.48) ensure that the dual variables $\widehat{\eta}_\alpha$ match the desired statistics η^*_α up to first order for all free indices (i.e., $\alpha \in \mathcal{J}$).

As a concrete illustration, in the case of naive mean field for an Ising model, the mean field stationarity conditions guarantee that

$$\mathbb{E}_{\widehat{\theta}}[x_s] = p(x_s = 1; \widehat{\theta}) \approx p(x_s = 1; \theta^*) = \mathbb{E}_{\theta^*}[x_s]$$

for all nodes $s \in \mathcal{V}$. That is, the single node marginals of the mean field approximation are approximately equal (up to first order) to those of the original model. To emphasize the difference with standard information geometry, recall from Example 2.2.10 that the m-projection of θ^* onto the set of fully factorized distributions $\mathcal{F}_0$ (i.e., computing $\arg\min_{\theta \in \mathcal{F}_0} D(\theta^* \parallel \theta)$) would guarantee the *equality* of these first order marginals.

Chapter 3

Perturbations and Bounds

■ 3.1 Introduction

In this chapter, we demonstrate the use of exponential representations of graph-structured distributions in application to two important problems:

(a) assessing model sensitivity to changes in parameters and structure;

(b) deriving computable bounds on quantities of interest (e.g., partition functions; marginal distributions).

The first problem is fundamental to all types of modeling; indeed, sensitivity analysis is critical in fitting and validating models. In this context, a useful tool is the *perturbation expansion*, which quantifies the deviations in model behavior as parameters are perturbed from a nominal setting. The first topic of this chapter, then, is the development of such perturbation expansions for graphical models. The second goal — that of developing bounds — is important for any graphical model in which exact inference is intractable. In particular, bounds are useful as an approximate inference tool [93, 96], for model fitting [e.g., 105], and also for large deviations analysis [e.g., 158]. Accordingly, the second part of this chapter focuses on the use of exponential representations in developing such bounds.

Although this chapter presents a number of new results, in the context of this thesis as a whole, it serves primarily as a basis for future developments. In particular, the bounds derived in this chapter will play important roles in Chapters 5, 6 and 7.

■ 3.1.1 Use of exponential representations

As we saw in Section 2.2, any exponential family is specified by a collection of functions $\boldsymbol{\phi} = \{\phi_\alpha\}$. When the exponential family represents a collection of graphical models, the ϕ_α are *potential functions* defined on cliques of the underlying graph. Specifying the collection $\boldsymbol{\phi}$, therefore, specifies the structure of the graphical model. The associated vector of exponential weights θ corresponds to the model parameters. For a given clique potential ϕ_α, the quantity θ_α represents its weight in the exponential representation. As a result, the exponential parameters can also be used to capture graph structure, since the absence or presence of any clique is controlled by whether or not the corresponding exponential parameters are zero (see Section 2.2.5). Indeed, the exponential

parameters corresponding to graphs with particular structure constraints (e.g., a bound on the maximal clique size) form e-flat manifolds in information geometry, as described Section 2.2.

Consider a particular graph-structured distribution, specified in an exponential fashion as $p(\mathbf{x}; \theta^*)$, which we shall refer to as the *target distribution*. Many quantities of interest can be represented by an expectation of the form

$$\mathbb{E}_{\theta^*}[f] = \sum_{\mathbf{x} \in \mathcal{X}^N} p(\mathbf{x}; \theta^*) f(\mathbf{x}) \tag{3.1}$$

for an appropriate function $f : \mathcal{X}^N \to \mathbb{R}$. (When $\mathbf{x}$ takes values in a continuous space, the summation of equation (3.1) should be replaced by an integral.) As an example, suppose that $\mathbf{x}$ is discrete-valued (i.e., $\mathcal{X} = \{0, \ldots, m-1\}$). If we choose f as the indicator function $\delta(x_s = j)$ for the random variable x_s to assume value $j \in \mathcal{X}$, then $\mathbb{E}_{\theta^*}[f] = p(x_s = j; \theta^*)$ is the marginal probability at node s. More generally, given a subset $S \subset \mathcal{V}$, let $\mathbf{x}_S$ denote the collection $\{x_s \mid s \in S\}$. For a configuration $\mathbf{e} \in \mathcal{X}^N$, let

$$\delta(\mathbf{x}_S = \mathbf{e}_S) \triangleq \prod_{s \in S} \delta(x_s = e_s) \tag{3.2}$$

denote an indicator function for the event that $x_s = e_s$ for all $s \in S$. Taking the expectation of such an indicator function is equivalent to computing the value of a marginal distribution over the nodes in S. On the other hand, as an example for a continuous-valued process, the conditional mean of the variable at node s corresponds to setting $f(\mathbf{x}) = x_s$.

Given a target distribution $p(\mathbf{x}; \theta^*)$, we develop expansions for the expectations $\mathbb{E}_{\theta^*}[f]$ and $\log \mathbb{E}_{\theta^*}[f]$ in terms of quantities computed using a related distribution $p(\mathbf{x}; \theta)$. At a conceptual level, the coefficients of these expansions provide valuable information on the sensitivity to specified perturbations. On the practical side, in the case where $p(\mathbf{x}; \theta^*)$ is intractable whereas $p(\mathbf{x}; \theta)$ is tractable, such expansions may be computationally useful, in that they provide a succession of approximations to $\mathbb{E}_{\theta^*}[f]$. In Chapter 4, we shall develop an exact inference algorithm for Gaussian processes based on such an idea.

The basic thrust in our development of bounds is similar to the perturbation expansions; in detail, however, the analysis is of a different flavor, since we require quantities that are explicit bounds and not just approximations. To develop bounds on the expectation $\mathbb{E}_{\theta^*}[f]$, we make use of results from convex analysis, applied to the log partition function of a suitably modified model. We first develop a set of bounds based on a *single* approximating distribution $p(\mathbf{x}; \theta)$; these bounds represent an extension of the mean field bounds described in Section 2.3.1. Indeed, for the special case $f(\mathbf{x}) \equiv \mathbf{1}$, our results reduce to the usual mean field bound on the log partition function (see equation (2.46)). It is not surprising, then, that the stationary conditions for the exponential parameter(s) optimizing these bounds are similar to the mean field conditions [e.g., 92, 105].

In the context of an exponential representation, it is natural to consider the idea of taking convex combinations of exponential parameters. The convexity of the log partition function then allows us to apply Jensen's inequality [e.g., 41], which leads to a new set of bounds. These bounds, in contrast to the first set, are based on *multiple* approximating points $\{p(\mathbf{x};\theta^i)\}$. We will return to these bounds in Chapter 7, where we consider the problem of optimizing both the weights in the convex combination as well as the choice of exponential parameters on spanning trees of the graph.

We then consider the problem of strengthening the bounds. In order to tighten both sets of bounds on $\mathbb{E}_{\theta^*}[f]$, we exploit the idea of an additive decomposition of the form $f = \sum_k f^k$. Such decompositions lead to a family of bounds, nested in terms of the fineness of the decomposition of f. Although refining the additive decomposition increases the cost of computing the bounds, we prove that refinements are, in general, *guaranteed* to yield stronger bounds.

The remainder of this chapter is organized as follows. In Section 3.2, we present perturbation expansions for $\mathbb{E}_{\theta^*}[f]$ and $\log \mathbb{E}_{\theta^*}[f]$, and illustrate their interpretation with some simple examples. In Section 3.3, we derive two sets of bounds on these same expectations, either based on a single approximating point, or multiple approximating points. Section 3.4 then is devoted to the development of techniques for strengthening the basic form of these bounds. In Section 3.5, we illustrate our results in application to bounding the log partition function of some simple graphs. We conclude in Section 3.6 with a summary, and discussion of role of these results in the remainder of the thesis.

■ 3.2 Perturbations and sensitivity analysis

Given the target distribution $p(\mathbf{x};\theta^*)$, consider the expectation $\mathbb{E}_{\theta^*}[f]$ of a function $f : \mathcal{X}^N \to \mathbb{R}$. In this section, we derive perturbation expansions for this expectation (as well as for $\log \mathbb{E}_{\theta^*}[f]$) about an approximating distribution $p(\mathbf{x};\theta)$. Coefficients of these expansions are given by cumulants computed under the approximating distribution.

Related results have been derived by other researchers [e.g., 11,29,51,115,120]. For example, Laskey [120] showed how to perform sensitivity analysis of a directed tractable Bayesian network by taking first derivatives with respect to model parameters. Darwiche [51], using a representation that is closely related to an overcomplete exponential parameterization (see Example 2.2.4), developed a differential approach that gives an alternative perspective on exact inference in tractable models. Perhaps most closely related to our work are the results of Barber and van der Laar [11], who developed perturbation expansions of the log partition function about a tractable distribution, and also considered methods, akin to mean field, for attempting to optimize such expansions. These results are basically equivalent to our expansions of $\mathbb{E}_{\theta^*}[f]$ when $f(\mathbf{x}) = 1$.

■ 3.2.1 Expansions of the expectation $\mathbb{E}_{\theta^*}[f]$

The starting point of our development is the fact, as pointed out in Section 2.2.2, that the log partition function $\Phi(\theta)$ is very closely related to the cumulant generating

function [80] of $p(\mathbf{x};\theta)$, or more precisely to the cumulant generating function of the random variables $\{\phi_\alpha(\mathbf{x})\}$ under this distribution. In particular, the first and second-order cumulants of these variables are given by, respectively:

$$\mathbb{E}_\theta[\phi_\alpha] = \frac{\partial \Phi(\theta)}{\partial \theta_\alpha} \tag{3.3a}$$

$$\mathrm{cov}_\theta\{\phi_\alpha, \phi_\beta\} = \frac{\partial^2 \Phi(\theta)}{\partial \theta_\alpha \, \partial \theta_\beta} \tag{3.3b}$$

Higher-order cumulants are specified by recursive differentiation. For example, the third-order cumulant $\mathrm{cum}_\theta\{\phi_\alpha, \phi_\beta, \phi_\sigma\}$ is given by $\frac{\partial}{\partial \theta_\sigma} \mathrm{cov}_\theta\{\phi_\alpha, \phi_\beta\}$, which can be evaluated as:

$$\begin{aligned}\mathrm{cum}_\theta\{\phi_\alpha, \phi_\beta, \phi_\sigma\} = \mathbb{E}_\theta[\phi_\alpha \, \phi_\beta \, \phi_\sigma] - \mathbb{E}_\theta[\phi_\alpha] \, \mathbb{E}_\theta[\phi_\beta \, \phi_\sigma] - \mathbb{E}_\theta[\phi_\beta] \, \mathbb{E}_\theta[\phi_\alpha \, \phi_\sigma] \\ - \mathbb{E}_\theta[\phi_\sigma] \, \mathbb{E}_\theta[\phi_\alpha \, \phi_\beta] + 2\mathbb{E}_\theta[\phi_\alpha]\mathbb{E}_\theta[\phi_\beta]\mathbb{E}_\theta[\phi_\sigma]\end{aligned} \tag{3.4}$$

Now for an arbitrary function $f : \mathcal{X}^N \to \mathbb{R}$, it is also possible to consider the expectation $\mathbb{E}_{\theta^*}[f]$ under $p(\mathbf{x};\theta^*)$ as a type of first-order cumulant. Consequently, it is straightforward to apply Taylor's theorem [161] in order to expand it about $p(\mathbf{x};\theta)$ in terms of higher-order cumulants. We summarize the result as follows:

Proposition 3.2.1. Let $\epsilon = \theta^* - \theta$ be the difference between two arbitrary parameter vectors, and let $f : \mathcal{X}^N \to \mathbb{R}$ be arbitrary. Then we have:

$$\mathbb{E}_{\theta^*}[f] = \mathbb{E}_\theta[f] + \sum_\alpha \mathrm{cov}_\theta\{f, \phi_\alpha\} \, \epsilon_\alpha + \frac{1}{2} \sum_{\alpha,\beta} \mathrm{cum}_\theta\{f, \phi_\alpha, \phi_\beta\} \epsilon_\alpha \epsilon_\beta + \mathcal{O}(\|\epsilon\|^3) \tag{3.5}$$

Remark: Although equation (3.5) gives terms only up to second order, it should be clear that we can continue such expansions to arbitrary order.

The first-order coefficient corresponding to the perturbation element ϵ_α is the covariance

$$\mathrm{cov}_\theta\{f, \phi_\alpha\} = \mathbb{E}_\theta[f \, \phi_\alpha] - \mathbb{E}_\theta[f]\mathbb{E}_\theta[\phi_\alpha] \tag{3.6}$$

It has a sensitivity interpretation as a (first-order) measure of the effect of perturbations in the strength of clique potential ϕ_α on the expectation $\mathbb{E}_{\theta^*}[f]$. If $f(\mathbf{x}) = \phi_\sigma(\mathbf{x})$ for some σ, then this covariance of equation (3.6) corresponds to the element $g_{\alpha\sigma}$ of the Fisher information matrix (see Section 2.2.3), in which case this sensitivity interpretation is well-known.

Suppose that the approximating distribution $p(\mathbf{x};\theta)$ is tractable, in which case $\mathbb{E}_\theta[f]$ can be computed and viewed as a zero$^{\text{th}}$ order approximation to $\mathbb{E}_{\theta^*}[f]$. Adding in the covariance terms gives rise to a first-order approximation, but is the computational cost of doing so prohibitive? This cost depends on the nature of the function f. By definition, the clique potential ϕ_α depends only on a limited subvector $\mathbf{x}_\alpha$ of the full vector

$\mathbf{x}$. If in addition the function f depends only a local and small subvector — say $\mathbf{x}_f$ — then these covariance terms will involve interactions only among relatively small subsets of variables, so that computation will be tractable. Natural choices of f for which this local support assumption holds are the indicator functions $f(\mathbf{x}) = \delta(x_s = j)$ and $f(\mathbf{x}) = \delta(x_s = j)\delta(x_t = k)$. In such cases, as long as the nominal distribution $p(\mathbf{x}; \theta)$ is tractable, computing these sensitivity coefficients will be computationally feasible. As an example, for an N-node graph with pairwise cliques and a tree-structured approximating distribution $p(\mathbf{x}; \theta)$, computing the sensitivity coefficients associated with $f(\mathbf{x}) = \delta(x_s = j)$ for a discrete-valued process assuming m states would entail a cost of at most $\mathcal{O}(m^4 N)$.

Example 3.2.1. To illustrate the sensitivity interpretation of Proposition 3.2.1, consider the choice $f(\mathbf{x}) = \delta(x_s = j)$ (so that the expectation $\mathbb{E}_{\theta^*}[f]$ is equivalent to the marginal probability $p(x_s = j; \theta^*)$ at node s). If the clique potential ϕ_α is a function only of the random variables at a subset of nodes sufficiently "far away" from node s, then the random variables $\phi_\alpha(\mathbf{x})$ and $f(\mathbf{x})$ should be approximately independent under $p(\mathbf{x}; \theta)$, in which case

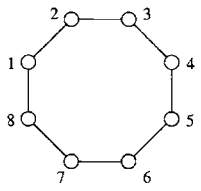
$$\mathrm{cov}_\theta\{f, \phi_\alpha\} = \mathbb{E}_\theta[f\ \phi_\alpha] - \mathbb{E}_\theta[f]\mathbb{E}_\theta[\phi_\alpha] \approx 0$$

That is, perturbations in the clique potential ϕ_α should have little effect on the expectation.

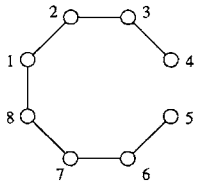
Figure 3.1 illustrates this effect for the single cycle in (a), and the tree in (b) obtained by removing the single edge $(4, 5)$. We formed a distribution $p(\mathbf{x}; \theta^*)$ over a binary-valued vector $\mathbf{x}$ on the single cycle in (a), using a set of relatively homogeneous set of attractive potentials (i.e., that encourage neighboring nodes to take the same value). The vector θ corresponds to θ^*, with the element corresponding to edge $(4, 5)$ set to zero. Panel (c) plots the error $\{\mathbb{E}_{\theta^*}[f] - \mathbb{E}_\theta[f]\}$ versus node number. Notice how the error is largest at nodes 4 and 5 (adjacent to the cut edge), and decays for distant nodes (e.g., 1 and 8).

Continuing the expansion of Proposition 3.2.1 to higher order provides, in principle, a sequence of approximations to $\mathbb{E}_{\theta^*}[f]$. (As noted earlier, the nominal expectation $\mathbb{E}_\theta[f]$ represents a zero$^{\text{th}}$-order approximation, whereas adding in the covariance terms would yield a first-order approximation.) One would expect that the approximation should improve as higher order terms are incorporated; however, such monotonicity is not guaranteed. Moreover, for a discrete process, the computation of higher order terms becomes progressively more costly, even in the case where f depends only on a local subvector and $p(\mathbf{x}; \theta)$ is a tractable distribution. In general, the k^{th}-order coefficient will require computing a term of the form $\mathbb{E}_\theta[f \prod_{i=1}^{k-1} \phi_{\alpha_i}]$. For a discrete-valued process, this will be an intractable computation for sufficiently large k.

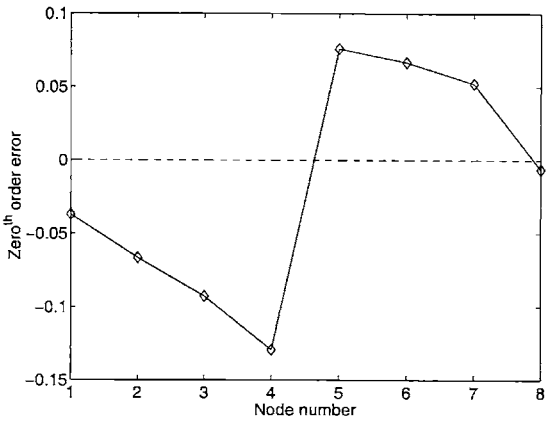
For a Gaussian process, it turns out that the necessary higher-order terms can be computed recursively in terms of lower order quantities. This leads to one derivation of an algorithm for exact inference of Gaussian processes, which we will explore in Chapter 4.



(a)



(b)



(c)

Figure 3.1. Panel (c) shows the error $\{\mathbb{E}_{\theta^*}[f] - \mathbb{E}_{\theta}[f]\}$ between actual marginals $\mathbb{E}_{\theta^*}[f]$ on a single cycle (a), and the zero$^{\text{th}}$-order approximations $\mathbb{E}_{\theta}[f]$ from a tree obtained by removing edge $(4, 5)$ (b). Note how the error is maximal around nodes 4 and 5, and decays as the distance between the node and the cut edge increases.

■ 3.2.2 Expansions for $\log \mathbb{E}_{\theta^*}[f]$

We now consider perturbation expansions of the quantity $\log \mathbb{E}_{\theta^*}[f]$. This expansion has an interesting form, and different properties than that of Proposition 3.2.1. It is based on a representation of $\log \mathbb{E}_{\theta^*}[f]$ as a difference between the original log partition function $\Phi(\theta^*)$, and a second log partition function that is suitably modified (in a manner to be described).

For subsequent developments, we need to ensure that $\mathbb{E}_{\theta^*}[f] > 0$ so that taking logarithms is well-defined. In the case of a strictly positive distribution (i.e., $p(\mathbf{x}; \theta^*) > 0$ for all $\mathbf{x} \in \mathcal{X}^N$), this condition is ensured by the following:

Assumption 3.2.1. The function f takes only non-negative values (i.e., $f(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathcal{X}^N$) and f is not identically zero (i.e., $f(\mathbf{x}) > 0$ for at least one $\mathbf{x} \in \mathcal{X}^N$).

For developments in the sequel, it is helpful to introduce now the notion of a *tilted distribution*. This concept is central in both importance sampling [150], and large deviations theory [e.g., 53, 158]. Suppose that we are given a function f satisfying Assumption 3.2.1, as well as a distribution in exponential form:

$$p(\mathbf{x}; \theta) = \exp\Big\{ \sum_{\alpha} \theta_\alpha \phi_\alpha(\mathbf{x}) - \Phi(\theta) \Big\}$$

We then form a new distribution, denoted $p(\mathbf{x};\theta_f)$, by "tilting" the original distribution with the function f. To be precise:

$$p(\mathbf{x};\theta_f) \quad \propto \quad \exp\Big\{ \sum_\alpha \theta_\alpha \phi_\alpha(\mathbf{x}) \Big\} \, f(\mathbf{x}) \tag{3.7}$$

We denote by $\Phi_f(\theta)$ the log partition function associated with this representation:

$$\Phi_f(\theta) \quad \triangleq \quad \log \Big[\sum_{\mathbf{x} \in \mathcal{X}^N} \exp\Big\{ \sum_\alpha \theta_\alpha \phi_\alpha(\mathbf{x}) \Big\} \, f(\mathbf{x}) \Big] \tag{3.8}$$

The function Φ_f has important property: by a simple re-arrangement of equation (3.8), we obtain the relation

$$\log \mathbb{E}_\theta[f] = \Phi_f(\theta) - \Phi(\theta) \tag{3.9}$$

This equation is a reformulation, in terms of exponential parameters and log partition functions, of a relation used in statistical physics for expressing moments as ratios of partition functions [e.g., 15].

Equation (3.9) will play a fundamental role in our development of bounds in Section 3.3. For now, we observe that the derivatives of $\Phi(\theta)$ (respectively $\Phi_f(\theta)$) correspond to cumulants of the random variables $\{\phi_\alpha(\mathbf{x})\}$ under the distribution $p(\mathbf{x};\theta)$ (respectively $p(\mathbf{x};\theta_f)$). On this basis, it is straightforward, again by a Taylor series approach [161], to obtain a perturbation expansion for $\log \mathbb{E}_{\theta^*}[f]$.

Proposition 3.2.2. Let $\epsilon = \theta^* - \theta$ be the difference between two arbitrary parameter vectors, and consider a function $f : \mathcal{X}^N \to [0, \infty)$ satisfying Assumption 3.2.1. Then we have the expansion:

$$\log \mathbb{E}_{\theta^*}[f] = \log \mathbb{E}_\theta[f] + \sum_\alpha \big\{ \mathbb{E}_{\theta_f}[\phi_\alpha] - \mathbb{E}_\theta[\phi_\alpha] \big\} \, \epsilon_\alpha + \frac{1}{2} \epsilon^T \big\{ G(\theta_f) - G(\theta) \big\} \epsilon \; + \; \mathcal{O}(\|\epsilon\|^3) \tag{3.10}$$

Here $\mathbb{E}_{\theta_f}[\phi_\alpha]$ denotes the expectation of $\phi_\alpha(\mathbf{x})$ under $p(\mathbf{x};\theta_f)$; and $G(\theta_f)$ and $G(\theta)$ are the Fisher information matrices corresponding to $p(\mathbf{x};\theta_f)$ and $p(\mathbf{x};\theta)$ respectively. (Explicitly, we have $[G(\theta)]_{\alpha\beta} = \mathrm{cov}_\theta\{\phi_\alpha, \phi_\beta\}$).

It is helpful to interpret Proposition 3.2.2 for particular choices of the function f. Given some subset $S \subseteq \mathcal{V}$, suppose, in particular, that f is an indicator function $\delta(\mathbf{x}_S = \mathbf{e}_S)$, as defined in equation (3.2), for $\mathbf{x}_S$ to assume the configuration $\mathbf{e}_S$. In this case, the distribution $p(\mathbf{x};\theta_f)$ is equivalent to $p(\mathbf{x};\theta)$ but with the variables $\mathbf{x}_S$ *fixed* to the values in $\mathbf{e}_S$. Thus, the first-order term $\{\mathbb{E}_{\theta_f}[\phi_\alpha] - \mathbb{E}_\theta[\phi_\alpha]\}$ corresponds to difference between the mean of $\phi_\alpha(\mathbf{x})$ under a clamped distribution, and its mean under the original distribution $p(\mathbf{x};\theta)$. Similarly, the second order term is the difference between the two respective Fisher information matrices. The factor controlling the accuracy of the expansion is how much cumulants of $\{\phi_\alpha(\mathbf{x})\}$ under the distribution $p(\mathbf{x};\theta)$ are affected by conditioning on the subset of variables $\mathbf{x}_S$.

■ 3.3 Bounds on expectations

The goal of this section is more ambitious; rather than approximating the expectation $\mathbb{E}_{\theta^*}[f]$, we seek to generate upper and lower bounds. Our analysis makes use of standard tools from convex analysis applied to log partition functions.

Central in our development is the representation of $\log \mathbb{E}_{\theta^*}[f]$ as the difference between two log partition functions, as given in equation (3.9). We established in Lemma 2.2.1 of Chapter 2 that Φ is convex as a function of θ, and strictly so for a minimal exponential representation (see Section 2.2.1). A similar argument establishes that Φ_f, as the log partition function of a tilted distribution, is also convex.

The convexity of these log partition functions allows us to exploit standard properties of convex functions to derive bounds on $\log \mathbb{E}_{\theta^*}[f]$. We use, in particular, the following two properties [see, e.g., 20] of any differentiable convex function F. First of all, for any two points $\mathbf{y}, \mathbf{z}$, the (first-order) tangent approximation to $F(\mathbf{y})$ based on $\mathbf{z}$ is an underestimate:

$$F(\mathbf{y}) \geq F(\mathbf{z}) + \nabla^T F(\mathbf{z}) \, (\mathbf{y} - \mathbf{z}) \tag{3.11}$$

Secondly, for any collection of points $\{\mathbf{y}^i\}$ and set of weights $\mu^i \geq 0$ such that $\sum_i \mu^i = 1$, we have Jensen's inequality [41]:

$$F(\sum_i \mu^i \mathbf{y}^i) \leq \sum_i \mu^i F(\mathbf{y}^i) \tag{3.12}$$

The analysis in this section will be performed under a slightly stronger version of Assumption 3.2.1:

Assumption 3.3.1. The function $f : \mathcal{X}^N \to [0, 1]$ and $f(\mathbf{x}) > 0$ for at least some $\mathbf{x} \in \mathcal{X}^N$.

For a discrete-valued process, this assumption entails no loss of generality, since we can define $m = \min_{\mathbf{x}} f(\mathbf{x})$ and $M = \max_{\mathbf{x}}[f(\mathbf{x}) - m]$, and then form the new function $\widetilde{f}(\mathbf{x}) = \frac{1}{M+1}[f(\mathbf{x}) - m]$ which satisfies Assumption 3.3.1. By the linearity of expectation, bounds for $\mathbb{E}_{\theta^*}[\widetilde{f}]$ can immediately be translated to bounds for $\mathbb{E}_{\theta^*}[f]$.

■ 3.3.1 Relation to previous work

The first set of bounds that we present are very closely related the standard mean field lower bound [e.g., 92, 105] on the log partition function. As described in Section 2.3.1, both naive and structured mean field are extensively studied and used [e.g., 12, 74, 123, 155, 176]. Instead of bounding the original log partition function $\Phi(\theta^*)$, as in ordinary mean field, we bound the tilted partition function $\Phi_f(\theta^*)$. This procedure leads to a bound on the expectation $\mathbb{E}_{\theta^*}[f]$ for an arbitrary function f. This bound has an interesting form, and reduces to the ordinary mean field bound when $f(\mathbf{x}) = 1$ for all $\mathbf{x} \in \mathcal{X}^N$. Accordingly, we show that the stationary conditions for the tightest form of

this bound are very similar to the corresponding mean field stationary conditions. The additional flexibility of allowing f to be arbitrary pays dividends in Section 3.4.1, in which we present a simple method for tightening any mean field bound.

Based on a review of the literature, it appears that upper bounds are more difficult to obtain. There are only a limited number of upper bounds, and their domain of applicability is limited. For example, using a variational upper bound on the function $\log[\exp(u) + \exp(-u)]$, Jaakkola and Jordan [94] derived a recursive procedure for obtaining quadratic upper bounds on the log partition function for the Boltzmann machine (i.e., a binary process with pairwise maximal cliques). For relatively weak interactions, these upper bounds are much stronger than the standard (linear) mean field lower bound. However, generalizing this procedure to discrete processes with more than two states is not straightforward. For the class of log concave models (a particular subclass of directed networks), Jaakkola and Jordan [93] developed upper bounds on marginal probabilities using other bounds from convex analysis.

In Section 3.3.3, we derive a new set of upper bounds on the expectation $\log \mathbb{E}_{\theta^*}[f]$ that are applicable to an arbitrary undirected graphical model. These upper bounds generalize an idea used by Jaakkola and Jordan [96] to obtain bounds for the QMR-DT network. We also show in Section 3.4.2 that the idea of additive decompositions can also be used to strengthen these bounds, again with an adjustable price in computation.

■ 3.3.2 Basic bounds based on a single approximating point

By applying the property in equation (3.11) to the tilted partition function Φ_f, it is straightforward to derive the following bound:

Proposition 3.3.1. Let $f : \mathcal{X}^N \to \mathbb{R}$ satisfy Assumption 3.3.1, and consider distributions at parameter values θ^* and θ. Then the expectation $\mathbb{E}_{\theta^*}[f]$ is bounded below and above as follows:

$$\mathbb{E}_{\theta^*}[f] \geq \mathbb{E}_{\theta}[f] \exp\left\{ - D(\theta \parallel \theta^*) + \frac{1}{\mathbb{E}_{\theta}[f]} \sum_{\alpha} \operatorname{cov}_{\theta}\{f, \phi_\alpha\}(\theta^* - \theta)_\alpha \right\} \tag{3.13a}$$

$$\mathbb{E}_{\theta^*}[f] \leq 1 - (1 - \mathbb{E}_{\theta}[f]) \exp\left\{ - D(\theta \parallel \theta^*) + \frac{1}{1 - \mathbb{E}_{\theta}[f]} \sum_{\alpha} \operatorname{cov}_{\theta}\{f, \phi_\alpha\}(\theta^* - \theta)_\alpha \right\} \tag{3.13b}$$

Here $\operatorname{cov}_{\theta}\{f, \phi_\alpha\} = \mathbb{E}_{\theta}[f\,\phi_\alpha] - \mathbb{E}_{\theta}[f]\mathbb{E}_{\theta}[\phi_\alpha]$ is the covariance between f and ϕ_α, and $D(\theta \parallel \theta^*)$ is the Kullback-Leibler divergence between $p(\mathbf{x}; \theta)$ and $p(\mathbf{x}; \theta^*)$.

Proof. See Appendix B.1. □

Since the function f satisfies Assumption 3.3.1, the expectation $\mathbb{E}_{\theta^*}[f]$ necessarily lies in $[0, 1]$. A desirable feature of both the lower and upper bounds of Proposition 3.3.1 is that they respect this interval requirement. Indeed, it can be seen that the RHS of

equation (3.13a) is always non-negative, and similarly, the RHS of equation (3.13b) is always less than or equal to one. Thus, the corresponding bounds are never vacuous.[1]

As noted above, a caveat associated with Proposition 3.3.1 is that the bounds, in the form given, contain the intractable log partition function $\Phi(\theta^*)$. (In particular, it appears as part of the KL divergence $D(\theta \,\|\, \theta^*)$ term). For a undirected graphical model, evaluating this partition function is, in general, as difficult as computing the original expectation. In order to evaluate these bounds, we require a computable *upper* bound on the log partition function. The methods presented in Section 3.3.3 provide precisely such bounds.

It is interesting to consider the bounds of Proposition 3.3.1 when we choose θ equal to an optimum mean field point (see Section 2.3.2). In particular, fix some substructure of the graph $\mathcal{G}$ — say, for concreteness, an embedded spanning tree — that is represented by the e-flat manifold $\mathcal{F}_{\text{tree}}$. (See equation (2.47) of Chapter 2). Now suppose that we perform structured mean field optimization; that is, we compute

$$\widehat{\theta} = \arg \min_{\theta \in \mathcal{F}_{\text{tree}}} D(\theta \,\|\, \theta^*) \tag{3.14}$$

The elements θ_β over which we optimize are those corresponding to any single node potential function, or any edge belonging to the tree. We obtain stationary conditions by setting to zero the gradient of $D(\theta \,\|\, \theta^*)$ with respect to each such element. From our analysis in Section 2.3.2, these stationary conditions are given by

$$\sum_\alpha g_{\alpha\beta}(\widehat{\theta}) \, [\theta^* - \widehat{\theta}]_\alpha = 0 \tag{3.15}$$

for all *free indices* β: i.e., those indices corresponding to elements θ_β that are free to vary in the variational problem (3.14). Here $g_{\alpha\beta}(\widehat{\theta}) = \text{cov}_{\widehat{\theta}}\{\phi_\alpha, \phi_\beta\}$ is an element of the Fisher information matrix evaluated at $p(\mathbf{x}; \widehat{\theta})$.

Suppose that the function f corresponds to a potential function ϕ_β for some free index β. For the tree example used here, such functions include the indicator function $f(\mathbf{x}) = \delta(x_s = a)$ for any node s and state $a \in \mathcal{X}$, as well as $f(\mathbf{x}) = \delta(x_s = a)\delta(x_t = b)$ for any edge (s,t) in the tree, and pair of states (a,b). For such choices of f, it can be seen that the summation $\sum_\alpha \text{cov}_{\widehat{\theta}}\{f, \phi_\alpha\}(\theta^* - \widehat{\theta})_\alpha$ in equation (3.13a) vanishes, so that the bound reduces to the much simpler form:

$$\mathbb{E}_{\theta^*}[f] \;\geq\; \mathbb{E}_{\widehat{\theta}}[f] \, \exp\big\{ - D(\widehat{\theta} \,\|\, \theta^*)\big\} \tag{3.16}$$

A similar simplification applies to equation (3.13b).

Optimizing single-point bounds

Suppose that we are allowed to choose the approximating point θ from some class of distributions (e.g., the e-flat manifold $\mathcal{F}_{\text{tree}}$ formed by a spanning tree, as above). It is

[1] Other types of bounds (e.g., the union bound) can give meaningless assertions (e.g., a probability is less than 3).

tempting to believe that equation (3.16), due to its attractive simplicity, corresponds to the optimized form of the bound (3.13a). A bit of reflection establishes that this is *not* the case; note that equation (3.16) does not take into account the particulars of f, as it would if it were optimized for f.

In order to optimize the bound of equation (3.13a), we need to return to its derivation. Recall that it is based on lower-bounding the tilted partition function $\Phi_f(\theta)$ by the first-order tangent approximation in equation (3.11). To optimize the bound, we want to make this tangent-approximation as tight as possible. This problem is equivalent to the mean field optimization problem, albeit applied to a tilted log partition function.

With this insight, it is straightforward to derive stationary conditions for a zero-gradient point of this optimization problem. We simply take derivatives of the logarithm of the RHS of equation (3.13a) with respect to parameters θ_β that are free to vary, and obtain the following necessary conditions for an optimum:

$$\sum_\alpha \left\{ \frac{\mathbb{E}_\theta[f\ \phi_\alpha\ \phi_\beta]}{\mathbb{E}_\theta[f]} - \frac{\mathbb{E}_\theta[f\ \phi_\alpha]}{\mathbb{E}_\theta[f]} \frac{\mathbb{E}_\theta[f\ \phi_\beta]}{\mathbb{E}_\theta[f]} \right\} \left[\theta - \theta^*\right]_\alpha = 0 \qquad \forall \text{ free indices } \beta \quad (3.17)$$

The term with curly braces in equation (3.17) can be recognized as an element $g_{\alpha\beta}(\theta_f)$ of the Fisher information matrix corresponding to the tilted distribution $p(\mathbf{x};\theta_f)$ defined in equation (3.7). Note the correspondence with the stationary conditions for ordinary mean field (see equation (2.48) of Section 2.3.2). This correspondence is not surprising, however, since the optimization problem is equivalent to mean field with the tilted log partition function. Thus, the set of gradient equations in (3.17) can be solved with the usual mean field updates [105], or other methods from nonlinear programming [e.g., 20].

■ 3.3.3 Bounds based on multiple approximating distributions

The bounds of Proposition 3.3.1 are based on a single (tractable) approximating distribution $p(\mathbf{x};\theta)$. In this section, we derive a new set of bounds, complementary to those of Proposition 3.3.1 in the sense that they are based on multiple approximating distributions. As a concrete example, suppose that the set of distributions that can be used to form approximations are those that are tree-structured. Then the bounds of Proposition 3.3.1 are based on using only a single tree. Since any graph with cycles has a large number of embedded trees, it is natural to consider bounds based on multiple trees.

We begin by letting $\boldsymbol{\theta} = \{\theta^i \mid i \in \mathcal{I}\}$ denote a collection of exponential parameters corresponding to a set of approximating distributions $\{\ p(\mathbf{x};\theta^i) \mid i \in \mathcal{I}\ \}$. We are interested in weighted combinations of these points, so that we define a vector of weights

$$\vec{\mu} \triangleq \{\ \mu^i,\ i \in \mathcal{I} \mid \mu^i \geq 0;\ \sum_i \mu^i = 1\ \} \quad (3.18)$$

The vector $\vec{\mu}$ can be viewed as a probability distribution over the set of approximating distributions.

We use these weights and approximating points to generate convex combinations of exponential parameters, which are defined as follows.

Definition 3.3.1. Given such a distribution $\vec{\mu}$ and a collection of exponential vectors $\boldsymbol{\theta}$, a *convex combination* of exponential parameter vectors is defined via the expectation:

$$\mathbb{E}_{\vec{\mu}}[\boldsymbol{\theta}] \equiv \mathbb{E}_{\vec{\mu}}[\theta^i] \triangleq \sum_{i \in \mathcal{I}} \mu^i \theta^i \tag{3.19}$$

Now recall that θ^* is the exponential parameter vector of a distribution $p(\mathbf{x}; \theta^*)$ defined by the original graph $\mathcal{G}$. We are interested in sets of approximating points $\boldsymbol{\theta}$ for which there exists a convex combination that is equal to θ^*. Accordingly, we define the following set of pairs $(\boldsymbol{\theta}; \vec{\mu})$:

$$\mathcal{A}(\theta^*) \triangleq \left\{ (\boldsymbol{\theta}; \vec{\mu}) \mid \mathbb{E}_{\vec{\mu}}[\boldsymbol{\theta}] = \theta^* \right\} \tag{3.20}$$

That is, $\mathcal{A}(\theta^*)$ is the set of all pairs $(\boldsymbol{\theta}; \vec{\mu})$ of exponential parameters $\boldsymbol{\theta} = \{ \theta^i \mid i \in \mathcal{I} \}$ and distributions $\vec{\mu}$ for which the convex combination $\mathbb{E}_{\vec{\mu}}[\boldsymbol{\theta}]$ is equal to the target parameter θ^*.

Note: The expectation notation will be used more generally in the following way: given some function F and the function values $F(\theta^i)$ for all $i \in \mathcal{I}$, we define

$$\mathbb{E}_{\vec{\mu}}[F(\boldsymbol{\theta})] = \mathbb{E}_{\vec{\mu}}[F(\theta^i)] = \sum_{i \in \mathcal{I}} \mu^i F(\theta^i)$$

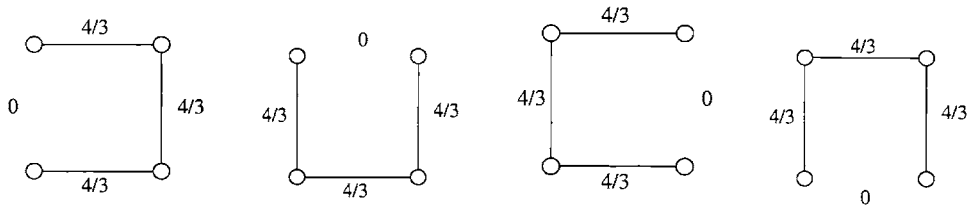
Example 3.3.1. To illustrate these concepts, consider a binary distribution defined by a single cycle on 4 nodes, as shown in Figure 3.2. Consider a target distribution of the form

$$p(\mathbf{x}; \theta^*) = \exp\{x_1 x_2 + x_2 x_3 + x_3 x_4 + x_4 x_1 - \Phi(\theta^*)\}$$

That is, the target distribution is specified by the minimal parameter $\theta^* = [0\ 0\ 0\ 0\ 1\ 1\ 1\ 1]$, where the zeros represent the fact that $\theta^*_s = 0$ for all $s \in \mathcal{V}$. We consider the four spanning trees associated with the single cycle on 4 nodes, and define a corresponding set of four exponential parameter vectors $\boldsymbol{\theta} = \{\theta^i \mid i = 1, 2, 3, 4\}$ as follows:

$$\begin{aligned}
\theta^1 &= (4/3)\ [0\ 0\ 0\ 0\ \ 1\ 1\ 1\ 0] \\
\theta^2 &= (4/3)\ [0\ 0\ 0\ 0\ \ 1\ 1\ 0\ 1] \\
\theta^3 &= (4/3)\ [0\ 0\ 0\ 0\ \ 1\ 0\ 1\ 1] \\
\theta^4 &= (4/3)\ [0\ 0\ 0\ 0\ \ 0\ 1\ 1\ 1]
\end{aligned}$$

Finally, we choose $\mu^i = 1/4$ for all $i = 1, 2, 3, 4$. It is not difficult to check that this choice of a uniform distribution ensures that $\mathbb{E}_{\vec{\mu}}[\theta^i] = \theta^*$; that is, the specified pair $(\boldsymbol{\theta}; \vec{\mu})$ belongs to $\mathcal{A}(\theta^*)$.



(a) θ^1 (b) θ^2 (c) θ^3 (d) θ^4

Figure 3.2. A convex combination of exponential parameters of four distributions $p(\mathbf{x}; \theta^i)$, each defined by a spanning tree, is used to approximate the target distribution $p(\mathbf{x}; \theta^*)$ on the single-cycle graph.

The motivation behind the convex combinations in Definition 3.3.1 is that they allow us to apply Jensen's inequality (3.12) to generate upper bounds on log partition functions. By using equation (3.9), these bounds can be translated to bounds on $\log \mathbb{E}_{\theta^*}[f]$. The results are summarized in the following:

Proposition 3.3.2. Let $f : \mathcal{X}^N \to \mathbb{R}$ satisfy Assumption 3.3.1, and let θ^* be the exponential parameter of target distribution $p(\mathbf{x}; \theta^*)$. For any pair $(\boldsymbol{\theta}; \vec{\boldsymbol{\mu}}) \in \mathcal{A}(\theta^*)$, we have the bounds:

$$\log \mathbb{E}_{\theta^*}[f] \leq \mathbb{E}_{\vec{\boldsymbol{\mu}}}\left[\log \mathbb{E}_{\theta^i}[f]\right] + \mathbb{E}_{\vec{\boldsymbol{\mu}}}\left[\Phi(\theta^i)\right] - \Phi(\theta^*) \tag{3.21a}$$

$$\leq \mathbb{E}_{\vec{\boldsymbol{\mu}}}\left[\log \mathbb{E}_{\theta^i}[f]\right] + \mathbb{E}_{\vec{\boldsymbol{\mu}}}\left\{\mathbb{E}_{\theta^i}\left[\sum_{\alpha}(\theta^i - \theta^*)_\alpha \phi_\alpha(\mathbf{x})\right]\right\} \tag{3.21b}$$

We can also derive a set of lower bounds of similar form by applying the upper bounds in equation (3.21) to the function $\widetilde{f}(\mathbf{x}) = 1 - f(\mathbf{x})$, which also satisfies Assumption 3.3.1.

Proof. See Appendix B.2. □

There are some caveats associated with the bounds of Proposition 3.3.2. First of all, recall that Assumption 3.3.1 implies that $\log \mathbb{E}_{\theta^*}[f] \leq 0$, so that the upper bounds of equation (3.21) are meaningless if the right-hand sides are larger than zero. Unlike Proposition 3.3.1, this condition is not guaranteed for these bounds.

Secondly, as with the form of bounds given in Proposition 3.3.1, the bound of equation (3.21a) is not computable, since it also involves the log partition function $\Phi(\theta^*)$. Required in this case is a lower bound on $\Phi(\theta^*)$. Standard mean field theory [e.g., 105], as described in Section 2.3.1, provides a well-known lower bound on this log partition function. Indeed, in deriving equation (3.21a) from equation (3.21b), we have made use of the mean field lower bound (see equation (2.46)). Of course, it is possible

to use tighter lower bounds on the log partition function that include higher order terms [e.g., 123] , which will lead to correspondingly tighter forms of equation (3.21b).

Proposition 3.3.2 also has important consequences for Proposition 3.3.1, in which the bounds were not computable due to the presence of a log partition function $\Phi(\theta^*)$ in the KL divergence term $D(\theta \,\|\, \theta^*)$. What is required in this case is an upper bound on $\Phi(\theta^*)$. In the special case that $f = \mathbf{1}$, equation (3.21a) provides such an upper bound. Indeed, all the terms involving f vanish, and we are left with the familiar form of Jensen's inequality:

$$\Phi(\theta^*) \leq \mathbb{E}_{\vec{\mu}}[\Phi(\theta^i)] = \sum_{i \in \mathcal{I}} \mu^i \Phi(\theta^i) \tag{3.22}$$

This upper bound can be applied in conjunction with Proposition 3.3.1 so as to yield computable bounds.

Optimizing multiple point bounds

It is also natural to consider the problem of optimizing the exponential parameters $\boldsymbol{\theta} = \{\theta^i\}$, as well as the distribution $\vec{\boldsymbol{\mu}}$. For concreteness, let us consider the special case of $f = \mathbf{1}$, in which case the problem is to minimize the RHS of equation (3.22) — that is, $F(\vec{\boldsymbol{\mu}}; \boldsymbol{\theta}) = \sum_i \mu^i \Phi(\theta^i)$ subject to the constraint

$$\mathbb{E}_{\vec{\mu}}[\theta^i] = \sum_{i \in \mathcal{I}} \mu^i \theta^i = \theta^* \tag{3.23}$$

Interestingly, the cost function F is convex in $\boldsymbol{\theta}$ with $\vec{\boldsymbol{\mu}}$ fixed, and linear (hence convex) in $\vec{\boldsymbol{\mu}}$ with $\boldsymbol{\theta}$ held constant. Moreover, the constraint of equation (3.23) is linear in $\boldsymbol{\theta}$ with $\vec{\boldsymbol{\mu}}$ fixed, and similarly for $\vec{\boldsymbol{\mu}}$ with $\boldsymbol{\theta}$ held fixed. The minimization of a convex function subject to linear constraints is well-behaved (e.g., the minimum is unique for a strictly convex function), and there are a variety of available algorithms [20]. Therefore, optimizing over the collection of exponential parameters $\boldsymbol{\theta}$ with $\vec{\boldsymbol{\mu}}$ fixed (or over $\vec{\boldsymbol{\mu}}$ with $\boldsymbol{\theta}$ fixed) is possible.

However, the joint optimization over $\boldsymbol{\theta}$ and $\vec{\boldsymbol{\mu}}$ is much trickier. In this case, the constraint set consists of (unpleasant) quadratic equality relations. Moreover, even if we could perform this joint optimization, there remains the nagging issue of choosing the set of possible approximating distributions.[2] To be concrete, suppose that we decide to optimize over the set of all spanning trees embedded within the graph. The number of such trees is prohibitively large for typical graphs — e.g., N^{N-2} for the complete graph K_N on N nodes [e.g., 168]. Due to this curse of dimensionality, optimizing the cost function F over all trees appears hopeless. So it is natural to restrict ourselves to a subset of trees, but how to choose them in a principled way?

Remarkably, it turns out these challenging issues — i.e., choice of trees, and the explosion in dimension — can be sidestepped entirely by a suitable dual reformulation

[2]This issue is equally applicable to the single optimization over $\boldsymbol{\theta}$ with $\vec{\boldsymbol{\mu}}$ fixed.

of this optimization problem. As we will show in Chapter 7, this dual reformulation allows us to develop an efficient algorithm for optimizing these bounds over *all* spanning trees of the graph, albeit in an implicit manner.

■ 3.4 Extension to the basic bounds

In this section, we describe a method for strengthening the basic bounds described in the previous section. In particular, we ask what factors control the tightness of the bounds in Proposition 3.3.1 and 3.3.2. One important factor turns out to be the choice of the function f. In particular, suppose that for some configuration $\mathbf{e} \in \mathcal{X}^N$, we set $f(\mathbf{x}) = \delta(\mathbf{x} = \mathbf{e})$. Note that the support of this function f is as small as possible without being empty; it consists only of the single configuration $\mathbf{e}$.

Now consider the equality:

$$\frac{p(\mathbf{x} = \mathbf{e}; \theta^*)}{p(\mathbf{x} = \mathbf{e}; \theta)} = \exp\left\{\Phi(\theta) - \Phi(\theta^*) + \sum_{\alpha} \phi_\alpha(\mathbf{e})\ (\theta^* - \theta)_\alpha\right\}$$

Since $\mathbb{E}_{\theta^*}[\delta(\mathbf{x} = \mathbf{e})] = p(\mathbf{x} = \mathbf{e}; \theta^*)$, it can be seen (following a bit of algebra) that this equation is equivalent to the bounds of Proposition 3.3.1 holding with equality.

This observation suggests that these bounds becomes tighter as the support of the function f decreases. It also forms the basis of a principled method for tightening bounds on the expectation $\mathbb{E}_{\theta^*}[f]$. In particular, given some function f satisfying Assumption 3.3.1, we consider *additive decompositions* of f in the form:

$$f = \sum_{k=1}^{L} f^k \tag{3.24}$$

We call the set $\{f^k\}$ a *partition* of f, and L is the size of the partition. Many functions of interest can be decomposed additively in this manner.

Example 3.4.1. For the choice $f(\mathbf{x}) = \delta(x_s = j)$, we have the decomposition

$$f(\mathbf{x}) = \sum_{k=1}^{m} \delta(x_s = j)\delta(x_t = k)$$

for some node t and state value k. If we take expectations with respect to $p(\mathbf{x}; \theta)$, then this is simply a decomposition of the single node marginal $p(x_s = j; \theta)$ into a sum of joint marginal terms $p(x_s = j, x_t = k; \theta)$.

In the following sections, we show how to exploit additive decompositions to tighten the bounds of both Proposition 3.3.1 and 3.3.2.

■ 3.4.1 Tighter single point bounds

The basic procedure is very simple: given an additive decomposition of the form in equation (3.24), we can use Proposition 3.3.1 to derive bounds on each $\mathbb{E}_{\theta^*}[f^k]$, and then add these individual bounds to derive a new bound on $\mathbb{E}_{\theta^*}[f]$. The following summarizes this procedure, and establishes that it will, in general, improve the bound on $\mathbb{E}_{\theta^*}[f]$.

Proposition 3.4.1. Consider the additive decomposition $f = \sum_{k=1}^{L} f^k$, where f and each f^k are functions from the state space $\mathcal{X}^N$ to $\mathbb{R}$ satisfying Assumption 3.3.1. Then we have the bound

$$\mathbb{E}_{\theta^*}[f] \geq \sum_k \left[\mathbb{E}_\theta[f^k] \exp\left\{ -D(\theta \,\|\, \theta^*) + \frac{1}{\mathbb{E}_\theta[f^k]} \sum_\alpha \operatorname{cov}_\theta\{f^k, \phi_\alpha\}(\theta^* - \theta)_\alpha \right\} \right] \tag{3.25}$$

Moreover, this bound is also at least as good as the bound (3.13a) of Proposition 3.3.1. It is *strictly superior* as long as the terms $\frac{1}{\mathbb{E}_\theta[f^k]} \sum_\alpha \operatorname{cov}_\theta\{f^k, \phi_\alpha\}(\theta^* - \theta)_\alpha$ are not equal for all indices k.

Proof. See Appendix B.3. □

Clearly, computing the bound of Proposition 3.4.1 requires more work — roughly L times more — than the bound of equation (3.13a) in Proposition 3.3.1 . Nonetheless, it has the desirable feature that (in general) performing more computation guarantees a superior result. We shall provide empirical results in Section 3.5.2 showing the gains that can be achieved by this strengthening procedure.

There is an interesting case for which the bound of Proposition 3.4.1 is no better than the lower bound of Proposition 3.3.1. Suppose that we perform mean field optimization over some structure (say a tree), thereby obtaining the optimal mean field parameter $\widehat{\theta}$. Suppose moreover that for each $k = 1, \ldots, L$, we have $f^k = \phi_{\beta(k)}$ for some free index $\beta(k)$. The free indices in mean field optimization over a tree correspond to any single node potentials, and any edge in the tree; see the discussion following Proposition 3.3.1.

In this case, the stationary conditions of mean field, as in equation (3.15), dictate that $\sum_\alpha \operatorname{cov}_{\widehat{\theta}}\{f^k, \phi_\alpha\}(\theta^* - \widehat{\theta})_\alpha = 0$ for all k. Returning to Proposition 3.4.1, the bound in equation (3.25) then reduces to

$$\begin{aligned} \mathbb{E}_{\theta^*}[f] &\geq \sum_k \left[\mathbb{E}_{\widehat{\theta}}[f^k] \exp\left\{ -D(\widehat{\theta} \,\|\, \theta^*) \right\} \right] \\ &= \mathbb{E}_{\widehat{\theta}}[f] \exp\left\{ -D(\widehat{\theta} \,\|\, \theta^*) \right\} \end{aligned} \tag{3.26}$$

As a consequence, the $\{f^k\}$ partition plays no role, and cannot improve the bound. Indeed, equation (3.26) is equivalent to the form of Proposition 3.3.1 that is obtained when θ is equal to a mean field optimum $\widehat{\theta}$. (In particular, compare it to equation (3.16).)

It should be noted that obtaining a (structured) mean field optimum can be a very computationally intensive procedure (much more so than computing bounds for the $\{f^k\}$), so that it is not always feasible. However, presuming that a mean field solution is obtained, using a partition $\{f^k\}$ that includes functions *not* involved in the mean field optimization will, of course, improve the bounds. Therefore, Proposition 3.4.1 can still be used to strengthen a mean field solution. Section 3.5.2 gives an empirical illustration of these phenomena.

■ 3.4.2 Tighter multiple point bounds

Similar intuition suggests that additive decompositions should also be useful for tightening the bounds in Proposition 3.3.2 based on multiple points. As before, we consider the function $f(\mathbf{x}) = \delta(\mathbf{x} = \mathbf{e})$. Then it is not hard to see that the following equality

$$\sum_\alpha \theta^*{}_\alpha \phi_\alpha(\mathbf{e}) - \Phi(\theta^*) \;=\; \mathbb{E}_{\vec{\mu}}\Big\{\sum_\alpha \theta^i_\alpha \phi_\alpha(\mathbf{e}) - \Phi(\theta^i)\Big\} + \mathbb{E}_{\vec{\mu}}[\Phi(\theta^i)] - \Phi(\theta^*)$$

corresponds to the bound in equation (3.21a) holding with equality. This observation leads us to suspect again that the tightness of the bounds increases as the support of f decreases.

This intuition is in fact correct: given an additive decomposition $f = \sum_k f^k$, we can strengthen the bound of Proposition 3.3.2 by bounding each f^k individually, and then summing the bounds. We summarize as follows:

Proposition 3.4.2. Consider the additive decomposition $f = \sum_k f^k$, where f and each f^k are functions from the state space $\mathcal{X}^N$ to $\mathbb{R}$ satisfying Assumption 3.3.1. Then we have the bounds

$$\mathbb{E}_{\theta^*}[f] \;\leq\; \sum_k \Bigg[\prod_i \Big(\mathbb{E}_{\theta^i}[f^k]\Big)^{\mu^i}\Bigg] \exp\Big\{\sum_i \mu^i \Phi(\theta^i) - \Phi(\theta^*)\Big\} \tag{3.27a}$$

$$\mathbb{E}_{\theta^*}[f] \;\geq\; 1 - \sum_k \Bigg[\prod_i \Big(1 - \mathbb{E}_{\theta^i}[f]\Big)^{\mu^i}\Bigg] \exp\Big\{\sum_i \mu^i \Phi(\theta^i) - \Phi(\theta^*)\Big\} \tag{3.27b}$$

Moreover, these bounds are tighter than those given in Proposition 3.3.2 as long as the quantities $\{\mathbb{E}_{\theta^i}[f^k]/\mathbb{E}_{\theta^i}[f]\}$ are not all equal.

Proof. See Appendix B.4. □

■ 3.5 Results on bounding the log partition function

By setting $f \equiv \mathbf{1}$, all of the bounds described in the previous sections reduce to particular bounds on the log partition function $\Phi(\theta^*)$. In this section, we present the results of applying these bounds to various problems. We focus, in particular, on the bounds of Propositions 3.3.1 and 3.4.1.

Consider a partition of $f \equiv \mathbf{1}$: i.e., a set of functions $\{f^k\}$ such that $\sum_k f^k = \mathbf{1}$. For any such partition, Proposition 3.4.1 gives a lower bound on the log partition function $\Phi(\theta^*)$:

$$\Phi(\theta^*) \geq \Phi(\theta) + \log\left[\sum_k \mathbb{E}_\theta[f^k] \exp\left\{\sum_\alpha [\theta^* - \theta]_\alpha \frac{\mathbb{E}_\theta[f^k \phi_\alpha]}{\mathbb{E}_\theta[f^k]}\right\}\right] \tag{3.28}$$

Equation (3.28) follows by substituting $f = \mathbf{1}$ in equation (3.25), and then taking logarithms and simplifying.

In order to illustrate this family of bounds, we focus on additive decompositions $\{f^k\}$ of the form:

$$\mathbf{1} = \sum_{\mathbf{e}_S \in \mathcal{X}^{|S|}} \delta(\mathbf{x}_S = \mathbf{e}_S) \tag{3.29}$$

The indicator function $\delta(\mathbf{x}_S = \mathbf{e}_S)$ for $\mathbf{x}_S$ to assume the configuration $\mathbf{e}_S$ is defined in equation (3.2). With these choices of functions $\{f^i\}$, the expectations $\mathbb{E}_\theta[f^i]$ in equation (3.28) correspond to the values of marginal distributions over the nodes in S.

For a given graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we performed simulations for a binary process $\mathbf{x} \in \{0,1\}^N$ by forming a distribution $p(\mathbf{x}; \theta^*)$ with a random choice of θ^* from either the uniform attractive ensemble, or the uniform mixed ensemble, in both cases using edge strength $d = 2$. See Section 2.2.1 for the definitions of these ensembles of distributions. In all cases (experimental conditions and graphs), we investigated the effect of increasing the number of nodes (and correspondingly, the number of functions) in the partition given in equation (3.29). Note that for a binary process, a partition of this form based on $|S|$ nodes consists of $2^{|S|}$ functions. The special case $|S| = 0$ corresponds to choosing only a single function $f^1(\mathbf{x}) = \mathbf{1}$, so that the bound of equation (3.28) reduces to the ordinary mean field bound, as in equation (2.46) of Section 2.3.1.

For each trial, we computed bounds based on some approximating point θ, chosen in a way to be specified below. Given this approximating distribution $p(\mathbf{x}; \theta)$, we first computed the ordinary mean field bound[3] for $|S| = 0$. Then for sizes $|S| = 1, 2, 3$, we computed the bound of equation (3.28) for each of the $\binom{N}{|S|}$ possible subsets of size $|S|$ in a graph on N nodes.

■ 3.5.1 Unoptimized bounds

We first investigated the effect of refining the partition on two graphs (a 3×3 grid, and the fully connected graph K_9 on $N = 9$ nodes). The small problem size facilitates comparison of the bounds to the true value of $\Phi(\theta^*)$. For each each graph, we performed simulations under both the attractive and mixed conditions. To form the approximating distribution, we first used Kruskal's algorithm [107, 116] to compute the maximum weight spanning tree $\mathcal{T}$ based on the edge weights $|\theta^*_{st}|$ on each edge $(s,t) \in \mathcal{E}$. Let

[3]This is not an optimized mean field bound unless θ is a mean field optimum.

$\mathcal{E}(\mathcal{T}) \subset \mathcal{E}$ be the edge set of maximum weight spanning tree. We then formed a tree-structured distribution by setting

$$\theta_\alpha = \begin{cases} \theta_s^* & \text{if} \quad \alpha = s \in \mathcal{V} \\ \theta_{st}^* & \text{if} \quad \alpha = (s,t) \in \mathcal{E}(\mathcal{T}) \\ 0 & \text{if} \quad \alpha = (s,t) \in \mathcal{E}/\mathcal{E}(\mathcal{T}) \end{cases}$$

Using the distribution $p(\mathbf{x};\theta)$, we computed bounds as described above.

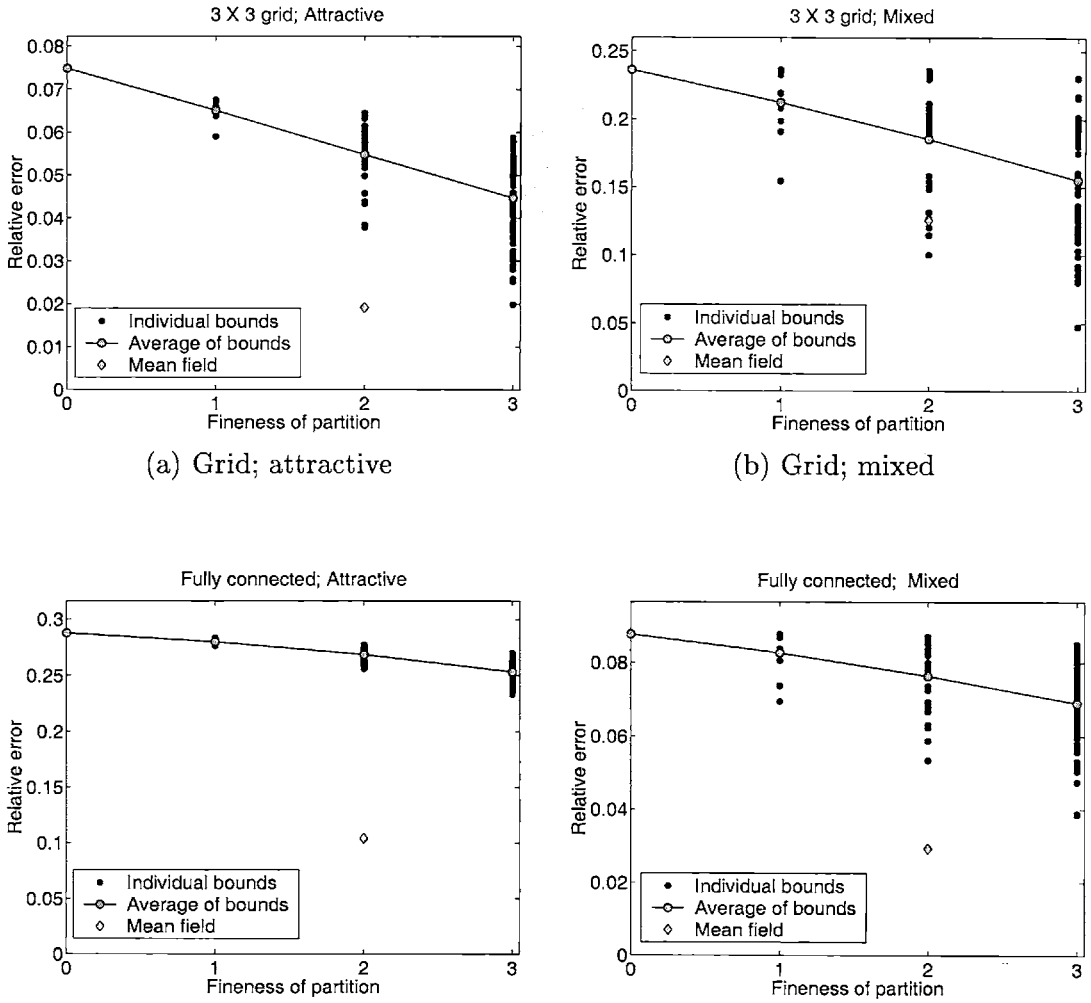
The results are shown in Figure 3.3, with plots for the 3×3 grid shown in the top row, and those for the fully connected graph shown in the bottom row. On the abscissa, we plot the number of nodes $|S|$ used in the refinement, ranging from 0 to 3; on the y-axis, we plot the relative error in bounds (i.e., $[\Phi(\theta^*) - \text{Bound}]/\Phi(\theta^*)$). For each size $|S|$, we show in a vertically-oriented scatter plot the relative error in all $\binom{9}{|S|}$ possible bounds based on subsets of this size. We also plot the mean relative error averaged over all possible subsets. Finally, for the purposes of comparison, in the column $|S| = 2$, we plot a single point with a diamond that corresponds to the relative error in the optimized structured mean field solution for the spanning tree $\mathcal{T}$.

We see that refining the partition (in general) improves the bounds, as illustrated by the downward trend in the means. The scatter plots of the individual bounds show that the tightness of the bounds varies a fair bit, especially for the case of mixed potentials. This variability underscores the fact that finding methods for choosing good subsets of nodes is important.

Note that Proposition 3.4.1 guarantees that refining the partition will (in general) improve a pair of bounds that are nested. For example, it ensures that the bound based on nodes $\{1,2\}$ is better than that based on node $\{1\}$; it does not, however, guarantee that the former bound will be better than that based on any other single node $s \neq 1,2$. As a consequence, we can see that for $k = 1,2$, not all of the bounds with $|S| = k+1$ are better than the best of the bounds with $|S| = k$. However, we see that the worst (respectively the best) of the bounds with $|S| = k+1$ are always better, or at least as good, as the worst (respectively the best) bounds with $|S| = k$.

The mean field solution (plotted with a diamond in column $|S| = 2$) is better than all of these bounds in three out of four cases. Of course, such a comparison is not really fair, since each iteration of structured mean field optimization[4] requires roughly $\mathcal{O}(N + |\mathcal{E}|)$ as much computation as a single bound of the form in equation (3.28). Moreover, many iterations are typically required; for these examples, mean field required more than 20 iterations to converge to a precision of 1×10^{-4}, measured in terms of percentage change in the bound value. In the following example, we shall do a more direct comparison to mean field.

[4]There are a variety of ways of performing structured mean field optimization, but roughly, each iteration requires computing the Fisher information matrix, which is expensive.



(a) Grid; attractive

(b) Grid; mixed

(c) Fully connected; attractive

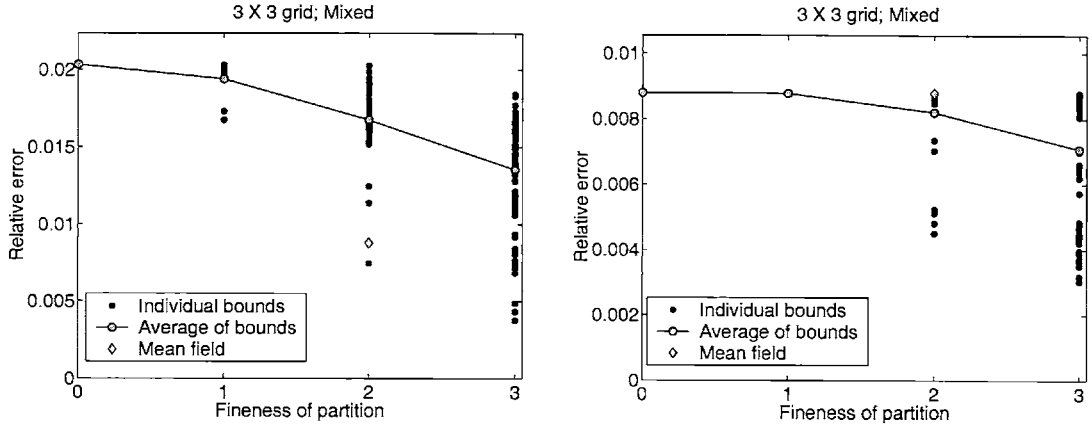
(d) Fully connected; mixed

Figure 3.3. Improved bounds on the log partition function based on refining the partition for an unoptimized approximating point θ. Each panel plots the relative error $[\Phi(\theta^*) - \text{Bound}]/\Phi(\theta^*)$ in the bounds versus the partition size (number of nodes $k = 0, 1, 2, 3$). Shown for each k are the errors for all $\binom{9}{k}$ possible bounds (corresponding to all possible combinations of k nodes from 9 nodes in total). Also shown are the average error, and the error in the structured mean field bound (plotted at $k = 2$). Top row: 3×3 grid. Bottom row: Fully connected graph on 9 nodes.

■ 3.5.2 Bounds with optimal mean field vector

In our second simulation, we used a mixed parameter vector θ^* on a 3×3 grid to compare the effect of refining the partition for an unoptimized parameter vector θ (chosen as in the previous experiment), and the structured mean field optimum $\widehat{\theta}$.

The results are plotted in Figure 3.4. Panel (a) shows a plot, analogous to those of Figure 3.3, for the unoptimized approximating point. The qualitative behavior is



(a) Unoptimized (b) Optimized (mean field)

Figure 3.4. Effect of refining the partition for unoptimized versus mean field solution on a 3×3 grid. Each panel plots the relative error $[\Phi(\theta^*) - \text{Bound}]/\Phi(\theta^*)$ in the bounds versus the partition size (number of nodes $k = 0, 1, 2, 3$). Shown for each k are the errors for all $\binom{9}{k}$ possible bounds (corresponding to all possible combinations of k nodes from 9 nodes in total), as well as the average errors. (a) Unoptimized solution on spanning tree. (b) Optimal structured mean field solution on spanning tree.

similar to the top panel of Figure 3.3. Note that the relative error in the optimized structured mean field bound, shown with a diamond, is quite good.

Panel (b), in contrast, shows the effect of refining the partition when using the optimal structured mean field vector $\widehat{\theta}$ as the approximating point. Since we are using this optimal point, the base error for $|S| = 0$ has decreased to ≈ 0.008 (or 0.8%). An interesting feature of this plot is that using a refined partition of size $|S| = 1$ has absolutely no effect on the tightness of the bound. The discussion following Proposition 3.4.1 gives a theoretical explanation of this effect: in particular, the functions $\{\delta(x_s = j)\}$ associated with any refinement of size one all correspond to functions that are optimized under structured mean field. Hence, refinements using these functions have no effect. A similar statement applies to certain subsets of size $|S| = 2$ — namely, those corresponding to edges in the approximating tree. As a consequence, the plot of the mean relative error is somewhat misleading. It is skewed upwards for both $|S| = 1$ and 2, since the average includes many subsets that we know *a priori* will not improve the mean field solution. For larger partitions, however, refinements will typically lead

to further improvements upon the optimized mean field solution.

■ 3.6 Discussion

Exponential families of distributions capture, in a compact manner, both the model structure and model parameters. In this chapter, we have demonstrated their power in application to two important problems: understanding model sensitivity via perturbation expansions, and deriving computable lower and upper bounds on quantities of interest, including marginal distributions and partition functions. Indeed, the new class of upper bounds derived in this chapter, as described Section 3.3.3, follow in an elegant way from the perspective of an exponential representation.

The bounds of this chapter play an important role in developments in the sequel. In particular, in Chapter 5, we will apply the results of this chapter, as well as those of Chapter 7, in order to derive upper and lower bounds on the approximation error that arises in applying the belief propagation error. Moreover, these same results will be used to bound the error in the more advanced techniques for approximate inference that are analyzed in Chapter 6. Finally, Chapter 7 is devoted to a detailed analysis of the upper bounds presented in Section 3.3.3.

Chapter 4

Embedded trees algorithm for Gaussian processes

4.1 Introduction

In areas like coding theory [71, 117], artificial intelligence [137], and speech processing [143], graphical models typically involve discrete-valued random variables. However, in other domains such as image processing, control theory, and oceanography [35, 62, 126], it is often more appropriate to consider random variables with a continuous distribution. In this context, Gaussian processes defined by graphical models are of great practical significance. Moreover, the Gaussian case provides a valuable setting for developing an understanding of estimation algorithms [152, 174].

Accordingly, the focus of this chapter is estimation of Gauss-Markov processes on graphs. Throughout this chapter, the term estimation refers to the computation of conditional means and error covariances at each node of the graph. For a Gauss-Markov process on a tree-structured graph, Chou et al. [35] developed a recursive and very efficient algorithm for exact estimation. This algorithm has a two-pass form, and represents a generalization of the Kalman filter [109, 110] and Rauch-Tung-Striebel smoother [146]. This estimation algorithm, and associated techniques for constructing tree-structured models [e.g., 63, 88, 89], have been used successfully in a wide variety of applications [e.g., 47, 62, 87, 126].

A well-known problem associated with tree models is the presence of boundary artifacts. In particular, tree models may introduce artificial discontinuities between pairs of nodes that, though spatially or temporally close, are separated by a great distance in the tree. Various methods have been proposed to deal with this problem [e.g., 88], but these proposals are not entirely satisfactory. The most natural solution is to add extra edges, as necessary, to account for statistical dependencies neglected by a tree model. With the addition of these edges, however, the resulting graph contains cycles, meaning that efficient tree algorithms [35] for exact estimation are no longer applicable.

An important problem, therefore, is to develop algorithms for exact estimation of a Gauss-Markov process defined on a graph with cycles. In this chapter, we develop and analyze an algorithm that exactly computes *both* the conditional mean and error variances of a Gaussian random vector $\mathbf{x}$ based on a set of noisy observations $\mathbf{y}$. As a central

engine, we exploit the existence of fast algorithms for performing exact computations with tree-structured distributions. Each step of the algorithm entails extracting a tree embedded within the original graph with cycles, and performing exact calculations with a modified distribution defined by this tree. For this reason, we call our technique the *embedded trees* (ET) algorithm. Given a set of noisy measurements, it computes the conditional means with an efficiency comparable to or better than other techniques for graphs with cycles. Unlike other methods, the ET algorithm also computes exact error covariances at each node of the graph. In many applications [e.g., 62, 126], these error statistics are as important as the conditional means.

This chapter is organized in the following manner. In Section 4.2, we provide background on estimation of Gaussian processes defined on graphs. Section 4.3 introduces the embedded trees (ET) algorithm, and presents results on its convergence properties. We conclude in Section 4.4 with a summary, and directions for future research. The work described in this chapter was based on collaboration with Erik Sudderth, and portions of it have appeared previously in the conference paper [172]. Extensions to this work are described in [163].

4.2 Estimation of Gaussian processes

This section provides the basics of linear-Gaussian estimation, with particular emphasis on Gaussian processes that are Markov with respect to a graph. More details on Gaussian processes and estimation can be found in the book [108], as well as in [163].

4.2.1 Prior model and observations

We consider a zero-mean[1] Gaussian random vector $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, P)$ with strictly positive definitive covariance matrix P. We assume that $\mathbf{x}$ is partitioned into a set of subvectors $\{ \mathbf{x}_s \mid s = 1, \ldots, N \}$. Denoting by $l(\mathbf{x}_s)$ the dimension of $\mathbf{x}_s$, the total number of elements in the vector $\mathbf{x}$ is given by $l(\mathbf{x}) = \sum_{s=1}^{N} l(\mathbf{x}_s)$. We let $d = \max_s l(\mathbf{x}_s)$ denote the maximal size of any of the subvectors $\mathbf{x}_s$.

Let $\mathbf{y}$ be a set of noisy observations of $\mathbf{x}$. In many problem domains, the observations[2] $\mathbf{y} = \{\mathbf{y}_s \mid s \in \mathcal{A} \subseteq \{1, \ldots, N\} \}$ are naturally expressed as a noise-corrupted linear function of $\mathbf{x}$ as follows:

$$\mathbf{y} = C\mathbf{x} + \mathbf{v} \tag{4.1}$$

Here $\mathbf{v} \sim \mathcal{N}(0, R)$ is zero-mean additive Gaussian noise, independent of $\mathbf{x}$. We assume that both C and R have a block-diagonal structure that respect the partition of $\mathbf{x}$ into subvectors $\{ \mathbf{x}_s \mid s = 1, \ldots, N \}$. As a consequence, observations $\mathbf{y}_s$ and $\mathbf{y}_t$ at distinct nodes $s \neq t$ are conditionally independent given $\mathbf{x}_s$ (or given $\mathbf{x}_t$).

[1] It is straightforward to incorporate a non-zero mean by the appropriate addition of terms.

[2] The set $\mathcal{A} \subseteq \{1, \ldots, N\}$ may be a subset, since we may not have observations of every subvector $\mathbf{x}_s$.

■ 4.2.2 Linear-Gaussian estimation

For estimation purposes, we are interested in the conditional density $p(\mathbf{x} \mid \mathbf{y})$ of $\mathbf{x}$ given the observations $\mathbf{y}$. With a linear observation model of the form in equation (4.1), it can be shown that $\mathbf{x}$ and $\mathbf{y}$ are jointly Gaussian [108], and moreover that $\mathbf{x}$ conditioned on $\mathbf{y}$ is Gaussian. Therefore, the density $p(\mathbf{x} \mid \mathbf{y})$ is Gaussian, and can be characterized completely by its mean $\widehat{\mathbf{x}}$ and covariance $\widehat{P}$. Also of interest are the marginal densities $p(\mathbf{x}_s \mid \mathbf{y})$ of $\mathbf{x}_s$ conditioned on the noisy observations $\mathbf{y}$ for each node $s \in \mathcal{V}$. Since the full conditional density is Gaussian, these marginal densities are also Gaussian; in particular, $p(\mathbf{x}_s \mid \mathbf{y}) \sim \mathcal{N}(\widehat{\mathbf{x}}_s, \widehat{P}_s)$. Standard formulae exist for the computation of these quantities — viz.:

$$\widehat{P}^{-1}\,\widehat{\mathbf{x}} = C^T R^{-1} \mathbf{y} \tag{4.2a}$$

$$\widehat{P}^{-1} = \left[P^{-1} + C^T R^{-1} C\right] \tag{4.2b}$$

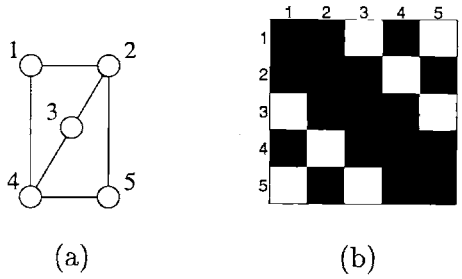
The vector $\widehat{\mathbf{x}}$ is the *conditional mean* of the random variable $\mathbf{x}$ conditioned on $\mathbf{y}$. The quantity $\widehat{P}$ is often called the *error covariance* matrix, since it corresponds to the covariance matrix of the error $\widetilde{\mathbf{e}} = \widehat{\mathbf{x}} - \mathbf{x}$ in the error. The smaller $l(\mathbf{x}_s) \times l(\mathbf{x}_s)$ covariance matrices $\widehat{P}_s$ correspond to block diagonal elements of the full error covariance $\widehat{P}$. Equations (4.2a) and (4.2b) are the *normal equations* [108] that define the problem of linear-Gaussian estimation.

■ 4.2.3 Gauss-Markov processes and sparse inverse covariance

As we will discuss, there exist iterative algorithms from numerical linear algebra [54] for solving the linear system in equation (4.2a). Otherwise, calculating the full error covariance $\widehat{P}$ by brute force matrix inversion would, in principle, provide error variances (as well as the conditional means). Since the computational complexity of matrix inversion is $\mathcal{O}([dN]^3)$, this proposal is not practically feasible in many applications, such as large-scale image processing and oceanography [e.g., 62, 126, 127, 171], where dN may be on the order of 10^5. The intractability of the general case motivates considering problems with more structure.

An important type of structure arises for a Gaussian random vector $\mathbf{x} \sim \mathcal{N}(0, P)$ that is *Markov*, in the sense of Definition 2.1.6, with respect to an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. With respect to this graph, the subvectors $\mathbf{x}_s$ forming $\mathbf{x}$ lie at particular nodes $s \in \mathcal{V} = \{1, \ldots, N\}$ of the graph. In this case, it can be shown [see 160] that the *inverse* covariance matrix P^{-1} inherits a sparse structure from $\mathcal{G}$. In particular, if P^{-1} is partitioned into blocks according to the subvectors $\{\, \mathbf{x}_s \mid s \in \mathcal{V} \,\}$, the $(s,t)^{th}$ block can be nonzero only if edge $(s,t) \in \mathcal{E}$.

For scalar Gaussian variables at each node, the relation between the structure of the graph and that of the inverse covariance is illustrated in Figure 4.1. Panel (a) shows a simple graph $\mathcal{G}$, whereas panel (b) shows the structure of an inverse covariance matrix consistent with a Gaussian random vector that is Markov with respect to $\mathcal{G}$. In particular, the locations of (possibly) non-zero entries are shown in black. The matrix



(a) (b)

Figure 4.1. Gauss-Markov processes have inverse covariances that respect graph structure. (a) Simple graph $\mathcal{G}$ with cycles. (b) Structure of inverse covariance with non-zero entries shown in black. Entry (s,t) is non-zero only if edge (s,t) belongs to the graph.

elements shown in white (e.g., $(3,5)$) must be zero, since the associated graph lacks an edge between the corresponding nodes (e.g., there is no edge between nodes 3 and 5).

■ 4.2.4 Estimation techniques

There are a variety of techniques for estimation of Gaussian processes that are based on exploiting Markov structure. First of all, when $\mathcal{G}$ is tree-structured, Chou et al. [35] have shown that both the conditional mean $\widehat{\mathbf{x}}_s$ and error covariances $\widehat{P}_s$ at each node can be computed by a very efficient $\mathcal{O}(d^3N)$ algorithm [35]. It entails first specifying an arbitrary node as the root of the tree, and then passing means and covariances up the tree from the leaves to the root, and then back down from the root to the leaves. Thus, it has a two-pass form, and represents a generalization of classic Kalman and Rauch-Tung-Striebel smoothing algorithms [109,146] for time series. A full derivation of this algorithm can be found in Chou et al. [35]; see also Appendix A for a related algorithm for estimation on trees.

Secondly, one of the best-known and most widely studied inference algorithms is *belief propagation* [137]. This algorithm has attracted a great deal of attention, due to its use in computer vision and artificial intelligence [e.g., 65,68,133], and also for its role in decoding turbo codes [130] and low density parity check codes [70], in which context it is known as the sum-product algorithm [e.g., 1,117]. Later in this thesis (Chapters 5 and 6), we will discuss belief propagation (BP) at much more length.[3] The viewpoint taken in these later chapters will be of BP as an approximate inference technique for discrete-valued processes.

Of interest in this chapter is BP in application to Gaussian problems. For tree-structured graphs, belief propagation produces results equivalent to the tree algorithm of Chou et al. [35]. In recent work, two groups [152,174] have analyzed BP in application to Gaussian processes defined on graphs with cycles. For graphs with cycles, these groups showed that when belief propagation (BP) converges, it computes the

[3]See Section 5.1 for an overview of previous work on BP, and Section 5.2.2 for the belief propagation equations.

correct conditional means. That is, the BP means are *exact* (when the algorithm converges). However, in general, the error covariances computed by BP are incorrect. The complexity per iteration of BP on a graph with cycles is $\mathcal{O}(d^3N)$, where one iteration corresponds to updating each message once.[4] See [163] for a more thorough exposition and analysis of Gaussian belief propagation.

Thirdly, it can be seen from equation (4.2a) that computing the conditional mean $\widehat{\mathbf{x}}$ is equivalent to solving a linear system. Given the sparsity of P^{-1}, a variety of iterative techniques from numerical linear algebra [54] could be used to solve this linear system. For a symmetric positive definite system like equation (4.2a), the method of choice is conjugate gradient [54, 85], for which the associated cost is $\mathcal{O}(d^2N)$ per iteration. However, such techniques compute only the means and not the error covariances.

■ 4.3 Embedded trees algorithm

In this section, we develop an iterative algorithm for computing both the conditional means and exact error covariances of a Gaussian process defined on any graph. Central to the algorithm is the operation of cutting edges from a graph with cycles to reveal an embedded tree — i.e., an acyclic subgraph of the original graph. Standard tree algorithms [35] can be used to exactly solve the modified problem, and the results are used in a subsequent iteration.

Interestingly, the algebraic analog of removing edges from the graph is a matrix splitting of the inverse covariance matrix. Matrix splitting is widely used in numerical linear algebra; see, for example, Demmel [54] for an overview of standard matrix splitting methods, and their role in Richardson methods like Gauss-Jacobi iterations. In contrast to classical matrix splittings, those considered here are based on exploiting particular features of the graph structure.

■ 4.3.1 Embedded trees and matrix splitting

An important fact is that embedded within any graph $\mathcal{G}$ are a large number of *spanning trees* — i.e., acyclic subgraphs that reach every node of $\mathcal{G}$. (See Section 2.1.1 for relevant definitions from graph theory). In general, the number of spanning trees in a graph can be computed via the Matrix-Tree theorem [e.g., 168]. Figure 4.2 provides an illustration for the 5×5 nearest-neighbor grid drawn in panel (a). Depicted in panels (b) and (c) are two of the 557,568,000 spanning trees embedded within the 5×5 grid.

For a Gaussian process on a graph, the operation of removing edges corresponds to a particular modification of the inverse covariance matrix. Specifically, given the original inverse covariance P^{-1}, we apply a matrix splitting

$$P_{\text{tree}}^{-1} = P^{-1} + K \tag{4.3}$$

[4]This complexity assumes that the graph is relatively sparse, in that the number of neighbors per node is $\mathcal{O}(1)$ relative to the total number of nodes N.

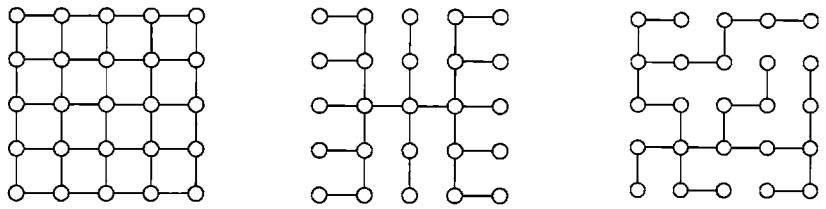


Figure 4.2. (a) Original graph $\mathcal{G}$ is a 5×5 grid. Panels (b) and (c) show two different spanning trees embedded within $\mathcal{G}$.

where K_n is a symmetric *cutting matrix*. It is chosen to ensure that P_{tree}^{-1} corresponds to a valid tree-structured inverse covariance matrix. I.e., P_{tree}^{-1} must be positive semidefinite, and respect the structure constraints of the associated tree.

For a Gaussian process with a scalar random variable at each node, Figure 4.3 illustrates the correspondence between the algebraic matrix splitting of equation (4.3), and its graphical consequences. Panel (a) shows the original graph $\mathcal{G}$ with cycles, whereas panel (b) shows the corresponding inverse covariance matrix (black squares indicate non-zero entries). We decide to cut to the spanning tree shown in panel (c); the corresponding tree-structured inverse covariance matrix is shown in panel (d). (Note that since the tree of (c) is, in fact, a chain, the inverse covariance of (d) has the familiar tridiagonal structure of a Markov time series.) Cutting to this tree entails removal of edges $(1,4)$ and $(2,5)$ from $\mathcal{G}$, as shown in (e). The structure of the simplest possible cutting matrix K is shown in (f). Algebraically, this cutting matrix can be written as

$$K = -P_{14}^{-1}\left[\mathbf{e}_1\mathbf{e}_4^T + \mathbf{e}_4\mathbf{e}_1^T\right] - P_{25}^{-1}\left[\mathbf{e}_2\mathbf{e}_5^T + \mathbf{e}_5\mathbf{e}_2^T\right]$$

where $\mathbf{e}_s$ denotes the vector of all zeros, with a single one in position s. Here we have assumed that the diagonal entries of K are zero, although modifying them is also a possibility.

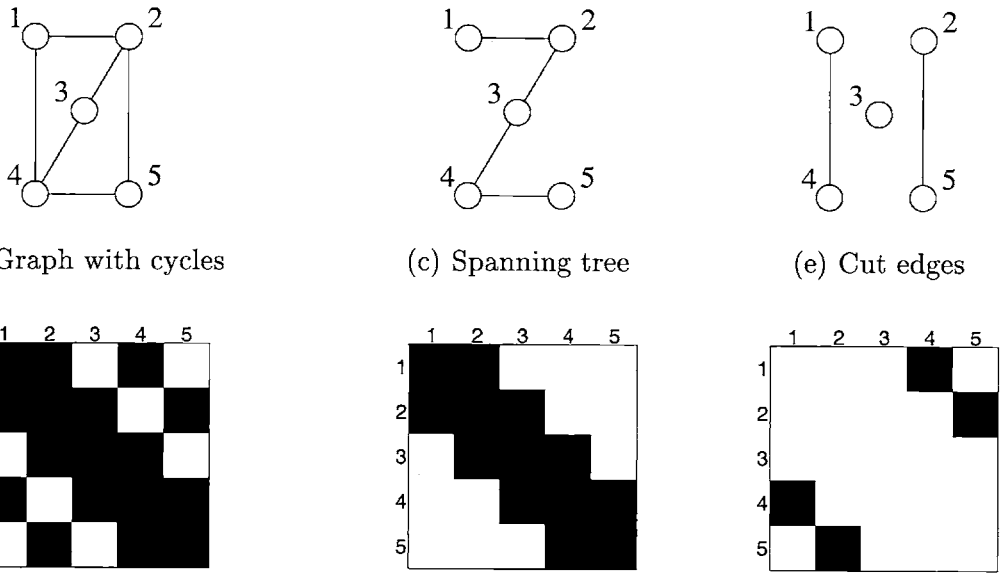
■ 4.3.2 Recursions for computing the conditional mean

On the basis of matrix splitting of equation (4.3), we can rewrite the defining normal equation (4.2a) for the conditional mean $\widehat{\mathbf{x}}$ as follows:

$$\left[P_{\text{tree}}^{-1} + C^T R^{-1} C\right]\widehat{\mathbf{x}} = K\,\widehat{\mathbf{x}} + C^T R^{-1}\mathbf{y} \tag{4.4}$$

Because the "observations" $(K\widehat{\mathbf{x}} + C^T R^{-1}\mathbf{y})$ in equation (4.4) depend on the conditional mean $\widehat{\mathbf{x}}$, equation (4.4) does not provide a direct solution to the original inference problem. It does, however, suggest a natural iterative solution. Let $\{\mathcal{T}^n\}_{n=0}^{L-1}$ be a set of spanning trees of $\mathcal{G}$, and $\{K_n\}_{n=0}^{L-1}$ a corresponding set of symmetric cutting matrices such that for each $n = 0, 1, \ldots, L-1$, the matrix

$$\widehat{J}_n \triangleq P^{-1} + K_n + C^T R^{-1} C \;\equiv\; P_{\text{tree}(n)}^{-1} + C^T R^{-1} C \tag{4.5}$$



(a) Graph with cycles (c) Spanning tree (e) Cut edges

(b) Original inverse covariance (d) Tree inverse covariance (f) Cutting matrix

Figure 4.3. Graphical illustration of tree-cutting operation for a Gaussian process with a scalar random variable at each node. (a) Structure of original graph $\mathcal{G}$ with cycles. (b) Inverse covariance P^{-1} for Gaussian process on original graph. Black squares correspond to non-zero entries. (c) Spanning tree embedded within original graph. (d) Tree-structured inverse covariance P_{tree}^{-1}. (e) Edges to be removed by the cutting matrix. (f) Structure of cutting matrix K.

has a sparsity pattern that respects the Markov properties of the tree $\mathcal{T}^n$. Moreover, we assume that each K_n is chosen such that each $\widehat{J}_n$ is positive definite.[5]

At each iteration, we choose a spanning tree index $i(n) \in \{0, \ldots, L-1\}$ according to some rule. A natural choice is the *cyclic ordering* in which

$$i(n) \equiv n \mod(L) \tag{4.6}$$

A variety of other orderings, some of them random, are discussed in [30].

Using equations (4.4) and (4.5), we may start with some initial vector $\mathbf{x}^0$, and generate a sequence of iterates $\{\widehat{\mathbf{x}}^n\}_{n=1}^{\infty}$ via the recursion

$$\widehat{J}_{i(n)}\widehat{\mathbf{x}}^n = K_{i(n)}\widehat{\mathbf{x}}^{n-1} + C^T R^{-1}\mathbf{y} \tag{4.7}$$

If the cutting matrix $K_{i(n)}$ is chosen so that $\widehat{J}_{i(n)}$ is positive definite, equation (4.7) is precisely equivalent to a Gaussian inference problem defined on a tree-structured

[5]We make this assumption in order to make a clear connection to tree-structured inference algorithms. More generally, however, it is sufficient to choose K_n so that $\widehat{J}_n$ is invertible.

Markov random field. It can therefore be solved using standard fast algorithms [e.g., 35], allowing $\widehat{\mathbf{x}}^n$ to be calculated as

$$\widehat{\mathbf{x}}^n = \widehat{J}_{i(n)}^{-1} \left(K_{i(n)} \widehat{\mathbf{x}}^{n-1} + C^T R^{-1} \mathbf{y} \right) \tag{4.8}$$

The computational cost associated with equation (4.8) is $\mathcal{O}(d^3 N + cd^2)$, where N is the number of nodes, $c = |\mathcal{E}| - (N-1)$ is the number of cut edges.[6] Typically c is at most $\mathcal{O}(N)$, and the overall cost of each iteration is $\mathcal{O}(d^3 N)$.

Taking the difference between the relations implied by equation (4.7) at iterations $n-1$ and n leads to the relation:

$$\widehat{J}_{i(n)} \widehat{\mathbf{x}}^n - \widehat{J}_{i(n-1)} \widehat{\mathbf{x}}^{n-1} = K_{i(n)} \widehat{\mathbf{x}}^{n-1} - K_{i(n-1)} \widehat{\mathbf{x}}^{n-2} \tag{4.9}$$

Noting from equation (4.5) that $\widehat{J}_{i(n)} - K_{i(n)} = \widehat{J}_{i(n-1)} - K_{i(n-1)}$, we may rewrite (4.9) as

$$\left(\widehat{\mathbf{x}}^n - \widehat{\mathbf{x}}^{n-1} \right) = \widehat{J}_{i(n)}^{-1} K_{i(n-1)} \left(\widehat{\mathbf{x}}^{n-1} - \widehat{\mathbf{x}}^{n-2} \right) \tag{4.10}$$

where the initial condition $(\widehat{\mathbf{x}}^1 - \widehat{\mathbf{x}}^0)$ is determined according to equation (4.8). Equation (4.10) explicitly reveals the important fact that the dynamics of the ET algorithm depend solely on the chosen set of cutting matrices K_n. The observations $\mathbf{y}$ act only to set the initial conditions, and do not affect the rate at which the the algorithm converges (i.e., the rate at which the successive differences $(\widehat{\mathbf{x}}^n - \widehat{\mathbf{x}}^{n-1})$ decay). This data independence will play an important role in our subsequent analysis of the convergence properties of the ET iterations.

■ 4.3.3 Convergence analysis

In this section, we determine the conditions under which the embedded trees iteration (4.7) converges. We have assumed that the cutting matrices K_n are chosen so that $\widehat{J}_n$ is positive definite, ensuring that each iterate may be unambiguously calculated using equation (4.8). This equation defines a linear system, so that eigenvalues play a crucial role in the analysis. Let the set of all eigenvalues of a matrix A by denoted by $\{\lambda_i(A)\}$. The spectral radius of A is defined as $\rho(A) \triangleq \max_{\lambda \in \{\lambda_i(A)\}} |\lambda|$.

Our analysis focuses on the evolution of the error $\widetilde{\mathbf{e}}^n \triangleq (\widehat{\mathbf{x}}^n - \widehat{\mathbf{x}})$ between the estimate $\widehat{\mathbf{x}}^n$ at the n^{th} iteration and the solution $\widehat{\mathbf{x}}$ of the original inference problem in equation (4.2a). Using equation (4.2a), we may rewrite the ET recursion (4.7) as

$$\widehat{J}_{i(n)} \widehat{\mathbf{x}}^n = K_{i(n)} \widehat{\mathbf{x}}^{n-1} + \widehat{J}_{\text{orig}} \widehat{\mathbf{x}} = K_{i(n)} \widehat{\mathbf{x}}^{n-1} + \left(\widehat{J}_{i(n)} - K_{i(n)} \right) \widehat{\mathbf{x}} \tag{4.11}$$

where $\widehat{J}_{\text{orig}} \triangleq P^{-1} + C^T R^{-1} C$. This equation may be rewritten to relate the errors at subsequent iterations:

$$\widetilde{\mathbf{e}}^n = \widehat{J}_{i(n)}^{-1} K_{i(n)} \widetilde{\mathbf{e}}^{n-1} \tag{4.12}$$

Together, equations (4.12) and (4.10) lead to the following result.

[6] Here we have used the fact that any spanning tree of a graph with N nodes has $N-1$ edges.

Proposition 4.3.1. For any starting point $\widehat{\mathbf{x}}^0$, the conditional mean $\widehat{\mathbf{x}}$ of the original inference problem (4.2a) is the unique fixed point of the iterates $\{\widehat{\mathbf{x}}^n\}_{n=1}^{\infty}$ generated by the embedded trees recursion (4.8). Moreover, the error $\widetilde{\mathbf{e}}^n \triangleq (\widehat{\mathbf{x}}^n - \widehat{\mathbf{x}})$ evolves according to:

$$\widetilde{\mathbf{e}}^n = \left[\widehat{J}_{i(n)}^{-1} K_{i(n)} \cdots \widehat{J}_{i(2)}^{-1} K_{i(2)} \widehat{J}_{i(1)}^{-1} K_{i(1)}\right] \widetilde{\mathbf{e}}^0 \tag{4.13}$$

Proof. The uniqueness of the fixed point $\widehat{\mathbf{x}}$ follows directly from the invertibility of $\widehat{J}_{\text{orig}}$ and $\widehat{J}_n = \widehat{J}_{\text{orig}} + K_n$. Equation (4.13) follows by induction from equation (4.12). □

Although Proposition 4.3.1 shows that the ET recursion has a unique fixed point at the optimal solution $\widehat{\mathbf{x}}$, it does not guarantee that $\widehat{\mathbf{x}}^n$ will converge to that fixed point. In fact, if the cutting matrices K_n are poorly chosen, $\widehat{\mathbf{x}}^n$ may diverge from $\widehat{\mathbf{x}}$ at a geometric rate. The following result specifies the conditions, for a cyclic ordering of trees, under which the ET recursions converge or diverge.

Proposition 4.3.2. With a cyclic ordering of trees, convergence of the ET algorithm is governed by the spectral radius of

$$\mathbf{A} \triangleq \left[\widehat{J}_{L-1}^{-1} K_{L-1} \cdots \widehat{J}_1^{-1} K_1 \widehat{J}_0^{-1} K_0\right] \tag{4.14}$$

In particular, if $\rho(\mathbf{A}) < 1$, then $(\widehat{\mathbf{x}}^n - \widehat{\mathbf{x}}) \stackrel{n\to\infty}{\longrightarrow} 0$ geometrically at rate $\gamma \triangleq \rho(\mathbf{A})^{\frac{1}{L}}$, whereas if $\rho(\mathbf{A}) > 1$, then the algorithm will not converge.

Proof. With a cyclic ordering of trees, the error $\widetilde{\mathbf{e}}^n$ in the ET algorithm evolves according to the dynamics of periodic time-varying linear system (see equation (4.13)). After subsampling it at intervals of L, it becomes a homogeneous linear system controlled by the matrix $\mathbf{A}$. Thus, the convergence or divergence of the ET iterates is controlled by the spectral radius of $\mathbf{A}$. □

On the basis of Proposition 4.3.2, we see that it is important to choose the cutting matrices so that the special radius of $\mathbf{A}$ is less than one. It is not straightforward to analyze this spectral radius in general, since it depends on interactions between successive cutting matrices. Nonetheless, for the special case of cutting to a single tree, the following theorem, adapted from results in [10], gives conditions guaranteeing the validity and convergence of the ET algorithm.

Theorem 4.3.1. Define $\widehat{J}_{\text{orig}} \triangleq P^{-1} + C^T R^{-1} C$, and $\widehat{J} \triangleq \widehat{J}_{\text{orig}} + K$. Suppose the cutting matrix K is symmetric and positive semidefinite. Then we are guaranteed that $\rho(\widehat{J}^{-1}K) < 1$. In particular, we have the bounds:

$$\frac{\lambda_{max}(K)}{\lambda_{max}(K) + \lambda_{max}(\widehat{J}_{\text{orig}})} \leq \rho(\widehat{J}^{-1}K) \leq \frac{\lambda_{max}(K)}{\lambda_{max}(K) + \lambda_{min}(\widehat{J}_{\text{orig}})} \tag{4.15}$$

Proof. First of all, since $\widehat{J}^{-1}$ and K are symmetric and positive semidefinite, we have

$$\lambda_{min}(\widehat{J}^{-1}K) \geq \lambda_{min}(\widehat{J}^{-1})\lambda_{min}(K)$$

so that the eigenvalues of $\widehat{J}^{-1}K$ are all non-negative. Therefore, the spectral radius $\rho(\widehat{J}^{-1}K)$ is given by the maximal eigenvalue. Equation (4.15) then follows from the bounds of Theorem 2.2 in Axelsson [10] on the maximal eigenvalue. □

Observe that the upper bound of equation (4.15) is always less than one for positive definite $\widehat{J}_{\text{orig}}$. Therefore, Theorem 4.3.1 gives sufficient conditions for the convergence of the ET algorithms. To illustrate Theorem 4.3.1, we consider the simple example of a single cycle.

Example 4.3.1 (Optimal tree for single cycle). Suppose that we have a Gaussian process with scalar variables at each node, defined on a graph $\mathcal{G}$ that is a single cycle. In this case, it suffices to cut a single edge in order to obtain a tree. Let $\mathbf{e}_u$ denote the vector of zeros with a single one at entry u. We consider a cutting matrix of the form

$$K = -P_{uv}^{-1}\left[\mathbf{e}_u\mathbf{e}_u^T + \mathbf{e}_v\mathbf{e}_v^T + \mathbf{e}_u\mathbf{e}_v^T + \mathbf{e}_v\mathbf{e}_u^T\right]$$

which corresponds to removing edge (u, v) from the graph. Note that the form of this cutting matrix is distinct from that illustrated in Figure 4.3; in particular, this cutting matrix also modifies the diagonal entries of the inverse covariance.

The matrix K is rank one, with only one non-zero eigenvalue $-2P_{uv}^{-1}$. We suppose that $P_{uv}^{-1} < 0$ for all edges (u, v), so that K is positive semidefinite, and Theorem 4.3.1 is applicable. To obtain an ET iteration that converges quickly, we would like to minimize the upper bound of equation (4.15). This corresponds to minimizing the largest eigenvalue of K. Consequently, for this single cycle case, removing the weakest edge (i.e., the edge with smallest $|P_{uv}^{-1}|$) from the graph leads to the best tree (in the sense of equation (4.15)). This finding agrees with the natural intuition.

A few remarks on Theorem 4.3.1 are in order. First of all, note that the hypotheses of the theorem require K to be positive semidefinite. Modifications to K so as to ensure positive semidefiniteness (e.g., adding multiples of the identity) are likely to increase the maximal eigenvalue $\lambda_{max}(K)$. As this maximal eigenvalue increases, the upper bound of equation (4.15) can become arbitrarily close to one. Thus, the theoretical convergence rate (at least the upper bound) can become extremely slow. In practice, we find that indefinite cutting matrices, as opposed to the positive semidefinite matrices required by the hypotheses of the theorem, typically lead to faster convergence.

Secondly, although the conditions of Theorem 4.3.1 are sufficient, they are by no means necessary to guarantee convergence of the ET algorithm. Even when cutting to a single tree, it is easy to construct examples in which the conditions of the theorem are not satisfied, but still $\widehat{J}$ is positive definite and $\rho(\widehat{J}^{-1}K) < 1$ so that the algorithm converges. A related caveat associated with Theorem 4.3.1 is its failure to address the superior performance typically achieved by cycling through several embedded trees.

Indeed, it is possible to combine two trees — a "good" tree for which $\rho(\widehat{J}_1^{-1}K_1)$ is small, and a "bad" tree for which $\rho(\widehat{J}_2^{-1}K_2)$ is large — such that the spectral radius of the combination $\rho(\widehat{J}_2^{-1}K_2\widehat{J}_1^{-1}K_1)$ is smaller than that of the good tree. We refer the interested reader to [163] for further examples of the benefits of cutting to multiple trees, as well as extensions to Theorem 4.3.1 that capture such effects.

■ 4.3.4 Calculation of error covariances

As described in Section 4.2.4, there exist a variety of iterative algorithms for computing the conditional mean of a linear-Gaussian problem. However, none of these methods correctly compute error covariances at each node. (See Schneider [156] for a Krylov subspace method that does compute error covariances, though under assumptions complementary to those of this chapter). We show here that the ET algorithm can efficiently compute these covariances in an iterative fashion. For many applications (e.g., oceanography [62]), obtaining these error statistics is equally as important as computing the conditional means. Indeed, in a statistical context, an estimate of the mean without any measure of uncertainty is arguably of limited use.

Assume for simplicity in notation that $\widehat{\mathbf{x}}^0 = 0$. We then expand the recursion of equation (4.8) for $n = 1, 2, 3, \ldots$ as follows:

$$\begin{aligned}
\widehat{\mathbf{x}}^1 &= \left[M_{i(1)}^{-1}\right] C^T R^{-1}\mathbf{y} \\
\widehat{\mathbf{x}}^2 &= \left[M_{i(2)}^{-1}K_{i(2)}M_{i(1)}^{-1} + M_{i(2)}^{-1}\right] C^T R^{-1}\mathbf{y} \\
\widehat{\mathbf{x}}^3 &= \left[M_{i(3)}^{-1}K_{i(3)}M_{i(2)}^{-1}K_{i(2)}M_{i(1)}^{-1} + M_{i(3)}^{-1}K_{i(3)}M_{i(2)}^{-1} + M_{i(3)}^{-1}\right] C^T R^{-1}\mathbf{y} \\
\vdots \quad &\vdots \qquad\qquad\qquad \vdots
\end{aligned}$$

From these equations, the general pattern can be discerned: for any iteration $n = 1, 2, \ldots$, we have:

$$\widehat{\mathbf{x}}^n \equiv \widehat{\mathbf{x}}^n(\mathbf{y}) = \left[\mathbf{F}^n + M_{i(n)}^{-1}\right] C^T R^{-1}\mathbf{y} \tag{4.16}$$

where the matrix $\mathbf{F}^n$ satisfies the recursion

$$\mathbf{F}^n = M_{i(n)}^{-1}K_{i(n)}\left[\mathbf{F}^{n-1} + M_{i(n-1)}^{-1}\right] \tag{4.17}$$

with the initial condition $\mathbf{F}^1 = \mathbf{0}$.

Let $\widehat{\mathbf{x}}(\mathbf{y})$ denote the correct conditional mean as a function of the data $\mathbf{y}$. By the data independence pointed out at the end of Section 4.3.2, if the recursion for the conditional means $\{\widehat{\mathbf{x}}^n(\mathbf{y})\}$, given in equation (4.8) is convergent for some data vector, then it converges to $\widehat{\mathbf{x}}(\mathbf{y})$ for all data vectors $\mathbf{y}$. Moreover, from equation (4.2a), we have

$$\widehat{\mathbf{x}}(\mathbf{y}) = \widehat{P}C^T R^{-1}\mathbf{y}$$

for all $\mathbf{y}$. Therefore, whenever the mean recursion converges, then the matrix sequence $\{\mathbf{F}^n + M_{i(n)}^{-1}\}$ converges to the full error covariance $\widehat{P}$.

Moreover, the cutting matrices K are typically of low rank, say $\mathcal{O}(c)$ where c is the number of cut edges. For example, given the edge set $\mathcal{E}(\mathcal{T})$ of some tree, the sparsest possible cutting matrix (i.e., one which does not modify the diagonal entries) can be written as

$$K = \sum_{(u,v)\in\mathcal{E}/\mathcal{E}(\mathcal{T})} w_{uv}\left[\mathbf{e}_u\mathbf{e}_v^T + \mathbf{e}_v\mathbf{e}_u^T\right] \tag{4.18}$$

where w_{uv} is a weight on edge (u, v). This cutting matrix is of rank (at most) $2c$.

With this type of low rank decomposition for K, it can be shown that each $\mathbf{F}^n$ can also be decomposed as a sum of $\mathcal{O}(c\,d)$ rank 1 matrices. Directly updating this low-rank decomposition of $\mathbf{F}^n$ from that of $\mathbf{F}^{n-1}$ requires $\mathcal{O}(d^5c^2N)$ operations. However, an efficient restructuring of this update requires only $\mathcal{O}(d^4cN)$ operations [see 163]. The diagonal blocks of the low-rank representation may be easily extracted and added to the diagonal blocks of $M_{i(n)}^{-1}$, which are computed by standard tree smoothers. All together, we may obtain these error variances in $\mathcal{O}(d^4cN)$ operations per iteration. Thus, the computation of error variances will be particularly efficient for graphs where the number of edges c that must be cut is small compared to the total number of nodes N.

Example 4.3.2 (Square grids). Consider a square grid with N nodes; the case $N = 5$ is illustrated in Figure 4.2(a). Place a single Gaussian random variable x_s at each node, thereby forming a random vector $\mathbf{x}$ of length N. It can be shown that the square grid has $2\sqrt{N}\,(\sqrt{N} - 1)$ edges in total. Any spanning tree on a graph with N nodes has $N - 1$ edges, so that we have to remove

$$c = 2\sqrt{N}\,(\sqrt{N} - 1) - [N - 1] = \left[\sqrt{N} - 1\right]^2$$

edges to form a tree. Asymptotically, $c \sim N$ so that the computational complexity of our error covariance computation for a square grid is $\mathcal{O}(N^2)$. This is inferior to the nested dissection method for matrix inversion [54], which has complexity $\mathcal{O}(N^{3/2})$. Nonetheless, there exist many graphs with less than $\mathcal{O}(\sqrt{N})$ additional edges (beyond those associated with a given spanning tree) for which our algorithm would lead to gains.

■ 4.3.5 Results

We have applied the ET algorithm to a variety of graphs, ranging from graphs with single cycles to densely connected MRFs on grids. Here we show some sample results; additional results on the empirical behavior of the ET algorithm are given in [163].

Figure 4.4(a) compares the rates of convergence for three algorithms: conjugate gradient (CG), embedded trees (ET), and belief propagation (BP) on a 20×20 nearest-neighbor grid. We made a random choice of the inverse covariance matrix P^{-1}, subject

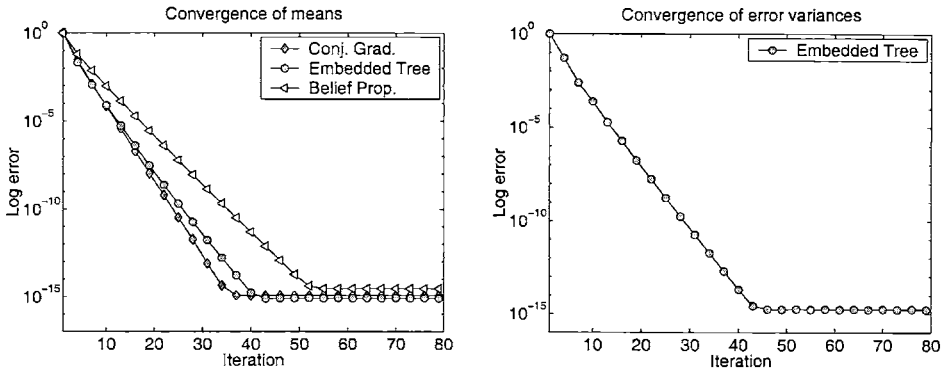


Figure 4.4. (a) Convergence rates for computing conditional means $\widehat{\mathbf{x}}$ (normalized L^2 error). Plot compares rates of ET to belief propagation (BP) and conjugate gradient (CG). (b) Convergence rate of ET algorithm for computing error variances.

to the constraint of being symmetric and positive definite. The measurement matrix C and noise covariance R were both chosen as the identity. The ET algorithm employed two embedded trees, one analogous to that shown in Figure 4.2(b) and the other a rotated version of this tree. We find that CG is usually fastest, and can exhibit supergeometric convergence. In accordance with Proposition 4.3.2, the ET algorithm converges geometrically. Either BP or ET can be made to converge faster, depending on the choice of clique potentials. However, we have not experimented with optimizing the performance of ET by adaptively choosing edges to cut. Figure 4.4(b) shows that in contrast to CG and BP, the ET algorithm can also be used to compute the conditional error variances, where the convergence rate is again geometric.

■ 4.4 Discussion

In this chapter, we developed the embedded trees algorithm for exact estimation of Gauss-Markov processes on graphs with cycles. Like structured mean field (see Section 2.3.1), this ET algorithm exploits the fact that exact computations can be performed efficiently for trees embedded within the graph with cycles. In contrast to mean field, the ET algorithm takes advantage of the fact that graphs with cycles have a (typically large) number of spanning trees. Indeed, although ET can be implemented using only a single spanning tree, its application is usually more powerful when it cycles through some set of embedded trees.

For computing means, the ET algorithm is comparable to other techniques. In contrast with other techniques, the ET algorithm also computes the correct covariances of the error in the estimate. The error covariance computation is especially efficient for graphs in which cutting a small number of edges reveals an embedded tree. Moreover, the ET algorithm suggests other ways in which embedded tree structures can be ex-

ploited: e.g., as preconditioners for the conjugate gradient method [54]. Extensions of this nature are discussed in more detail in [163], and also in Chapter 8 of this thesis.

Although the focus of this chapter was Gaussian processes, we shall see in the following chapter that similar concepts can be developed for discrete-valued processes. Indeed, the focus of Chapter 5 is tree-based reparameterization, which also entails performing (different) exact computations using distributions defined by embedded trees.

Chapter 5

Tree-based reparameterization for approximate estimation

5.1 Introduction

Given a distribution $p(\mathbf{x})$ defined by a graphical model, one important problem is computing marginal distributions of variables at each node on the graph. For tree-structured graphs, standard and highly efficient algorithms exist for this task; see Appendix A for description of one such algorithm. In contrast, exact solutions are prohibitively complex for more general graphs of any substantial size [39]. As a result, there has been considerable interest and effort aimed at developing approximate inference algorithms for large graphs with cycles.

Perhaps the best-known and most widely studied [e.g., 3, 130, 147, 173, 180] approximation method is that known variously as *belief propagation* in the graphical model community [137], and the *sum-product algorithm* in coding theory [e.g., 117, 130]. The interest in this algorithm has been fueled in part by its use in fields such as artificial intelligence and computer vision [e.g., 65, 68, 133], and also by the success of turbo codes and other compound codes, for which the decoding algorithm is a particular instantiation of belief propagation [e.g., 71, 117, 130]. While there are various equivalent forms for belief propagation [137], the best known formulation, which we refer to here as the BP algorithm, entails the exchange of statistical information among neighboring nodes via message-passing. If the graph is a tree, the resulting algorithm can be shown to produce exact solutions in a finite number of iterations. The message-passing formulation is thus equivalent to other techniques for optimal inference on trees, some of which involve more global and efficient computational procedures. On the other hand, if the graph contains cycles, then it is the local message-passing algorithm that is most generally applicable. It is well-known that the resulting algorithm may not converge; moreover, when it does converge, the quality of the resulting approximations varies substantially.

Recent work has yielded some insight into the dynamics and convergence properties of BP. For example, several researchers [2, 8, 106, 173] have analyzed the single cycle case, where belief propagation can be reformulated as a matrix powering method. For the special case of graphs corresponding to turbo codes, Richardson [147] developed a

geometric approach, through which he was able to establish the existence of fixed points, and give conditions for their stability. More recently, Yedidia et al. [180] showed that BP can be viewed as performing a constrained minimization of the so-called Bethe free energy associated with the graphical distribution,[1] which inspired other researchers [e.g., 175, 181] to develop more sophisticated algorithms for the minimization of the Bethe free energy. Yedidia et al. also proposed extensions to BP based on cluster variational methods [114]; related extensions using higher order structures have been proposed by Minka [131]. These advances notwithstanding, much remains to be understood about the behavior of this algorithm, and more generally about other (perhaps superior) approximation algorithms.

This important area constitutes the focus of this chapter. In particular, this chapter provides a new conceptual view of a large class of iterative algorithms that includes BP. Central to the framework presented here is the idea of performing exact computations over acyclic subgraphs embedded within a graph with cycles. This idea was exploited in Chapter 4 to develop an iterative algorithm for exact estimation of Gaussian processes on graphs. One of the motivations for the research presented in this chapter is to show how such tree-based updates can also be applied to discrete processes on graphs with cycles.

As discussed in Section 2.1, a key idea in graphical models is the representation of a probability distribution as a product of factors, each of which involves variables only at a subset of nodes corresponding to a clique of the graph. Such factorized representations are far from unique, which suggests the goal of seeking a *reparameterization* of the distribution consisting of factors that correspond, either exactly or approximately, to the desired marginal distributions. If the graph is cycle-free (i.e., a tree), then there exists a unique reparameterization specified by exact marginal distributions over cliques. Indeed, such a parameterization is the cornerstone of the junction tree representation (see Section 2.1.5).

For a graph with cycles, on the other hand, exact factorizations exposing these marginals do not generally exist. Nevertheless, it is always possible to reparameterize certain *portions* of any factorized representation — namely, any subset of factors corresponding to a cycle-free subgraph of the original graph. We are thus led to consider iterative reparameterization of different subsets, each corresponding to an acyclic subgraph. As we will show, BP can be interpreted in exactly this manner, in which each reparameterization takes place over the extremely simple subgraph consisting of a pair of neighboring nodes. One of the consequences of this interpretation is a more storage-efficient "message-free" implementation of BP.

More significantly, this interpretation suggests a more general class of updates where reparameterization is performed over arbitrary cycle-free subgraphs. Although the choice of cycle-free subgraphs is arbitrary, in this chapter we focus primarily on updates involving maximal cycle-free subgraphs — that is, spanning trees. We refer to

[1] Several researchers have investigated the utility of Bethe tree approximations for graphical models; we refer the reader to [e.g., 164, 178].

this class as *tree-based reparameterization* (TRP) algorithms. Since each update on a spanning tree propagates information globally to each node of the graph, one might expect a TRP algorithm to have better convergence properties than the purely local two-node updates of BP. Indeed, experimentation with TRP supports this conclusion: when applied to problems for which BP converges, TRP typically converges at least as quickly, and for many problems much more quickly. More importantly, we find that TRP converges in cases where BP does not.

At one level, the more global updates of TRP can be viewed as a schedule for message-passing based on spanning trees (though with a more efficient implementation via reparameterization). Indeed, one of the practical contributions of this chapter is to demonstrate that such tree-based updates have convergence properties superior to those of BP. At a more abstract level, however, the reparameterization perspective leads to a number of new conceptual insights, including a novel characterization of fixed points; and an invariance intrinsic to the TRP or BP algorithms. These two properties, when applied in conjunction, allow us to characterize the approximation error. Many of our results, though not obvious from the more traditional message-passing viewpoint, follow in a natural way from the reparameterization framework.

In the next section, we introduce the background and notation that underlies our development. In the process, we illustrate how distributions over cycle-free graphs can be reparameterized in terms of local marginal distributions. In Section 5.3, we introduce the class of TRP algorithms. In this context, it is convenient to represent distributions in an exponential form using an overcomplete basis. Our choice of an overcomplete basis, though unorthodox, makes the idea of reparameterization more transparent, and easily stated. In this section, we also show an equivalent formulation of BP as a sequence of local reparameterizations. Moreover, we present some experimental results illustrating the benefits of more global TRP updates, which include a greater range of problems for which convergence is obtained, as well as increased speed of convergence.

Section 5.4 contains analysis of the fixed points of the TRP updates, as well as the question of convergence. Central to the analysis is a geometric characterization of successive iterates, which reveals interesting links between tree-based reparameterization and successive projection algorithms for constrained minimization of Bregman distances [e.g., 30]. On this basis, we show that fixed points of the TRP algorithm satisfy the necessary conditions to be a local minimum of a certain cost function that arises as an approximation to the Kullback-Leibler divergence. This result allows us to make contact with the work of Yedidia et al. [180]. Specifically, we show that although the cost function minimized by our TRP algorithms is not the same as the Bethe free energy, TRP fixed points do coincide with extremal points of the Bethe free energy (i.e., with the fixed points of BP). An important benefit of our formulation is a new and intuitive characterization of the fixed points: in particular, any fixed point must be consistent, in a suitable sense to be defined, with respect to any acyclic subgraph; and at least one such fixed point of this type is guaranteed to exist. In addition, the geometric viewpoint also allows us to formulate sufficient conditions for convergence in the case of TRP using

two spanning trees.

A fundamental property of our reparameterization updates is that they leave invariant the distribution on the full graph. This result has a number of important consequences, which are also developed in Section 5.4. For example, by adapting this invariance to the Gaussian (as opposed to discrete) case, we obtain a short and elementary proof of a known result [152, 174] – namely, the means must be exact when TRP/BP converges.

The final topic of this chapter is the analysis of the approximation error arising from application of TRP or BP. Previous results on this error have been obtained in certain special cases. For a single cycle, Weiss [173] derived a relation between the exact marginals and the BP approximations, and for a binary processes showed how local corrections could be applied to compute the exact marginals. Empirically, he also observed that in certain cases, approximation accuracy is correlated with the convergence rate of BP. In the context of turbo decoding, Richardson [147] provided a heuristic analysis of the associated error. Despite these encouraging results, a deep and broadly applicable understanding of the approximation error remains a challenging and important problem. Our characterization of the TRP/BP fixed points, in conjunction with the invariance property, allows us to contribute to this goal by analyzing the approximation error for arbitrary graphs. In particular, our development in Section 5.5 begins with the derivation of an *exact* relation between the correct marginals and the approximate marginals computed by TRP or BP. We then exploit this exact relation to derive both upper and lower bounds on the approximation error. The interpretation of these bounds provides an understanding of the conditions that govern the performance of approximation techniques like TRP or BP. Moreover, using results from Chapter 7, these bounds are computable in polynomial time. We illustrate their performance on some sample problems. The chapter concludes in Section 5.6 with a summary.

■ 5.2 Estimation in graphical models

The focus of this chapter is the (approximate) computation of marginal distributions associated with a graph-structured distribution $p(\mathbf{x})$. In particular, the distribution $p(\mathbf{x})$ is defined by a product of compatibility functions ψ_C over the cliques of a graph $\mathcal{G}$, as in equation (2.3). Throughout this chapter, we shall assume that the clique set $\mathbf{C}$ of $\mathcal{G}$ consists only of singletons and edges (i.e., $\mathbf{C} = \mathcal{V} \cup \mathcal{E}$). At a purely formal level, this assumption entails no loss of generality since it is always possible to cluster the nodes of any graph so as to form an equivalent graph with a maximal clique size of two [e.g., 66]. However, modifying the graph so as to create pairwise cliques can lead to aggregated nodes with excessively high state cardinalities, so that clustering may not be helpful in a practical sense. Although we focus on the case of pairwise cliques, it is straightforward to extend our reparameterization approach to larger cliques, as in cluster variational methods [e.g., 114].

Under these assumptions, the prior distribution $p(\mathbf{x})$ is defined by a product of

singleton and edge terms as follow:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{s \in \mathcal{V}} \psi_s(x_s) \prod_{(s,t) \in \mathcal{E}} \psi_{st}(x_s, x_t) \tag{5.1}$$

Figure 5.1(a) gives an example of the assignment of pairwise compatibility functions ψ_{st} and single-node functions ψ_s. With a minor abuse of notation, for an $m-$state discrete process, the quantity ψ_{st} lying on the edges (s,t) can be thought of as a $m \times m$ matrix, where the (j,k) element $\psi_{st;jk}$ is equal to the function value of ψ_{st} for $\{x_s = j, x_t = k\}$. Similarly, the single node functions ψ_s can be thought of as an m-vector, where the j^{th} component $\psi_{s;j}$ equals the value of ψ_s for $\{x_s = j\}$.

The specific goal of this chapter is to (approximately) compute the marginal probabilities $P_s = p(x_s)$ of $p(\mathbf{x})$ at each node of the graph. For general graphs with cycles, this task requires summations involving exponentially many terms; indeed, it can be shown to be a NP-hard problem [39]. For tree-structured graphs, there exist direct algorithms for optimal estimation. For graphs with cycles, suboptimal algorithms (such as BP) are used in an attempt to compute approximations to the desired marginals. In the following sections, we elaborate on both of these topics.

■ 5.2.1 Exact estimation on trees as reparameterization

Algorithms for optimal estimation on trees have appeared in the literature of various fields, including coding theory [117], artificial intelligence [137], and system theory [14]. See Appendix A for a detailed derivation of one algorithm for optimal estimation on trees. As described in Section 2.1.5, such tree inference algorithms can, in principle, be applied to any graph by clustering nodes so as to form a *junction tree*. However, in many cases of interest, the aggregated nodes of the junction tree have exponentially large state cardinalities, meaning that applying tree algorithms is prohibitively complex. This explosion in the state cardinality is another demonstration of the intrinsic complexity of exact computations for graphs with cycles.

An important observation that arises from the junction tree perspective is that any exact algorithm for optimal estimation on trees actually computes marginal distributions for pairs (s,t) of neighboring nodes. In doing so, it produces an alternative factorization of the distribution $p(\mathbf{x})$, namely:

$$p(\mathbf{x}) = \prod_{s \in \mathcal{V}} P_s \prod_{(s,t) \in \mathcal{E}} \frac{P_{st}}{P_s \, P_t} \tag{5.2}$$

where $P_s = p(x_s)$ and $P_{st} = p(x_s, x_t)$. As an illustration, Figure 5.1(a) shows a simple example of a tree, labeled in terms of compatibility functions ψ_s and ψ_{st}, which leads to the factorization in equation (5.1). Figure 5.1(b) shows this same tree, now *reparameterized* in terms of the local marginal distributions P_s and P_{st}. The representation of equation (5.2) can be deduced from a more general factorization result on junction

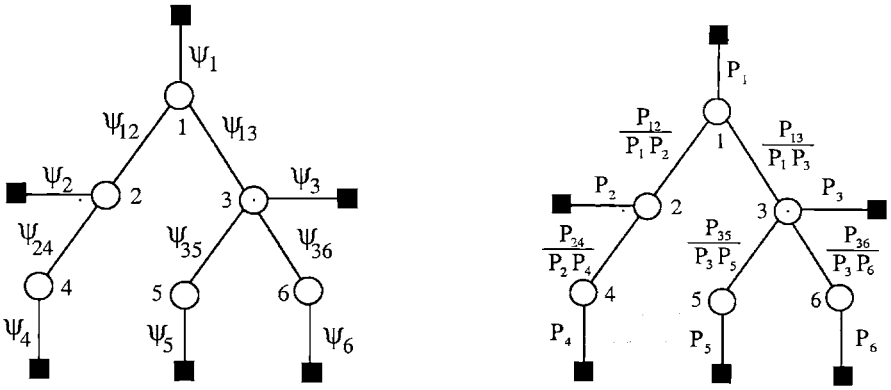


Figure 5.1. A simple example of a graphical model; circles correspond to state variables x_s, whereas squares correspond to observations. (a) Original parameterization of distribution $p(\mathbf{x}) = \frac{1}{Z} \prod_{s \in \mathcal{V}} \psi_s \prod_{(s,t) \in \mathcal{E}} \psi_{st}$ on the tree in terms of compatibility functions ψ_{st} and ψ_s. (b) Final parameterization $p(\mathbf{x}) = \prod_{s \in \mathcal{V}} P_s \prod_{(s,t) \in \mathcal{E}} \frac{P_{st}}{P_s P_t}$ in terms of marginal probabilities P_s and joint probabilities P_{st}.

trees [e.g. 101, 122].[2] We thus arrive at an alternative interpretation of exact inference on trees: it entails computing a reparameterized factorization of the distribution $p(\mathbf{x})$ that explicitly exposes the local marginal distributions; and also does not require any additional normalization (i.e., with partition function $Z = 1$).

■ 5.2.2 Belief propagation for graphs with cycles

As we have indicated, the message-passing form of belief propagation (BP), in addition to being exact in application to trees, yields an iterative message-passing algorithm for graphs with cycles. In this section, we summarize for future reference the equations governing the BP dynamics. The message passed from node s to node t, denoted by M_{st}, is an m-vector in which element $M_{st;k}$ gives its value when $x_t = k$. Let $\mathcal{N}(s) = \{t \in \mathcal{V} \mid (s,t) \in \mathcal{E}\}$ be the set of neighbors of s in $\mathcal{G}$. With this notation, the

[2] Alternatively, equation (5.2) can be seen as a symmetrized generalization of the well-known factorization(s) of Markov chains. For example, the variables at the three nodes $\{1, 2, 4\}$ in Figure 5.1(b) form a simple Markov chain, meaning that the joint distribution can be written as

$$\begin{aligned} P_{124} &= P_1 \, (P_{2|1})(P_{4|2}) \\ &= P_1 \, (P_{12}/P_1)(P_{24}/P_2) \\ &= P_1 P_2 P_4 \, (P_{12}/P_1 P_2)(P_{24}/P_2 P_4) \end{aligned}$$

where the last equality is precisely the form of equation (5.2). Note that the final line removes the asymmetry present in those that precede it (which resulted from beginning the factorization from node 1, as opposed to node 2 or 4).

message at iteration $(n+1)$ is updated based on the messages at the previous iteration n as follows:

$$M_{st;k}^{n+1} = \kappa \sum_{j=0}^{m-1} \psi_{st;jk}\psi_{s;j} \prod_{u \in \mathcal{N}(s)/t} M_{us;j}^{n} \tag{5.3}$$

where κ denotes a normalization constant.[3] At any iteration, the "beliefs" — that is, approximations to the marginal distributions — are given by

$$B_{s;j}^{n} = \kappa\, \psi_{s;j} \prod_{u \in \mathcal{N}(s)} M_{us;j}^{n} \tag{5.4}$$

■ 5.3 Tree-based reparameterization framework

Key to TRP is the concept of an *embedded tree* within an arbitrary graph $\mathcal{G}$ with cycles — i.e., a tree formed by removing edges from the graph. A *spanning tree* is an embedded tree that connects all nodes of the original graph. Figure 5.2 illustrates

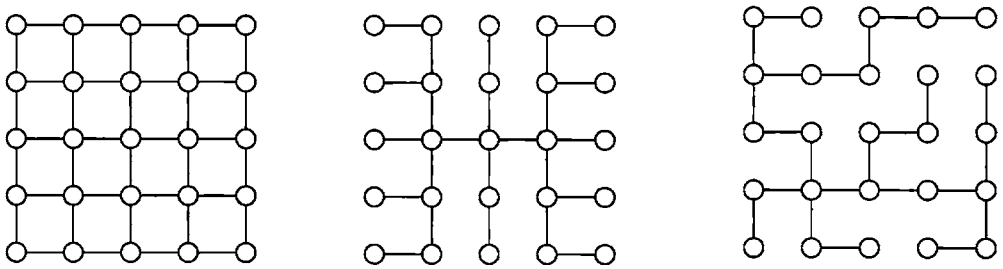


Figure 5.2. Illustration of spanning trees. (a) Original graph is a nearest neighbor grid. (b) One particular choice of spanning tree for the grid. (c) Another spanning tree.

these definitions: panel (a) shows a nearest neighbor grid, whereas panels (b) and (c) illustrate embedded spanning trees. Of course, these are just two examples of such embedded spanning trees. Indeed, a graph generally has a (large) number of spanning trees, and we exploit this fact in our work. Specifically, suppose that $\mathcal{T}^0, \ldots, \mathcal{T}^{L-1}$ (with corresponding edge sets $\mathcal{E}^0, \ldots, \mathcal{E}^{L-1} \subset \mathcal{E}$) is a given set of spanning trees for the graph $\mathcal{G}$. Then for any $i \in \{0, \ldots, L-1\}$, the distribution $p(\mathbf{x})$ of a stochastic process over $\mathcal{G}$ (as in equation (5.1)) can be factored as:

$$p(\mathbf{x}) = p^i(\mathbf{x})\, r^i(\mathbf{x}) \tag{5.5}$$

[3]Throughout this paper, we will use κ to refer to an arbitrary normalization constant, the definition of which may change from line to line. In all cases, it is easy to determine κ by local calculations.

where $p^i(\mathbf{x})$ includes the factors in equation (5.1) corresponding to cliques of $\mathcal{T}^i$, and $r^i(\mathbf{x})$ absorbs the remaining terms, corresponding to edges in $\mathcal{E}/\mathcal{E}^i$ removed to form $\mathcal{T}^i$.

Because $\mathcal{T}^i$ is a tree, the reparameterization operation in equation (5.2) can be applied to $p^i(\mathbf{x})$ in order to obtain an alternative factorization of the distribution $p^i(\mathbf{x})$. With reference to the full graph $\mathcal{G}$ and distribution $p(\mathbf{x})$, this operation simply modifies the compatibility functions for cliques in $\mathcal{G}$, but without modifying the actual distribution $p(\mathbf{x})$. In a subsequent update using this new set of functions and choosing a *different* tree $\mathcal{T}^j$, we can write $p(\mathbf{x}) = p^j(\mathbf{x})r^j(\mathbf{x})$, where $p^j(\mathbf{x})$ includes compatibility functions over cliques in $\mathcal{T}^j$. We can then perform reparameterization for $p^j(\mathbf{x})$, and repeat the process, choosing one of the $\mathcal{T}^i$ at each step of the iteration.

Figure 5.3 illustrates the basic steps of this procedure for a simple graph with cycles. Panel (a) shows the original parameterization of $p(\mathbf{x})$ in terms of compatibility functions ψ_s and ψ_{st}, as in equation (2.3). A spanning tree, formed by removing edges $(4,5)$ and $(5,6)$, is shown in panel (b): that is, $r^i(\mathbf{x}) = \psi_{45}\,\psi_{56}$ in this case. The tree

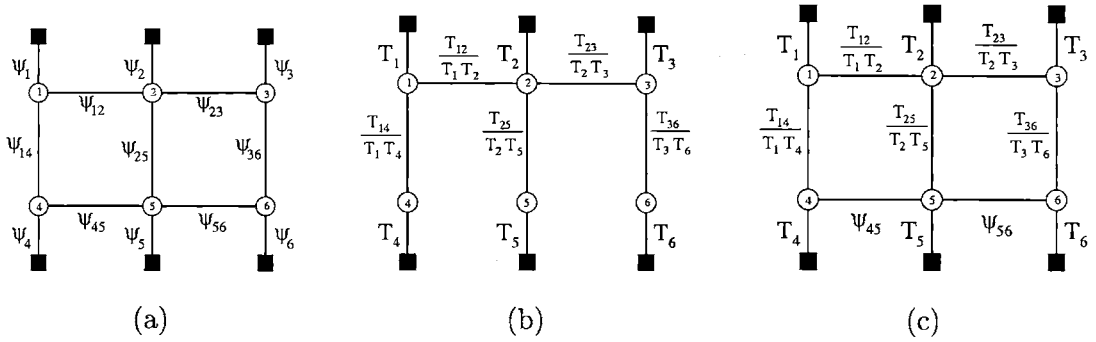


Figure 5.3. Illustration of TRP update. (a) Original parameterization in terms of compatibility functions ψ_s and ψ_{st}. (b) Tree reparameterization update on the spanning tree. (c) New parameterization after a single iteration.

distribution $p^i(\mathbf{x})$, corresponding to the product of all the other compatibility functions, is reparameterized in terms of marginals T_s and T_{st} computed from the tree $\mathcal{T}^i$. The quantities $\{T_s, T_{st}\}$ are exact marginals for the tree, but represent approximations to the actual marginals $\{P_s, P_{st}\}$ of the graph with cycles. The graph compatibility functions after this first update are shown in panel (c). In a subsequent update, a different tree is chosen over which reparameterization is to be performed.

As should be clear from the preceding discussion, each step of the algorithm[4] reparameterizes the density $p(\mathbf{x})$ but *does not* modify it (aside from the normalization constant). To formalize this basic idea, in this section we introduce a particular exponential parameterization of distributions $p(\mathbf{x};\theta)$, such that iterations of the type just described can be represented as explicit functional updates $\theta^n \mapsto \theta^{n+1}$ on these parameters. We also show that BP iterations can be interpreted as reparameterization operations using

[4]Here we have described an unrelaxed form of the updates; in the sequel, we present and analyze a suitably relaxed formulation.

especially simple non-spanning embedded trees, and we present experimental results illustrating the potential advantages of TRP over BP.

■ 5.3.1 Exponential families of distributions

Recall from Section 2.2 the definition of an exponential family of distributions:

$$p(\mathbf{x};\theta) = \exp\Big\{\sum_{\alpha}\theta_\alpha\phi_\alpha(\mathbf{x}) - \Phi(\theta)\Big\} \tag{5.6a}$$

$$\Phi(\theta) = \log\Big(\sum_{\mathbf{x}\in\mathcal{X}^N}\exp\{\sum_{\alpha}\theta_\alpha\phi_\alpha(\mathbf{x})\}\Big) \tag{5.6b}$$

where Φ is the *log partition function* that normalizes the distribution.

It is standard to specify an exponential family of the form in equation (5.6a) using a set of functions $\boldsymbol{\phi} = \{\phi_\alpha \mid \alpha \in \mathcal{A}\}$ that are linearly independent. This gives rise to a so-called minimal representation [e.g., 13]. However, in this chapter, we will find it convenient to use a non-minimal set of functions. Specifically, let $s,t \in \mathcal{V}$ be indices parameterizing the nodes of the graph, and let the indices j, k run over the m possible states of the discrete random variables. We then take the index set for α, denoted by $\mathcal{A}$, to be the set of pairs $(s;j)$ or 4-tuples $(st;jk)$, and choose the potentials ϕ_α as indicator functions for x to take on the indicated value (or values) at the indicated node (or pair of nodes). That is,

$$\phi_\alpha(\mathbf{x}) = \delta(x_s = j) \quad \text{for } \alpha = (s;j) \tag{5.7a}$$

$$\phi_\alpha(\mathbf{x}) = \delta(x_s = j)\delta(x_t = k) \quad \text{for } \alpha = (st;jk) \tag{5.7b}$$

Here, the indicator or delta function $\delta(x_s = j)$ is equal to 1 when node x_s takes the state value j, and 0 otherwise. With this choice of $\{\phi_\alpha\}$, the length of θ is given by

$$d(\theta) = mN + m^2|\mathcal{E}| \tag{5.8}$$

In contrast to a minimal representation, the exponential parameterization of equation (5.7) is *overcomplete* (i.e., there are linear relations among the functions $\{\phi_\alpha\}$).[5] As an example, for any edge $(s,t) \in \mathcal{E}$, we have the linear dependence

$$\sum_{j=0}^{m-1}\delta(x_s = j)\delta(x_t = k) = \delta(x_t = k) \qquad \text{for all} \quad k = 0,\ldots,m-1$$

An important consequence of overcompleteness is the existence of distinct parameter vectors $\theta \neq \theta^*$ that induce the same distribution (i.e., $p(\mathbf{x};\theta) = p(\mathbf{x},\theta^*)$). This many-to-one correspondence between parameters and distributions is of paramount importance to our analysis because it permits reparameterization operations that leave the overall distribution unchanged.

[5]There are also nonlinear relations, but it is the linear dependencies that distinguish between minimal and overcomplete representations.

5.3.2 Basic operators

Given a distribution $p(\mathbf{x};\theta)$ defined by a graph $\mathcal{G}$, the quantities that we wish to compute are elements of the *marginal probability vector*

$$P = \{ P_s \mid s \in \mathcal{V} \} \cup \{ P_{st} \mid (s,t) \in \mathcal{E} \} \tag{5.9}$$

where $P_{s;j} = p(x_s = j;\theta)$ defines the elements of the single-node marginal P_s; and $P_{st;jk} = p(x_s = j,\ x_t = k;\ \theta)$ defines the elements of the pairwise marginal P_{st}.

We now observe that elements of the marginal probability vector P arise as expectations under $p(\mathbf{x};\theta)$ of the potential functions $\{\phi_\alpha\}$ defined in equation (5.7) — viz.:

$$P_{s;j} = \mathbb{E}_\theta[\delta(x_s = j)] \tag{5.10a}$$

$$P_{st;jk} = \mathbb{E}_\theta[\delta(x_s = j)\,\delta(x_t = k)] \tag{5.10b}$$

On this basis, we conclude that P constitutes a set of *mean parameters* dual to the exponential parameters θ. These two parameters are coupled via the relation:

$$P = \Lambda(\theta) \tag{5.11}$$

where Λ is the Legendre transform. (See Section 2.2.4 for more information about the Legendre transform and its properties). Therefore, the vector P can be viewed as an alternative set of parameters for the distribution $p(\mathbf{x};\theta)$.

Note that the range of Λ, denoted $Ra(\Lambda)$, is a highly constrained set of vectors. First of all, any $T \in Ra(\Lambda)$ must belong to the unit hypercube $(0,1)^{d(\theta)}$. Secondly, there are normalization constraints (single-node and joint marginal probabilities must sum to one); and marginalization constraints (pairwise joint distributions, when marginalized, must be consistent with the single node marginals). That is, $Ra(\Lambda) \subseteq \mathbb{C}$, where

$$\mathbb{C} = \left\{ T \;\middle|\; T \in (0,1)^{d(\theta)};\ \sum_j T_{s;j} = 1 \text{ for } s \in \mathcal{V};\ \sum_k T_{st;jk} = T_{s;j} \text{ for } (s,t) \in \mathcal{E} \right\} \tag{5.12}$$

The elements of $T \in \mathbb{C}$ define a *locally consistent* set of pairwise and single node marginal distributions on the graph. When $\mathcal{G}$ is a tree, then any $T \in \mathbb{C}$ can be extended (via the tree factorization of equation (5.2)) to a unique distribution $p(\mathbf{x};\theta)$ such that $T = \Lambda(\theta)$. Thus, for tree-structured graphs, $Ra(\Lambda) = \mathbb{C}$.

For a graph with cycles, in contrast, $Ra(\Lambda)$ is strictly contained within $\mathbb{C}$. Indeed, for graphs with cycles, there exist elements of $\mathbb{C}$ that cannot be realized as the marginals of any distribution (Markov or otherwise). This strict containment reflects the fact that for a graph with cycles, the local consistency conditions defining $\mathbb{C}$ are no longer sufficient to guarantee the existence of a globally consistent distribution.

For a general graph with cycles, of course, the computation of $\Lambda(\theta)$ in equation (5.11) is very difficult. Indeed, algorithms like BP and TRP can be formulated as iteratively

generating approximations to $\Lambda(\theta)$. To make a sharp distinction from exact marginal vectors $P \in Ra(\Lambda) \subseteq \mathbb{C}$, we use the symbol T to denote such *pseudomarginal vectors* — i.e., vectors that belong the unit hypercube $(0,1)^{d(\theta)}$, but need not satisfy the marginal constraints (i.e., $\sum_k T_{st;jk} = T_{s;j}$) required for membership in $\mathbb{C}$.

We will also make use of the following mapping that is defined for any such T:

$$\left[\Theta(T)\right]_\alpha = \begin{cases} \log T_{s;j} & \text{if } \alpha = (s;j) \in \mathcal{A} \\ \log\left[T_{st;jk}/(\sum_j T_{st;jk})(\sum_k T_{st;jk})\right] & \text{if } \alpha = (st;jk) \in \mathcal{A} \end{cases} \tag{5.13}$$

The quantity $\Theta(T)$ can be viewed as an exponential parameter vector that indexes a distribution $p(\mathbf{x};\Theta(T))$ on the graph $\mathcal{G}$. In fact, consider a marginal vector $P \in Ra(\Lambda)$. If $\mathcal{G}$ is a tree, then not only is the computation of (5.11) simple, but we are also guaranteed $\Theta(P)$ indexes the same graphical distribution as that corresponding to the marginal vector P — that is:

$$\Lambda(\Theta(P)) = P \tag{5.14}$$

This equality is simply a restatement of the factorization of equation (5.2) for any tree-structured distribution in terms of its single-node and joint pairwise marginals. However, if $\mathcal{G}$ has cycles, then in general the marginal distributions of $p(\mathbf{x};\Theta(P))$ need not agree with the original marginals P (i.e., the equality of equation (5.14) does not hold). In fact, determining the exponential parameters corresponding to P for a graph with cycles is as difficult as the computation of $\Lambda(\theta)$ in equation (5.11). Thus, the composition of operators $\Lambda \circ \Theta$, mapping one marginal vector to another, is the identity for trees but not for general graphs.

Alternatively, we can consider composing Θ and Λ in the other order:

$$\mathcal{R}(\theta) = (\Theta \circ \Lambda)(\theta) \tag{5.15}$$

which defines a mapping from one exponential parameter vector to another. For a general graph, the operator $\mathcal{R}$ will alter the distribution (that is, $p(\mathbf{x};\theta) \neq p(\mathbf{x};\mathcal{R}(\theta))$). For a tree-structured graph, while $\mathcal{R}$ is not the identity mapping, it does leave the probability distribution unchanged; indeed, applying $\mathcal{R}$ corresponds to shifting from the original parameterization of the tree distribution in terms of θ to a new exponential parameter $\mathcal{R}(\theta)$ that corresponds directly to the factorization of equation (5.2). As a result, in application to trees, the operator $\mathcal{R}$ is idempotent (i.e., $\mathcal{R} \circ \mathcal{R} = \mathcal{R}$).

■ 5.3.3 Tree-based reparameterization updates

The basic idea of TRP is to perform reparameterization updates on a set of spanning trees $\mathcal{T}^0, \ldots, \mathcal{T}^{L-1}$ in succession. The update on any given spanning tree $\mathcal{T}^i$ involves only a subset $\mathcal{A}^i = \{(s;j),\ (st;jk) \mid s \in \mathcal{V},\ (s,t) \in \mathcal{E}^i\}$ of all the elements of θ. To move back and forth between parameter vectors on the full graph and those on spanning tree

$\mathcal{T}^i$, we define projection and injection operators

$$\Pi^i(\theta) = \{\theta_\alpha \mid \alpha \in \mathcal{A}^i\} \tag{5.16a}$$

$$\mathcal{I}^i(\Pi^i(\theta)) = \begin{cases} \theta_\alpha & \text{if } \alpha \in \mathcal{A}^i \\ 0 & \text{if } \alpha \notin \mathcal{A}^i \end{cases} \tag{5.16b}$$

We let Λ^i, Θ^i and $\mathcal{R}^i$ denote operators analogous to those in equations (5.11), (5.13) and (5.15) respectively, but as defined for $\mathcal{T}^i$.

Each TRP update acts on the full-dimensional vector θ, but changes only the lower-dimensional subvector $\Pi^i(\theta) = \{\theta_\alpha \mid \alpha \in \mathcal{A}^i\}$. For this reason, it is convenient to use the underbar notation to define operators of the following type:

$$\underline{\mathcal{R}}^i(\theta) = \mathcal{I}^i(\mathcal{R}^i(\Pi^i(\theta))) \tag{5.17a}$$

$$\underline{\Lambda}^i(\theta) = \mathcal{I}^i(\Lambda^i(\Pi^i(\theta))) \tag{5.17b}$$

For instance, $\underline{\Lambda}^i$ projects the exponential parameter vector θ onto spanning tree $\mathcal{T}^i$; computes the corresponding marginal vector for the distribution $p(\mathbf{x}; \Pi^i(\theta))$ induced on the tree; and then injects back to the higher dimensional space by inserting zeroes for elements of edges not in $\mathcal{T}^i$ (i.e., for indices $\alpha \in \mathcal{A}/\mathcal{A}^i$). Moreover, analogous to $\mathbb{C}$, we define a constraint set $\mathbb{C}^i$ by imposing marginalization constraints only for edges in the spanning tree (i.e., as in equation (5.12) with $\mathcal{E}$ replaced by $\mathcal{E}^i$). Note that $\mathbb{C}^i \supseteq \mathbb{C}$, and since every edge is included in at least one spanning tree, we have that $\cap_i \mathbb{C}^i = \mathbb{C}$.

Using this notation, the operation of performing tree-reparameterization on spanning tree $\mathcal{T}^i$ can be written compactly as transforming a parameter vector θ into the new vector given by:

$$\mathcal{Q}^i(\theta) = \underline{\mathcal{R}}^i(\theta) + \left[I - \mathcal{I}^i \circ \Pi^i\right](\theta) \tag{5.18a}$$

$$= \theta + \left[\underline{\mathcal{R}}^i(\theta) - \mathcal{I}^i(\Pi^i(\theta))\right] \tag{5.18b}$$

where I is the identity operator. The two terms in equation (5.18a) parallel the decomposition of equation (5.5): namely, the operator $\underline{\mathcal{R}}^i$ performs reparameterization of the distribution $p^i(\mathbf{x})$, whereas the operator $[I - \mathcal{I}^i \circ \Pi^i]$ corresponds to leaving the residual term $r^i(\mathbf{x})$ unchanged. Thus, equation (5.18a) is a precise statement of a spanning tree update (as illustrated in Figure 5.3), specified in terms of the exponential parameter θ.

Given a parameter vector θ, computing $\mathcal{Q}^i(\theta)$ is straightforward, since it only involves operations on the spanning tree $\mathcal{T}^i$. The tree-based reparameterization algorithm generates a sequence of parameter vectors $\{\theta^n\}$ by successive application of these operators $\mathcal{Q}^i$. The sequence is initialized[6] at θ^0 using the original set of graph functions $\{\psi_s\}$ and $\{\psi_{st}\}$ as follows:

$$\theta^0_\alpha = \begin{cases} \log \psi_{s;j} & \text{if } \alpha = (s;j) \\ \log \psi_{st;jk} & \text{if } \alpha = (st;jk) \end{cases}$$

[6]Other initializations are also possible. More generally, θ^0 can be chosen as any exponential parameter that induces the same distribution as the original compatibility functions $\{\psi_s\}$ and $\{\psi_{st}\}$.

At each iteration n, we choose some spanning tree index $i(n)$ from $\{0, \ldots, L-1\}$, and then update using the operator on spanning tree $\mathcal{T}^i$:

$$\theta^{n+1} = \mathcal{Q}^{i(n)}(\theta^n) \tag{5.19}$$

In the sequel, we will also consider a relaxed iteration, involving a step size $\lambda^n \in (0, 1]$ for each iteration:

$$\theta^{n+1} = \lambda^n \mathcal{Q}^{i(n)}(\theta^n) + (1 - \lambda^n)\theta^n \tag{5.20}$$

where $\lambda^n = 1$ recovers the unrelaxed version.

The only restriction that we impose on the set of spanning trees is that each edge of the full graph $\mathcal{G}$ is included in at least one spanning tree (i.e., $\cup_i \mathcal{A}^i = \mathcal{A}$). It is also necessary to specify an order in which to apply the spanning trees — that is, how to choose the index $i(n)$. A natural choice is the *cyclic ordering*, in which we set $i(n) \equiv n \pmod{L}$. More generally, any ordering — possibly random — in which each spanning tree occurs infinitely often is acceptable. A variety of possible orderings for successive projection algorithms are discussed in [30].

■ 5.3.4 Belief propagation as reparameterization

In this section, we show that BP can be reformulated in a message-free manner as a sequence of local rather than global reparameterization operations. Specifically, in each step, new compatibility functions are determined by performing exact calculations over extremely simple (non-spanning) trees formed of two nodes and the corresponding edge joining them.

We denote by M_{st}^0 the $m-$vector corresponding to the chosen initialization of the messages. This choice is often the vector of all ones, but any initialization with strictly positive components is permissible. The message-free version of BP iteratively updates approximations to the exact marginals $P = \{P_s, P_{st}\}$. Initial values of the approximations $T = \{T_s, T_{st}\}$ are determined from the initial messages M_{st}^0 and the original compatibility functions of the graphical model as follows:

$$T_s^0 = \kappa\, \psi_s \prod_{u \in \mathcal{N}(s)} M_{us}^0 \quad \text{for all } s \in \mathcal{V} \tag{5.21a}$$

$$T_{st}^0 = \kappa\, \psi_{st}\psi_s\psi_t \prod_{u \in \mathcal{N}(s)/t} M_{us}^0 \prod_{u \in \mathcal{N}(t)/s} M_{ut}^0 \quad \text{for all } (s,t) \in \mathcal{E} \tag{5.21b}$$

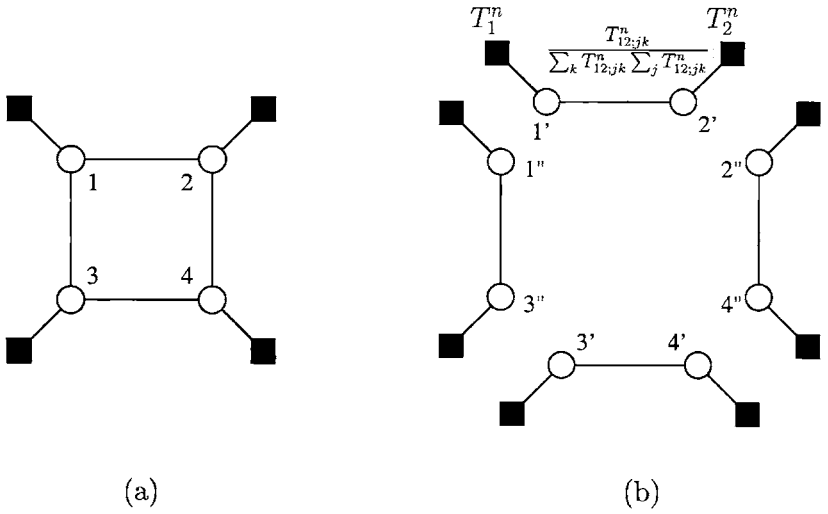
where κ denotes a normalization factor.

At iteration n, these quantities are updated according to the following recursions:

$$T_{s;j}^n = \kappa\, T_{s;j}^{n-1} \prod_{t \in \mathcal{N}(s)} \frac{1}{T_{s;j}^{n-1}} \sum_{k=0}^{m-1} T_{st;jk}^{n-1} \tag{5.22a}$$

$$T_{st;jk}^n = \kappa \frac{T_{st;jk}^{n-1}}{\left(\sum_{j=0}^{m-1} T_{st;jk}^{n-1}\right)\left(\sum_{k=0}^{m-1} T_{st;jk}^{n-1}\right)} T_{s;j}^n T_{t;k}^n \tag{5.22b}$$

The update in equation (5.22b) is especially noteworthy: it corresponds to performing optimal estimation on the very simple two-node tree formed by edge (s,t). As an illustration, Figure 5.4(b) shows the decomposition of a single-cycle graph into such two-node trees. This simple reparameterization algorithm operates by performing optimal estimation on this set of non-spanning trees, one for each edge in the graph, as in equation (5.22b). The single-node marginals from each such tree are merged via equation (5.22a).



(a) (b)

Figure 5.4. (a) Toy example of original graph. (b) Two node trees used for updates in message-free version of belief propagation. Computations are performed exactly on each two-node tree formed by a single edge and the two associated observation potentials as in equation (5.22b). The node marginals from each two-node tree are merged via equation (5.22a).

We now claim that this reparameterization algorithm is equivalent to belief propagation, summarizing the result as follows:

Proposition 5.3.1. The reparameterization algorithm specified by equations (5.21) and (5.22) is equivalent to the message-passing form of BP given in equations (5.3) and (5.4). In particular, for each iteration $n = 0, 1, \ldots$ and initial message vector M_{st}^0, we have the following relations:

$$M_{st;k}^{n+1} = \kappa \, M_{st;k}^0 \prod_{i=0}^{n} \frac{1}{T_{t;k}^i} \sum_{j=0}^{m-1} T_{st;jk}^i \quad \text{for all } (s,t) \in \mathcal{E} \tag{5.23a}$$

$$B_{s;j}^n = T_{s;j}^n \quad \text{for all } s \in \mathcal{V} \tag{5.23b}$$

where κ denotes a normalization factor.

Proof. See Appendix C.1. □

Despite the equivalence of this alternative form of BP with the message-passing version, the two schemes differ in their implementation; in particular, the reparameterization form permits in-place computation wherein new potentials overwrite the old ones. As a consequence, the reparameterization form requires only half of the storage used by the standard message-passing updates.

■ 5.3.5 Empirical comparisons of BP versus TRP

Given the more global nature of a TRP iteration (in contrast with the local updates of BP), one might expect TRP to have superior convergence properties. Indeed, this has proven to be the case in various experiments that we have performed. In this section, we present results from simulations of a binary process ($m = 2$) on three different graphs: a dense 5-node graph, a 15-node single cycle, and a 7×7 grid.

Convergence rates

At first sight, the more global nature of TRP might suggest that each TRP iteration is more complex computationally than the corresponding BP iteration. In fact, the opposite statement is true. Each TRP update involves $\mathcal{O}(m^2(N-1))$ operations, whereas each iteration of BP requires $\mathcal{O}(m^2|\mathcal{E}|)$ operations, where $|\mathcal{E}| \geq N-1$ is the number of edges in the graph. Consequently, each TRP iteration is slightly cheaper than a BP iteration for tree-like graphs (e.g., a single cycle); and considerably cheaper for denser graphs (such as the grid used in this section) where $|\mathcal{E}| \gg N$. Therefore, in order to make comparisons fair in terms of actual computation required, whenever we report iteration numbers, they are rescaled in terms of relative cost (i.e., for each graph, TRP iterations are rescaled by the ratio $(N-1)/|\mathcal{E}| < 1$).

For each graph, we performed simulations under three conditions: edge potentials that are *repulsive* (i.e., that encourage neighboring nodes to take opposite values); *attractive* (that encourage neighbors to take the same value); and *mixed* (in which some potentials are attractive, while others are repulsive). For each of these experimental conditions, each run involved a random selection of the initial parameter vector θ^0 defining the distribution $p(\mathbf{x}; \theta^0)$. In all experiments reported here, we generated the single node parameters $\theta_{s;j}$ as follows:[7] for each node $s \in \mathcal{V}$, sample $a_s \sim \mathcal{N}(0, (0.25)^2)$, and set $[\theta_{s;0} \;\; \theta_{s;1}] = [a_s \;\; -a_s]$. To generate the edge potential components $\theta_{st;jk}$, we began by sampling $b_{st} \sim \mathcal{N}(0,1)$ for each edge (s,t), With δ_{jk} denoting the Kronecker delta for j,k, we set the edge potential components in one of three ways depending on the experimental condition:

(a) repulsive condition: $\theta_{st;jk} = -(2\delta_{jk} - 1)\,|b_{st}|$.

(b) attractive condition: $\theta_{st;jk} = (2\delta_{jk} - 1)\,|b_{st}|$.

(c) mixed condition: $\theta_{st;jk} = (2\delta_{jk} - 1)\,b_{st}$.

[7]The notation $\mathcal{N}(0, \sigma^2)$ denotes a zero-mean Gaussian with variance σ^2.

For each graph and experimental condition, we ran a total of 500 trials, comparing the performance of TRP to BP. On any given run, an algorithm was deemed to converge when the mean L^2 difference between successive node elements $(\frac{1}{n}\sum_s \|\theta_s^{n+1} - \theta_s^n\|^2)$ reached a threshold[8] of $\epsilon = 1 \times 10^{-16}$. A run in which which a given algorithm failed

Graph	Single 15-cycle						7×7 grid					
	R		A		M		R		A		M	
BP	500	23.2	500	23.4	500	23.6	455	62.3	457	65.8	267	310.1
TRP	500	8.1	500	8.0	500	8.2	500	30.5	500	30.8	282	103.2

Table 5.1. Comparison of convergence behavior of TRP versus BP for a single cycle of 15 nodes; and a 7×7 grid. Potentials were chosen randomly in each of the three conditions: repulsive potentials (R); attractive potentials (A); mixed potentials (M). First and second numbers in each box denote the number of convergent runs out of 500; and the mean number of iterations (rescaled by relative cost and computed using only runs where both TRP and BP converged) respectively.

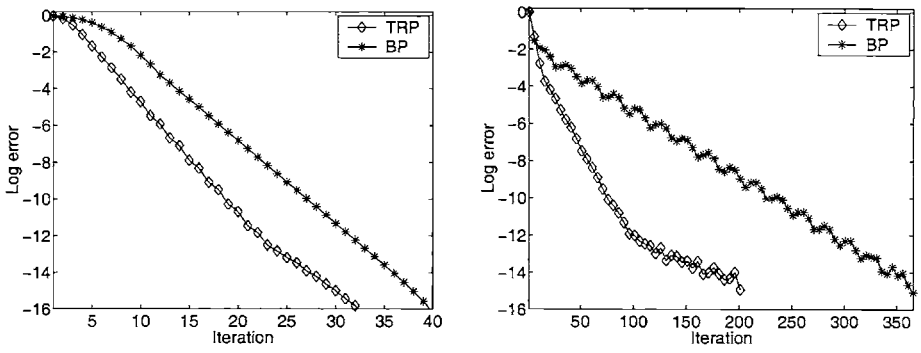
to reach this threshold within 3000 iterations was classified as a failure to converge. In each condition, we report the total number of convergent trials (out of 500); and also the mean number of iterations required to converge, rescaled by the ratio $(N-1)/|\mathcal{E}|$ and based only on trials where both TRP and BP converged.

Table 5.1 shows some summary statistics for the two graphs used in these experiments. For the single cycle, we implemented TRP with two spanning trees, whereas we used four spanning trees for the grid. Although both algorithms converged on all trials for the single cycle, the rate of TRP convergence was significantly (roughly 3 times) faster. The superiority of TRP is easily understandable in the single cycle case. A single iteration of TRP suffices to transmit information from each node to every other node in the graph, whereas the local updates of BP will require as many iterations as the graph diameter ($\lfloor N/2 \rfloor$ for a cycle of N nodes).

For the grid, algorithm behavior depends more on the experimental condition. The repulsive and attractive conditions are relatively easy, though still difficult enough for BP that it failed to converge on roughly 10% of the trials, in contrast to the perfect convergence percentage of TRP. In terms of mean convergence rates, TRP converged more than twice as quickly as BP. The mixed condition is difficult for suitably strong edge potentials on a grid: in this case both algorithms failed to converge on almost half the trials, although TRP converged more frequently than BP. Moreover, on runs where both algorithms converged, the TRP mean rate of convergence was three times faster than BP.

Figure 5.5 compares the convergence behavior of BP and TRP. Plotted is the log error between the single-node elements of θ^n and θ^* at each iteration, where θ^* is a fixed point common to BP and TRP, versus the iteration number. Panel (a) illustrates the

[8]This value was chosen by examining the behavior of each algorithm, and the successive differences in iterate values over a number of runs.



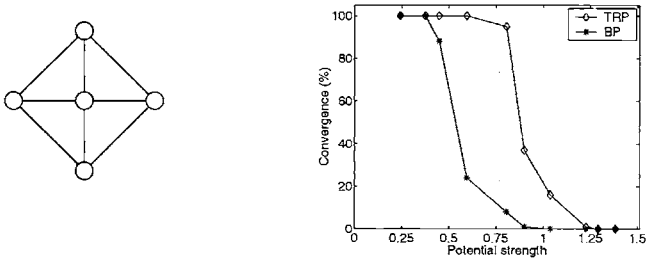
(a) Attractive potentials (b) Mixed potentials

Figure 5.5. Convergence plots of log error ($\log \sum_{s,j} \|\theta^n_{s;j} - \theta^*_{s;j}\|^2$) versus iteration number n for the 7×7 grid under two conditions. Here both BP and TRP converge to the same fixed point θ^*.

attractive condition, in which the error in TRP begins decreasing earlier than BP, with the asymptotic rate being similar. Examples in the repulsive condition show similar convergence behavior. On the other hand, in the mixed condition shown in (b), the global updates of the tree-reparameterization updates take effect after a few iterations, leading to much swifter convergence for the first 80 iterations or so. For later iterations, the convergence of TRP slows down, which may be due to numerical precision issues or other effects related to the choice of trees in TRP iterations. For example, each TRP update ignores some local interactions corresponding to the edges removed to form the spanning tree. These edges are covered by other spanning trees in the set used; however, it remains an open question how to choose trees so as to maximize the rate of convergence. In this context, one could imagine a hybrid algorithm (in which BP iterations are interspersed with TRP iterations). If one considers the example of Figure 5.5(b), a switch from TRP to BP after iteration 80 would lead to faster convergence than either algorithm alone. The exploration of such issues remains for future research.

Domain of convergence

The dense 5-node graph shown in Figure 5.6(a) serves to illustrate how TRP updates tend to converge for a wider range of potentials than BP. We simulated a binary process over a range of potential strengths μ ranging from -0.3 to -1.0. Explicitly, for each value of μ, we made a deterministic assignment of the potential for each edge (s,t) of the graph as $\theta_{st;jk} = (2\delta_{jk} - 1)\mu$. For each potential strength we conducted 100 trials, where on each trial the single-node potentials were set randomly by sampling $a_s \sim \mathcal{N}(0, (0.25)^2)$ and setting $[\theta_{s;0} \;\; \theta_{s;1}] = [a_s \;\; -a_s]$. On any given trial, the convergence of a given algorithm was assessed as in Section 5.3.5. Plotted in Figure 5.6(b) is the percentage of successfully converged trials versus potential strength for TRP and



(a) (b)

Figure 5.6. (a) Simple 5-node graph. (b) Comparison of BP and TRP convergence percentages versus function of potential strength on graph in (a). Plotted along the abscissa as a measure of potential strength is the multi-information $D(p(x_1, \ldots, x_n) \,\|\, \prod_{s=1}^{n} p(x_s))$. Both TRP and BP exhibit a threshold phenomenon, with TRP converging for a wider range of potentials.

BP. Both algorithms exhibit a type of threshold behavior, in which they converge with 100% success up to a certain potential strength, after which their performance degrades rapidly. However, TRP updates extend the effective range of convergence, even on such a small graph. Moreover, there is a range of potential settings for which BP almost never converges, whereas TRP converges for almost every trial.[9]

■ 5.4 Analysis of fixed points and convergence

In this section, we present a number of results related to the fixed points and convergence of TRP and BP. In Section 5.4.1, we define and develop properties of a cost function G that arises as an approximation to the Kullback-Leibler divergence. In Section 5.4.2, we show that fixed points of the TRP algorithm satisfy necessary conditions to be a constrained minimum of this cost function. At the heart of our analysis is a geometric characterization of successive iterates; it is of independent interest because it establishes links to successive projection techniques for constrained minimization of Bregman distances [e.g., 30]. By combining our results with those of Yedidia et al. [180], we are able conclude that fixed points of the TRP algorithm coincide with those of BP. Moreover, our analysis allows us to formulate sufficient conditions for convergence of the TRP algorithm in the case of two spanning trees, which we present in Section 5.4.3. In Section 5.4.4, we formalize the fundamental property of reparameterization updates — namely, that they leave unchanged the distribution on the full graph. Some of the consequences of this invariance can be found in an investigation of the geometry of TRP in Section 5.4.4, and in the elementary proof in Section 5.4.5 of the result originally developed in [152, 174] concerning the behavior of BP for jointly Gaussian

[9]This result is not dependent on the symmetry of the problem induced by our choice of edge potentials; for instance, the results are similar if edge potentials are perturbed from their nominal strengths by small random quantities.

distributions.

■ 5.4.1 Approximation to the Kullback-Leibler divergence

The cost function G central to our analysis arises as an approximation to the Kullback-Leibler (KL) divergence [41], one which is exact for a tree. Let $T \in (0,1)^{d(\theta)}$ be a pseudomarginal vector, and let θ be a parameter vector for the original graph $\mathcal{G}$ with cycles. We then define

$$G(T;\theta) \triangleq \sum_{\alpha \in \mathcal{A}} T_\alpha \big[\Theta(T) - \theta\big]_\alpha \tag{5.24}$$

To see how this cost function is related to the KL divergence as defined in equation (2.31), consider the analogous function defined on spanning tree $\mathcal{T}^i$ for a vector $T \in \mathbb{C}^i$:

$$G^i(\Pi^i(T);\Pi^i(\theta)) = \sum_{\alpha \in \mathcal{A}^i} T_\alpha \big[\Theta^i(\Pi^i(T)) - \theta\big]_\alpha \tag{5.25}$$

where $\Pi^i(\theta)$ and $\Pi^i(T)$ are exponential parameter vectors and marginal vectors, respectively, defined on $\mathcal{T}^i$. With the exponential parameterization of equation (5.7) applied to any tree, we have $T_\alpha = \mathbb{E}_{\Theta^i(\Pi^i(T))}[\phi_\alpha]$ for all indices $\alpha \in \mathcal{A}^i$. As a result, the function G^i is related to the KL divergence as follows:

$$D(\Theta^i(\Pi^i(T)) \parallel \Pi^i(\theta)) = G^i(\Pi^i(T);\Pi^i(\theta)) + \Phi(\Pi^i(\theta)) \tag{5.26}$$

In establishing this equivalence, we have used the fact that the partition function of the factorization in equation (5.2) is unity, so that the corresponding log partition function is zero (i.e., $\Phi(\Theta^i(\Pi^i(T))) = 0$). Therefore, aside from an additive constant $\Phi(\Pi^i(\theta))$ independent of T, the quantity $G^i(\Pi^i(T);\Pi^i(\theta))$, when viewed as a function of $\Pi^i(T)$, is equivalent to the KL divergence.

Now consider the problem of minimizing the KL divergence as a function of T, subject to the constraint $T \in \mathbb{C}^i$. The KL divergence in equation (5.26) assumes its minimum value of zero at the vector of correct marginals on the spanning tree — namely, $\widehat{P} = \underline{\Lambda}^i(\Pi^i(\theta)) \in \mathbb{C}^i$. By the equivalence shown in equation (5.26), minimizing the function $G^i(\Pi^i(T);\Pi^i(\theta))$ over $T \in \mathbb{C}^i$ will also yield the same minimizing argument $\widehat{P}$.

For the original graph $\mathcal{G}$ with cycles, the cost function G of equation (5.24) is not equivalent to the KL divergence. The argument leading up to equation (5.26) cannot be applied because $\Lambda(\Theta(T)) \neq T$ for a general graph with cycles. Nevertheless, this cost function lies at the core of our analysis of TRP. Indeed, we show in Section 5.4.2 how the TRP algorithm can be viewed as a successive projection technique for constrained minimization of the cost function G, in which the reparameterization update on spanning tree $\mathcal{T}^i$ as in equation (5.19) corresponds to a projection onto constraint set $\mathbb{C}^i$. Moreover, we will see that G and the Bethe free energy, though different for

points outside the set $\mathbb{C}$ defined in equation (5.12), agree on this constraint set. This allows us to use recent results of Yedidia et al. [180] to link the fixed points of TRP with those of BP.

■ 5.4.2 Tree reparameterization as a successive projection technique

The first result of this section is of a geometric nature: it shows how successive iterates θ^n and θ^{n+1} are related via the cost function G.

Proposition 5.4.1. Assume that the sequence $\{\theta^n\}$ generated by equation (5.20) with step sizes λ^n remains bounded. Let $i = i(n)$ be the tree index used at iteration n. Then for all $U \in \mathbb{C}^i$:

$$G(U;\theta^n) = G(U;\theta^{n+1}) + \lambda^n G(\underline{\Lambda}^i(\mathcal{Q}^i(\theta^n));\theta^n) \tag{5.27}$$

where $\underline{\Lambda}^i$ is defined in equation (5.17b).

Proof. See Appendix C.2. □

An important special case of Proposition 5.4.1 is the unrelaxed update ($\lambda^n = 1$), in which case equation (5.27) simplifies to

$$G(U;\theta^n) = G(U;\theta^{n+1}) + G(\underline{\Lambda}^i(\theta^{n+1});\theta^n) \tag{5.28}$$

Figure 5.7 illustrates the geometry of Proposition 5.4.1 in this unrelaxed setting, for which we are guaranteed the existence[10] of a pseudomarginal T^n such that $\theta^n = \Theta(T^n)$. We project the point T^n onto the constraint set $\mathbb{C}^i$, where the function G^i serves as the distance measure. This projection yields the point $\underline{\Lambda}^i(\theta^{n+1}) \in \mathbb{C}^i$, and we have depicted its relation to an arbitrary U also in $\mathbb{C}^i$.

We note that a result analogous to equation (5.28) holds for the minimum of a Bregman distance over a linear constraint set [e.g., 30]. Well-known examples of Bregman distances include the L^2 norm, and the KL divergence. Choosing the KL divergence as the Bregman distance leads to the I-projection in information geometry [e.g., 7, 33, 43]. Even when the distance is not the L^2 norm, results of the form in equation (5.28) are still called Pythagorean. Indeed, the function G plays the role of the squared Euclidean distance, with the three points T^n, $\underline{\Lambda}^i(\theta^{n+1})$ and U analogous to the vertices of a right triangle, as illustrated in Figure 5.7. In addition, a wide class of algorithms can be formulated as successive projection techniques for minimizing a Bregman distance over a set formed by an intersection of linear constraints (e.g., generalized iterative scaling [50]). The Pythagorean relation is instrumental in establishing the convergence of such techniques [30, 43].

[10] The image of the unit hypercube $(0,1)^{d(\theta)}$ under the map Θ is not all of $\mathbb{R}^{d(\theta)}$, since, for example, given any pseudomarginal $T \in (0,1)^{d(\theta)}$, we have $[\Theta(T)]_{s;j} = \log T_{s;j} < 0$. Nonetheless, for unrelaxed updates producing iterates θ^n, it can be seen that the inverse image of a point θ^n under Θ will be non-empty as soon as each edge has been updated at least once.

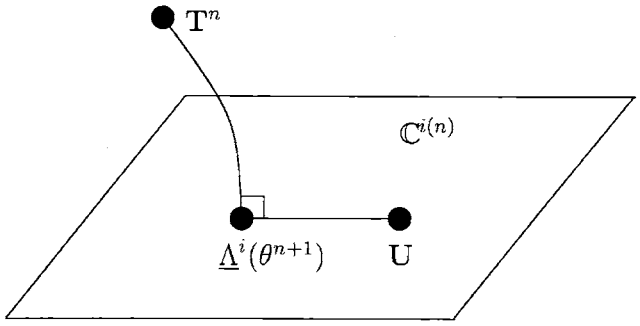


Figure 5.7. Illustration of the geometry of Proposition 5.4.1. The pseudomarginal vector T^n is projected onto the linear constraint set $\mathbb{C}^i$. This yields the point $\underline{\Lambda}^i(\theta^{n+1})$ that minimizes the cost function G^i over the constraint set $\mathbb{C}^i$.

The problem at hand is similar, since we are interested in minimizing a function over a constraint set formed as an intersection of linear constraint sets (i.e., $\mathbb{C} = \cap_i \mathbb{C}^i$). However, the function G is certainly not a Bregman distance since, for instance, it can assume negative values. Nonetheless, Proposition 5.4.1 allows us to show that any fixed point θ^* of the TRP algorithm satisfies the necessary conditions for it to be a local minimum of $G(T;\theta^0)$ over the constraint set $\mathbb{C}$. Although the result extends to other orderings, for concreteness we state it here for a *cyclic ordering* of spanning trees $\mathcal{T}^0, \ldots, \mathcal{T}^{L-1}$: i.e., the tree index for iteration n is chosen as $i(n) = n(\mathrm{mod}\, L)$.

Theorem 5.4.1. Consider a sequence of iterates $\{\theta^n\}$ generated by equation (5.20) with a cyclic tree ordering, and using step sizes $\lambda^n \in [\epsilon, 1]$ for some $\epsilon > 0$. Suppose that the sequence $\{\theta^n\}$ remains bounded and converges to some θ^*. Then

(a) The point θ^* is a fixed point of all the tree operators $\mathcal{Q}^i$. I.e., $\theta^* = \mathcal{Q}^i(\theta^*)$ for all indices $i = 0, \ldots, L-1$. Therefore, each fixed point θ^* is associated with a unique pseudomarginal vector $T^* \in \mathbb{C}$.

(b) The pseudomarginal vector T^* satisfies the necessary conditions for it to be a local minimum of $G(T;\theta^0)$ over the constraint set $\mathbb{C}$:

$$\sum_\alpha \frac{\partial G}{\partial T_\alpha}(T^*;\theta^0)\left[U - T^*\right]_\alpha \;=\; 0$$

for all U in the constraint set $\mathbb{C}$.

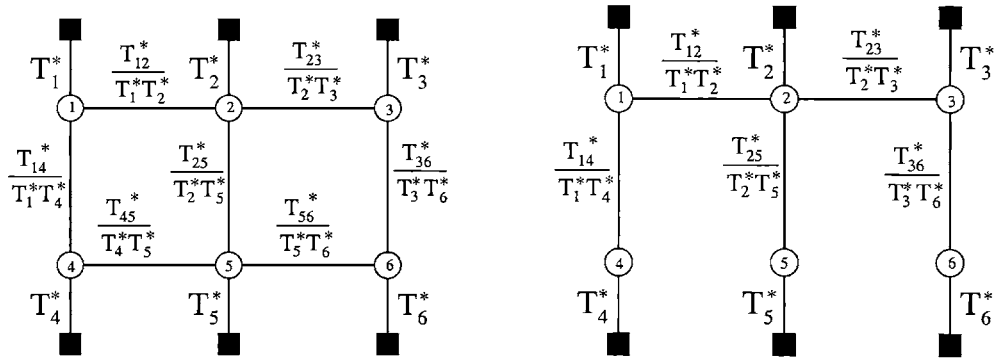
(c) The TRP algorithm always possesses at least one fixed point.

(d) Fixed points of the TRP algorithm coincide with those of BP.

Proof. See Appendix C.2. □

A few remarks about Theorem 5.4.1 are in order. First of all, to clarify the result stated in (a), the unique pseudomarginal vector T^* associated with θ^* can be constructed explicitly as follows. For an arbitrary index α, pick a spanning tree $\mathcal{T}^i$ such that $\alpha \in \mathcal{A}^i$. Then define $T^*_\alpha = [\Lambda^i(\Pi^i(\theta^*))]_\alpha$; that is, T^*_α is the value of this (single node or pairwise) marginal for the tree distribution on $\mathcal{T}^i$ specified by $\Pi^i(\theta^*)$. Note that this is a consistent definition of T^*_α, because the condition of part (a) means that $[\Lambda^i(\Pi^i(\theta^*))]_\alpha$ is the same for all spanning tree indices $i \in \{0, \ldots, L-1\}$ such that $\alpha \in \mathcal{A}^i$. Moreover, this construction ensures that $T^* \in \mathbb{C}$, since it must satisfy the normalization and marginalization constraints associated with every node and edge.

Figure 5.8 illustrates this characterization of fixed points in terms of T^*. Shown



(a) Fixed point on full graph (b) Tree consistency condition.

Figure 5.8. Illustration of fixed point consistency condition. (a) Fixed point $\{T^*_s, T^*_{st}\}$ on the full graph with cycles. (b) Illustration of consistency condition on an embedded tree. The quantities $\{T^*_s, T^*_{st}\}$ must be a consistent set of marginal probabilities for any tree embedded within the full graph.

in panel (a) is an example of a graph with cycles, parameterized according to the approximate marginals T^*_{st} and T^*_s. The consistency condition implies that if edges are removed from the full graph to form a spanning tree, as shown in panel (b), then the quantities T^*_{st} and T^*_s correspond to a consistent set of marginal distributions over the tree. This statement holds for *any* acyclic substructure embedded within the full graph with cycles — not just the spanning trees used to implement the algorithm. Thus, Theorem 5.4.1 provides an alternative and very intuitive view of BP or TRP: such algorithms attempt to reparameterize a distribution on a graph with cycles so that it is consistent with respect to each embedded tree. In this regard, part (c) of Theorem 5.4.1 is noteworthy, in that it guarantees that any positive distribution on a graph can be reparameterized in a form that satisfies the tree-based consistency condition of part (a). It is interesting that such a result, though obvious for any tree, should also hold for a positive distribution on an arbitrary graph with cycles.

It is also interesting to note the link between this fixed point characterization, and Freeman and Weiss' [66] analysis of the max-product algorithm. This is a technique for

computing an approximate MAP assignment, in contrast with the approximate marginal distributions computed by BP/TRP. In particular, they show that the approximate MAP assignment of max-product is guaranteed to be correct on node-induced subgraphs that contain at most a single cycle.

Part (d) of Theorem 5.4.1 follows because our cost function G and the Bethe free energy coincide on the constraint set $\mathbb{C}$. It is also possible to establish this equivalence in a more constructive manner. In particular, any fixed point $\{M_{st}^*\}$ of the message updates in equation (5.3) can be transformed into a pseudomarginal vector $T^* = \{T_s^*, T_{st}^*\}$ as follows:

$$T_{s;j}^* = \kappa \, \psi_{s;j} \prod_{u \in \mathcal{N}(s)} M_{us;j}^* \tag{5.29a}$$

$$T_{st;jk}^* = \kappa \; \psi_{st;jk} \, \psi_{s;j} \, \psi_{t;k} \prod_{u \in \mathcal{N}(s)/t} M_{us;j}^* \prod_{u \in \mathcal{N}(t)/s} M_{ut;k}^* \tag{5.29b}$$

The message fixed point condition of equation (5.3) guarantees that the corresponding T^* belongs to the constraint set $\mathbb{C}$. Membership in $\mathbb{C}$ guarantees that T^* is locally consistent with respect to all the simple two-node trees formed by single edges $(s,t) \in \mathcal{E}$, as illustrated in Figure 5.4. This local consistency on two-node trees then implies that the vector T^* must be consistent on any acyclic substructure (using the equivalence of local and global consistency for trees). That is, T^* satisfies the tree-based consistency condition of part (a), as illustrated in Figure 5.8.

Even more generally, a similar line of reasoning establishes that *any* constrained local minimum of the Bethe free energy, whether obtained by TRP/BP or an alternative minimization technique [e.g., 175, 181], can be identified with a pseudomarginal vector T^* satisfying the conditions of Theorem 5.4.1(a). Therefore, although the fixed point characterization of Theorem 5.4.1(a) (as illustrated in Figure 5.8) emerges very naturally from the TRP perspective, it is actually an *algorithm-independent* result.

■ 5.4.3 Sufficient conditions for convergence for two spanning trees

Proposition 5.4.1 can also be used to derive a set of conditions that are sufficient to guarantee the convergence in the case of two spanning trees. To convey the intuition of the proof, suppose that it were possible to interpret the cost function G as a distance function. Moreover, suppose U were an arbitrary element of $\mathbb{C} = \cap_i \mathbb{C}^i$, so that we could apply Proposition 5.4.1 for each index i. Then equation (5.27) would show that the "distance" between θ^n and an arbitrary element $U \in \mathbb{C}$, as measured by G, decreases at each iteration. As with proofs on the convergence of successive projection techniques for Bregman distances [e.g., 30, 43], this property would allow us to establish convergence of the algorithm.

Of course, there are two problems with the use of G as a type of distance: it is not necessarily non-negative, and it is possible that $G(\Lambda^i(\mathcal{Q}^i(\theta)); \theta) = 0$ for some $\theta \neq \mathcal{Q}^i(\theta)$. With respect to the first issue, we are able to show in general that an appropriate choice

of step size will ensure the non-negativity of $G(\Lambda^i(\mathcal{Q}^i(\theta)); \theta)$ (see Appendix C.4). The following result then states sufficient conditions (including assuming that the second problem does not arise along TRP trajectories) for convergence in the case of two spanning trees:

Theorem 5.4.2. Consider the application of TRP with two spanning trees $\mathcal{T}^0$ and $\mathcal{T}^1$. Suppose that the sequence of iterates $\{\theta^n\}$ remains bounded, and that:
(a) for $i = 0,\ 1$, the condition $G(\underline{\Lambda}^i(\mathcal{Q}^i(\theta^n)); \theta^n) \to 0$ implies that $[\mathcal{Q}^i(\theta^n) - \theta^n] \to 0$.
(b) there exists some integer K such that the condition

$$G(\underline{\Lambda}^0(\mathcal{Q}^1(\theta^n)); \theta^n) G(\underline{\Lambda}^1(\mathcal{Q}^0(\theta^n)); \theta^n) > 0$$

holds for all $n \geq K$.
Then there exist choices of the step sizes λ^n such that the sequence θ^n converges to some θ^* in the desired constraint set. I.e., $\theta^* = \mathcal{Q}^i(\theta^*)$ for $i = 0, 1$.

Proof. See Appendix C.4, which includes a characterization of the step size choices that ensure convergence. □

This result, though its hypotheses cannot be checked a priori, provides some insight into the factors that cause failures of convergence when applying TRP/BP. In particular, the proof of Theorem 5.4.2 shows that assumption (a) is analogous to the gradient-relatedness condition of standard descent algorithms for nonlinear optimization [20].

■ 5.4.4 Geometry and invariance of TRP updates

In this section, we establish a fundamental property of TRP updates— namely, that they leave invariant the full distribution on the graph with cycles. We then exploit this invariance to develop the geometry of TRP updates.

Recall that a crucial feature of the exponential θ−parameterization of equation (5.7) is its overcompleteness. For this reason, given a fixed exponential parameter $\widetilde{\theta}$, it is interesting to consider the following subset of $\mathbb{R}^{d(\theta)}$:

$$\mathcal{M}(\widetilde{\theta}) \triangleq \{\theta \in \mathbb{R}^{d(\theta)} \mid p(\mathbf{x}; \theta) \equiv p(\mathbf{x}; \widetilde{\theta})\} \tag{5.30}$$

where $d(\theta)$ denotes the length of θ as defined in equation (5.8). This set can be seen to be a closed submanifold of $\mathbb{R}^{d(\theta)}$— in particular, note that it is the inverse image of the point $\widetilde{\theta}$ under the continuous mapping $\theta \mapsto p(\mathbf{x}; \theta)$. To further understand the structure of $\mathcal{M}(\widetilde{\theta})$, we need to link the overcomplete θ-parameterization to a minimal parameterization, specified by a linearly independent collection of functions.

We begin with the special case of binary-valued nodes ($m = 2$). Recall from Example 2.2.2 of Section 2.2.1 that the standard minimal representation of a distribution on a binary vector with pairwise potentials has the form:

$$p(\mathbf{x}; \gamma) = \exp\Big\{\sum_s \gamma_s x_s + \sum_{s,t \in \mathcal{E}} \gamma_{st} x_s x_t - \Phi(\gamma)\Big\} \tag{5.31}$$

Here we use the parameter γ to distinguish this minimal representation from the overcomplete parameter θ used in TRP updates. Similarly, as shown by the discussion of Section 2.2.1, an m-ary process on a graph with pairwise potentials has a minimal representation in terms of the collection of functions:

$$\mathcal{R}(s) \triangleq \{x_s^a \mid a = 1, \ldots, m-1\} \quad \text{for } s \in \mathcal{V} \tag{5.32a}$$

$$\mathcal{R}(s,t) = \{x_s^a x_t^b \mid a, b = 1, \ldots, m-1\} \quad \text{for } (s,t) \in \mathcal{E} \tag{5.32b}$$

As in the binary case illustrated above, we let γ be a parameter vector of weights on these functions.

In contrast to the overcomplete case, the minimal representation induces a one-to-one correspondence between parameter vectors γ and distributions $p(\mathbf{x}; \gamma)$. Therefore, associated with the distribution $p(\mathbf{x}; \widetilde{\theta})$ is a unique vector $\widetilde{\gamma}$ such that $p(\mathbf{x}; \widetilde{\theta}) \equiv p(\mathbf{x}; \widetilde{\gamma})$. The dimension of the exponential family [see 5] is given by the length of γ, which we denote by $d(\gamma)$. From equation (5.32), we see that this dimension is given by $d(\gamma) = [\,(m-1)\,N + (m-1)^2\,|\mathcal{E}|\,]$. On the basis of these equivalent representations, the set $\mathcal{M}(\widetilde{\theta})$ can be characterized as follows:

Proposition 5.4.2. The set $\mathcal{M}(\widetilde{\theta})$ of equation (5.30) is a linear submanifold of $\mathbb{R}^{d(\theta)}$ of codimension $d(\gamma)$. It has the form $\{\theta \in \mathbb{R}^{d(\theta)} \mid A\,\theta = \widetilde{\gamma}\,\}$, where A is an appropriately defined $d(\gamma) \times d(\theta)$ matrix of constraints.

Proof. See Appendix C.5. □

Based on this proposition, we now prove a fundamental property of TRP updates:

Theorem 5.4.3. Consider a sequence of TRP iterates $\{\theta^n\}$ generated by the relaxed updates:

$$\theta^{n+1} \;=\; \lambda^n \mathcal{Q}^{i(n)}(\theta^n) + (1 - \lambda^n)\theta^n \tag{5.33}$$

Then the distribution on the full graph with cycles is invariant under the updates: that is, $\theta^n \in \mathcal{M}(\theta^0) = \{\theta \in \mathbb{R}^{d(\theta)} \mid p(\mathbf{x}; \theta) \equiv p(\mathbf{x}; \theta^0)\}$ for all $n = 1, 2, \ldots$. Moreover, any limit point θ^* of the sequence also belongs to $\mathcal{M}(\theta^0)$. In addition, the same statements hold for the reparameterization form of BP presented in Section 5.3.4.

Proof. As previously described, the unrelaxed TRP update of equation (5.19) does indeed leave the distribution unchanged, so that $\mathcal{Q}^i(\theta) \in \mathcal{M}(\theta)$ for all θ. The relaxed update of equation (5.33) is nothing more than a convex combination of two exponential vectors (θ^n and $\mathcal{Q}^{i(n)}(\theta^n)$) that parameterize the same distribution, so that by recourse to Proposition 5.4.2, the proof of the first statement is complete. As noted earlier, $\mathcal{M}(\theta^0)$ is a closed submanifold, so that any limit point of the sequence $\{\theta^n\}$ must also belong to $\mathcal{M}(\theta^0)$. An inductive argument establishes that the reparameterization form of BP also leaves invariant the distribution on the full graph.

□

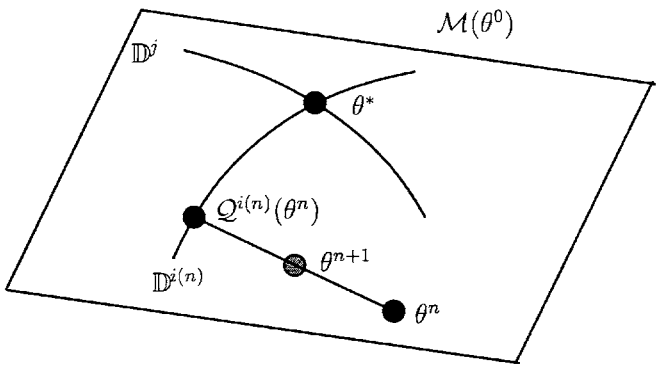


Figure 5.9. Geometry of tree-reparameterization updates in the exponential domain. Iterates are confined to the linear manifold $\mathcal{M}(\theta^0)$. Curved lines within $\mathcal{M}(\theta^0)$ correspond to the intersection $\mathbb{D}^i \cap \mathcal{M}(\theta^0)$, for a particular spanning tree constraint set $\mathbb{D}^i$. Each update entails moving along the line between θ^n and the point $\mathcal{Q}^{i(n)}(\theta^n)$ on $\mathbb{D}^{i(n)}$. Any fixed point θ^* belongs to $\mathbb{D} = \cap_i \mathbb{D}^i$.

Like Theorem 5.4.1, the most important part of Theorem 5.4.3 — namely, that the fixed point θ^* is invariant — is also an *algorithm-independent* result. In particular, it is not difficult to show that any local minimum of the Bethe free energy, no matter what algorithm [e.g., 175, 181] is used to obtain it, is also invariant in the sense of Theorem 5.4.3. The special property of TRP/BP updates are that all iterates — not just the fixed points — are invariant.

In conjunction, Proposition 5.4.2 and Theorem 5.4.3 also lead to a geometric understanding of the TRP updates in the exponential domain (i.e., in terms of the parameter vector θ). In order to describe this geometry, we begin by defining an exponential analog of the constraint set $\mathbb{C}$ as follows:

$$\mathbb{D} = \{\theta \mid \theta = \Theta(T) \text{ for some } T \in \mathbb{C}\} = \Theta(\mathbb{C}) \tag{5.34}$$

If a vector θ belongs to the set $\mathbb{D}$, then it must satisfy certain nonlinear convex constraints (e.g., $\log[\sum_j \exp(\theta_{s;j})] = 0$ for all $s \in \mathcal{V}$; and $\log[\sum_j \exp(\theta_{st;jk} + \theta_{s;j})] = 0$ for all $(s,t) \in \mathcal{E}$). For each spanning tree constraint set $\mathbb{C}^i$, we also define the set $\mathbb{D}^i$ in an analogous manner, and note that by construction $\mathbb{D} = \cap_i \mathbb{D}^i$.

Figure 5.9 illustrates the geometry of the TRP updates in the exponential domain. First of all, in agreement with Theorem 5.4.3, all iterates θ^n are confined to the linear manifold $\mathcal{M}(\theta^0)$. The intersection $\mathbb{D}^i \cap \mathcal{M}(\theta^0)$ will trace out some curved set within the linear manifold. Each step of TRP entails moving along the line between the current iterate θ^n and the projection $\mathcal{Q}^{i(n)}(\theta^n)$. This latter point belongs to the spanning tree constraint set $\mathbb{D}^{i(n)}$. When the sequence θ^n converges to some θ^*, then we are

guaranteed that $\theta^* \in \mathbb{D} \cap \mathcal{M}(\theta^0)$.

■ 5.4.5 Implications for continuous processes

The reparameterization approach can be extended and has important implications for continuous processes as well. In particular, by extension to the Gaussian case, we obtain an elementary proof of a generalization to TRP of the result [152, 174] that when BP converges, the means are, in fact, correct. To establish this result, let us consider the Gaussian analog of TRP. For simplicity in notation, we treat the case of scalar Gaussian random variables at each node (though the ideas extend easily to the vector case). In the scalar Gaussian case, the approximate marginal distribution $T_s(x_s)$ at each node $s \in \mathcal{V}$ is parameterized by a mean μ_s and variance σ_s^2. Similarly, the approximate joint distribution $T_{st}(x_s, x_t)$ can be parameterized by a mean vector $\nu_{st} \triangleq [\nu_{st;s} \;\; \nu_{st;t}]'$, and a covariance matrix. At each iteration, the edge (s,t) is labeled with the edge function $T_{st}/\widetilde{T}_{st;s}\widetilde{T}_{st;t}$, where $\widetilde{T}_{st;s}(x_s) = \int_{-\infty}^{\infty} T_{st}(x_s, x_t)dx_t$ is the marginal distribution over x_s induced by T_{st}. This edge function is parameterized by the mean vector ν_{st}, and a quadratic form $A_{st} = [a_{st;s} \;\; a_{st} \; ; \; a_{st} \;\; a_{st;t}]$. With this set-up, we have:

Proposition 5.4.3. Consider the Gaussian analog of TRP or BP, and suppose that it converges. Then the computed means are exact, whereas in general the error covariances are incorrect.

Proof. From the original problem specification, we have

$$-\log p(\mathbf{x}) \;\; = \;\; 1/2 \; (\mathbf{x} - \widehat{\mu})^T \; P^{-1} \; (\mathbf{x} - \widehat{\mu}) + C \tag{5.35}$$

where P^{-1} is the inverse covariance; C is a constant independent of $\mathbf{x}$; and $\widehat{\mu}$ are the correct means on the graph with cycles.

We begin by noting that the Gaussian analog of Theorem 5.4.3 guarantees that this distribution will remain invariant under the reparameterization updates of TRP (or BP). At any iteration, the distribution is reparameterized in terms of T_s and the edge functions as follows:

$$\begin{aligned} -\log p(\mathbf{x}) \;\; = \;\; & \frac{1}{2} \sum_{(s,t) \in \mathcal{E}} \left\{ a_{st;s}(x_s - \nu_{st;s})^2 + 2a_{st}(x_s - \nu_{st;s})(x_t - \nu_{st;t}) + a_{st;t}(x_t - \nu_{st;t})^2 \right\} \\ & + \frac{1}{2} \sum_s (x_s - \mu_s)^2 / \sigma_s^2 + C \end{aligned} \tag{5.36}$$

Note that the pseudomarginal vector $\{T_s, T_{st}\}$ need not be consistent so that, for example, $\widetilde{T}_{st;s}(x_s)$ need not equal $T_s(x_s)$. However, suppose that TRP (or BP) converges so that these quantities are equal, which, in particular, implies that $\mu_s = \nu_{st;s}$ for all (s,t) such that $t \in \mathcal{N}(s)$. That is, the means parameterizing the edge functions must agree with the means at the node marginals. In this case, equations (5.35) and (5.36) are two alternative representations of the same quadratic form, so that we must have

$\widehat{\mu}_s = \mu_s$ for each node $s \in \mathcal{V}$. Therefore, the means computed by TRP or BP must be exact. In contrast to the means, there is no reason to expect that the error covariances in a graph with cycles need be exact. □

It is worth remarking that there exist highly efficient techniques from numerical linear algebra (e.g., conjugate gradient [54]) for computing the means of a linear-Gaussian problem on a graph. Therefore, although TRP and BP compute the correct means (if they converge), there is little reason to apply them in practice. There remains, however, the interesting problem of computing correct error covariances at each node: we refer the reader to [172] for description of an embedded spanning tree method that efficiently computes both means and error covariances for a linear-Gaussian problem on a graph with cycles.

■ 5.4.6 When does TRP/BP yield exact marginals?

It is clear that the TRP/BP algorithm will yield the exact single node marginals of any $p(\mathbf{x})$ defined on a tree-structured graph. In this section, we address of the question of whether there exist particular problems on graphs with cycles for which a TRP/BP solution will be exact. If so, how large is the set of such problems? Theorems 5.4.1 and 5.4.3 provide the insights that are key to our analysis; these theorems place very severe restrictions on cases where TRP/BP fixed points can be exact.

Let $\mathbf{T}^* \in \mathbb{C}$ be a consistent fixed point of the TRP algorithm in the sense of Theorem 5.4.1. Let P_s denote the actual marginals of the given distribution $p(\mathbf{x})$. We begin by defining two distinct notions of exactness:

Definition 5.4.1 (Exactness).

(a) The point $\mathbf{T}^*$ is *weakly exact* if all the single node marginals are correct. I.e.,

$$T^*_{s;j} = P_{s;j} \qquad \text{for all} \quad s \in \mathcal{V}, \quad j = 0, 1 \tag{5.37}$$

(b) The point $\mathbf{T}^*$ is *strongly exact* if all the marginals, both single node and pairwise, are correct. I.e., in addition to equation (5.37), we have

$$T^*_{st;jk} = P_{st;jk} \qquad \text{for all} \quad (s,t) \in \mathcal{E}, \quad j,k = 0, 1 \ldots m-1 \tag{5.38}$$

The fixed point characterization of Theorem 5.4.1 provides a straightforward technique for constructing TRP/BP fixed points. In particular, we simply specify a distribution in terms of a vector $\mathbf{T}^* \in \mathbb{C}$ that is locally consistent (see equation 5.12) as follows:

$$p(\mathbf{x}; \mathbf{T}^*) \propto \prod_{s \in \mathcal{V}} T^*_s \prod_{(s,t) \in \mathcal{E}} \frac{T^*_{st}}{T^*_s T^*_s} \tag{5.39}$$

Since $\mathbf{T}^*$ belongs to $\mathbb{C}$, it is consistent on any spanning tree embedded within the graph (i.e., it belongs to the constraint set $\mathbb{C}^i$ corresponding to tree $\mathcal{T}^i$) and therefore

is guaranteed to be a fixed point of the TRP updates. The invariance result of Theorem 5.4.3 guarantees that any distribution can be put into the form of equation (5.39) without altering the distribution. As a consequence, the question of exactness reduces to understanding when the T_s^* and T_{st}^* are equivalent to the corresponding marginal distributions of P_s and P_{st} of $p(\mathbf{x}; \mathbf{T}^*)$.

Example 5.4.1 (Symmetric cases). By exploiting the fact that TRP/BP updates preserve any symmetries in the problem, it is easy to develop symmetric examples that are weakly exact. For example, for a binary-valued vector $\mathbf{x}$ defined on any graph, let us specify a set of symmetric pseudomarginals as follows:

$$T_s^* = [0.5 \;\; 0.5]' \tag{5.40a}$$

$$T_{st}^* = \begin{pmatrix} \mu & 0.5 - \mu \\ 0.5 - \mu & \mu \end{pmatrix} \tag{5.40b}$$

where $\mu \in [0, 0.5]$ is arbitrary. It is clear that the corresponding vector $\mathbf{T}^*$ for any such problem instance is an element of $\mathbb{C}$, and a fixed point of TRP. Moreover, for such a choice of $\mathbf{T}^*$, symmetry considerations dictate that the actual single-node marginals of $p(\mathbf{x}; \mathbf{T}^*)$, formed as in equation (5.39), will be uniform $[0.5 \;\; 0.5]'$. Therefore, TRP/BP is weakly exact for any such problem instance.

We now investigate the relation between the joint pairwise pseudomarginals T_{st}^*, and the actual marginals P_{st}. From equation (5.29b), it can be seen that any pseudomarginal T_{st}^* is always related to the original compatibility function ψ_{st} via:

$$T_{st;jk}^* = \kappa \;\; \psi_{st;jk} \; \rho_{s;j} \rho_{t;k}$$

for some vectors ρ_s, ρ_t, and normalization constant κ. For any tree-structured distribution, a relation of this form also holds for the actual marginals.[11] Indeed, a tree-structured distribution is characterized by the property that the dependency between x_s and x_t, for any pair of nodes $(s,t) \in \mathcal{E}$, is mediated entirely by the compatibility function ψ_{st}. Indeed, if the compatibility function ψ_{st} is removed, then x_s and x_t will be independent in the new distribution.

This intuition motivates us to define a notion of degenerate compatibility functions, for which TRP/BP will be exact for uninteresting reasons. To understand the idea of degeneracy, first consider a distribution $p(\mathbf{x})$ of a binary-valued vector $\mathbf{x}$ for which at least one compatibility function ψ_{st} is rank one. I.e., ψ_{st} be written as the outer product $\psi_{st} = \varphi_s \; \varphi_t'$ for a pair of $2-$vectors φ_s and φ_t. These $2-$vectors can be absorbed into the single-node functions ψ_s and ψ_t, so that edge (s,t) can effectively be removed from the graph. Thus, for example, for a binary process, any distribution on a single cycle with at least one rank one compatibility function is equivalent to a tree-structured distribution.

[11] This is necessarily the case, since TRP/BP is exact for tree-structured problems.

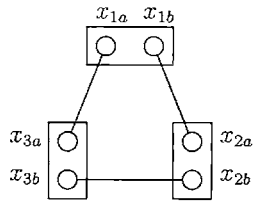


Figure 5.10. Degeneracy of compatibility functions for a 4-state process. Each random variable x_s is decomposed into two random components x_{sa} and x_{sb}, each of which is a binary random variable. These subcomponents are coupled by compatibility functions indicated in lines. Example courtesy of Tommi Jaakkola.

The picture for m-ary processes is a bit more complicated. Here it is possible that all compatibility functions have a rank larger than one, and yet the overall distribution still exhibits degeneracies. Figure 5.10 illustrates a particular example of this degenerate behavior. The graph is a single cycle formed of three nodes, for which the associated random variables $\{ x_s,\ s = 1, 2, 3 \}$ each assume 4 states. Each x_s is decomposed into two random components x_{sa} and x_{sb}, each of which is a binary random variable. The connections between the random variables x_s are shown in solid lines. For instance, x_{2b} is directly coupled to x_{3b}, but not to x_{3a}. None of these compatibility functions are rank one, so that the associated edges cannot be removed without altering the distribution. However, any distribution on this graph still has the property that removing the compatibility function between any pair of variables x_u and x_v leaves them independent in the new distribution.

These illustrative examples motivate the following:

Definition 5.4.2 (Degeneracy). A set of compatibility functions (or the associated distribution) is *degenerate* if for at least one $(s,t) \in \mathcal{E}$, the compatibility function ψ_{st} (viewed as a $m \times m$ matrix) has rank strictly less than m.

The significance of Definition 5.4.2 will become clear in the proof of Proposition 5.4.4, which relies on the fact for a non-degenerate distribution on a tree, two random variables x_u and x_v (for arbitrary $u, v \in \mathcal{V}$) are never independent.[12] Therefore, given a non-degenerate distribution defined by a a single cycle, it is never possible to make a pair of random variables x_u and x_v independent by removing only a single edge.

Proposition 5.4.4 (No exact pairwise marginals on single cycles).
Consider a distribution $p(\mathbf{x})$ of a binary-valued vector $\mathbf{x}$ defined by a set of non-degenerate compatibility functions on a single cycle. Let $\mathbf{T}^*$ be TRP/BP fixed point for this problem. Then *none* of the joint pairwise marginals are correct (i.e., $T^*_{st} \neq P_{st}$ for all $(s,t) \in \mathcal{E}$).

[12]The same statement does *not* hold for a graph with cycles, as Example 5.4.2 will demonstrate.

Proof. A remark on notation before proceeding: throughout this proof, we shall treat the quantities $T^*_{st}(x_s, x_t)$ as functions of x_s, x_t, rather than matrices.

Let $(u, v) \in \mathcal{E}$ be an arbitrary edge. By definition, we have:

$$P_{uv}(x_u, x_v) = = \sum_{\mathbf{x} \in \mathcal{X}^N;\ (x'_u, x'_v) = (x_u, x_v)} p(\mathbf{x}'; \mathbf{T}^*) \tag{5.41a}$$

$$= \kappa \left[\sum_{(x'_u, x'_v) = (x_u, x_v)} \prod_{s \in \mathcal{V}} T^*_s(x'_s) \prod_{(s,t) \in \mathcal{E}/(u,v)} \frac{T^*_{st}(x'_s, x'_t)}{T^*_s(x'_s) T^*_t(x'_t)} \right] \frac{T^*_{uv}(x_u, x_v)}{T^*_u(x_u) T^*_v(x_v)} \tag{5.41b}$$

where κ is a normalization constant. The quantity within square brackets is the joint marginal distribution $\widehat{T}_{uv}(x_u, x_v)$ of a distribution structured according to the tree $\mathcal{T}$ specified by the subset of edges $\mathcal{E}(\mathcal{T}) = \mathcal{E}/(u, v)$. (For future reference, we refer to this

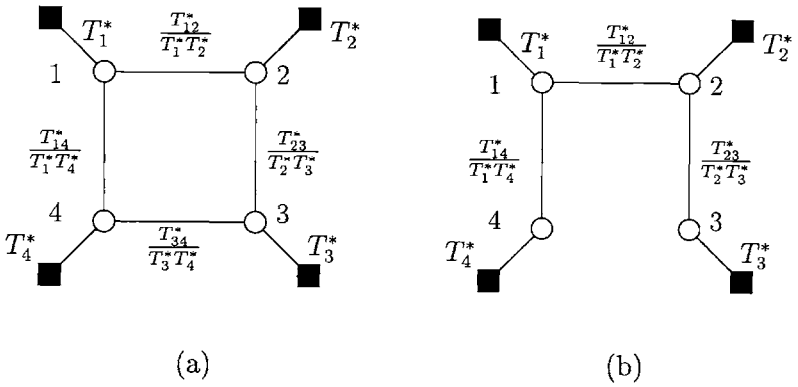


Figure 5.11. Relevant graphs for analyzing non-exactness of pairwise marginals on a single cycle. (a) Distribution $p(\mathbf{x}; \mathbf{T}^*)$ on graph $\mathcal{G}$. (b) Tree-structured distribution $p(\mathbf{x}; \Pi^{\mathcal{T}}(\mathbf{T}^*))$ formed by removing edge $(u, v) = (3, 4)$.

tree-structured distribution as $p(\mathbf{x}; \Pi^{\mathcal{T}}(\mathbf{T}^*))$.) We rewrite equation (5.41b) as:

$$P_{uv}(x_u, x_v) = \kappa\ \widehat{T}_{uv}(x_u, x_v)\ \frac{T^*_{uv}(x_u, x_v)}{T^*_u(x_u) T^*_v(x_v)} \tag{5.42}$$

We now proceed via proof by contradiction. If $P_{uv}(x_u, x_v) = T^*_{uv}(x_u, x_v)$ for all x_u, x_v, then equation (5.42) reduces to

$$\widehat{T}_{uv}(x_u, x_v) = T^*_u(x_u) T^*_v(x_v) \tag{5.43}$$

which implies that x_u and x_v are statistically independent under the tree-structured distribution $p(\mathbf{x}; \Pi^{\mathcal{T}}(\mathbf{T}))$. This can occur only if at least one of the potential functions T^*_{st} for $(s, t) \in \mathcal{E}/(u, v)$ is degenerate. This degeneracy contradicts the assumptions of the proposition, allowing us to conclude that $P_{uv} \neq T^*_{uv}$ for all $(u, v) \in \mathcal{E}$. □

Interestingly, the proof of Proposition 5.4.4 is not valid for graphs with multiple cycles. The final step of the proof is based on the fact that for a tree-structured distribution, the condition of x_u and x_v being independent is equivalent to degeneracy (in the sense of Definition 5.4.2) of the compatibility functions.

This property is not true for graphs with cycles, so that if we try to extend Proposition 5.4.4 to graphs with multiple cycles, the proof breaks down at the final stage. Indeed, for a graph with cycles, it is possible to construct a distribution such that x_u and x_v are independent, even though all the compatibility functions in the graph are non-degenerate (i.e., full rank).

Example 5.4.2. In this example, we shall construct a family of problems for which the TRP pseudomarginals are correct for all single nodes, and for all but a single edge. I.e., the TRP solution is weakly exact, and strongly exact with the exception of a single edge.

Consider the 2-cycle graph shown in Figure 5.12(a), and a distribution $p(\mathbf{x}; \mathbf{T}^*)$ of a binary vector $\mathbf{x}$ parameterized in terms of a TRP fixed point $\mathbf{T}^*$. We specify the set

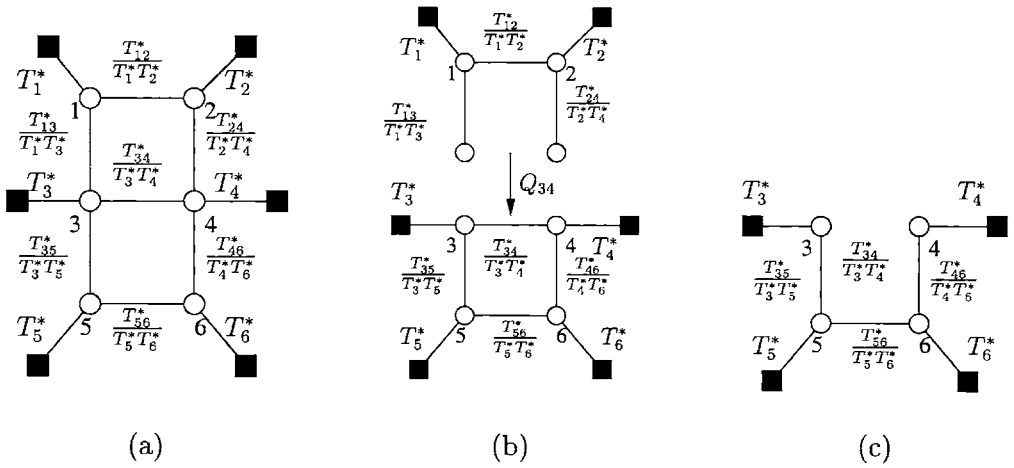


Figure 5.12. Example of a graph and compatibility functions for which a TRP/BP solution is nearly exact. (a) Original graph parameterized in terms of a TRP fixed point $\mathbf{T}^*$. (b) Computing exact marginals on the graph by decomposing into two separate subgraphs. The top subgraph sends a message Q_{34} to the bottom graph. We choose the compatibility functions to ensure that $Q_{34}\frac{T^*_{34}}{T^*_3 T^*_4}$ is a constant function. (c) The cancellation leaves us with a tree.

of pseudomarginals as follows:

$$T_s^* = [0.5 \ 0.5]' \quad \text{for all } s \in \mathcal{V} \tag{5.44a}$$

$$T_{st}^* = \begin{pmatrix} \mu & 0.5-\mu \\ 0.5-\mu & \mu \end{pmatrix} \quad \text{for all } (s,t) \in \mathcal{E}/(3,4) \tag{5.44b}$$

$$T_{34}^* = \begin{pmatrix} \beta & 0.5-\beta \\ 0.5-\beta & \beta \end{pmatrix} \tag{5.44c}$$

The parameter $\mu \in (0, 0.5)$ is arbitrary; we shall specify $\beta \in (0, 0.5)$ as a function of μ.

It can be seen that the vector $\mathbf{T}^*$ is an TRP fixed point, and is therefore consistent on any embedded tree. Consider now the following procedure for obtaining the exact marginals at nodes 3, 4, 5 and 6. We split the graph of Figure 5.12(a) into two subgraphs, as illustrated in panel (b). We marginalize the distribution over nodes 1 and 2, which yields a 2×2 matrix Q_{34} (a function of x_3 and x_4) that is passed as a message to the remaining nodes in the lower subgraph.

We now choose β, thereby specifying T_{34}^*, in such a way to ensure that:

$$Q_{34;jk} \frac{T_{34;jk}^*}{T_{3;j}^* T_{4;k}^*} = \kappa$$

where κ is a normalization constant. Since the single node marginals are uniform, this is equivalent to $T_{34;jk}^* = \kappa' / Q_{34;jk}$, where κ' is chosen so that the entries of T_{34}^* sum to one.

With this choice of T_{34}^*, Q_{34} effectively cancels out the compatibility function on edge $(3,4)$. To complete the computation of the actual marginals, we simply need to operate over the tree shown in Figure 5.12(c). The compatibility functions on this tree are already in standard form, so that the pseudomarginals $\{ T_s^* \mid s = 3,4,5,6 \}$ and $\{ T_{st}^* \mid (s,t) \in \{(3,5),(4,5),(5,6)\} \}$ must be equivalent to the exact marginals P_s and P_{st}. By symmetry, a similar argument applies to the upper subgraph. Therefore, all of the single node marginals T_s^* agree with the actual ones, and all of the joint pairwise marginals T_{st}^*, *except* T_{34}^*, are correct.

As a concrete numerical example, it can be verified that with $\mu = 0.4$, the choice $\beta = 0.196$ yields a graph with the above property. It makes intuitive sense that $\beta < 0.25$, since the compatibility function on edge $(3,4)$ serves to weaken the dependencies that build up between the other two indirect paths between 3 and 4.

■ 5.5 Analysis of the approximation error

An important but very difficult problem is analysis of the errors that arise from approximation techniques such as BP and TRP. In fact, to date very few results are available on characterizing the error, as described briefly Section 5.1. Empirical simulations show that BP gives good approximations for certain graphs (e.g., those with long cycles relative to potential strength); in other cases, however, the approximations can be very

poor. In the absence of error analysis, it is impossible to gauge the validity of the approximation. In this section, we provide a contribution to this important problem.

Our analysis of the approximation error is based on two fundamental properties of a fixed point θ^*. First of all, part (a) of Theorem 5.4.1 dictates that for an arbitrary spanning tree $\mathcal{T}^i$, the single node elements $\theta^*_{s;j} = \log T^*_{s;j}$ correspond to a consistent set of marginal distributions on the spanning tree. That is, the quantities $T^*_{s;j}$ have two distinct interpretations:

(a) as the BP or TRP approximations to the exact marginals on the original graph with cycles.

(b) as the single node marginals of a distribution defined by the spanning tree $\mathcal{T}^i$.

Secondly, by the invariance stated in Theorem 5.4.3, the distribution $p(\mathbf{x}; \theta^*)$ induced by the fixed point θ^* is equivalent to the original distribution $p(\mathbf{x}; \theta^0)$. In conjunction, these two properties imply that the exact marginals on the full graph with cycles are related to the approximations $T^*_{s;j}$ by a relatively simply perturbation — namely, removing edges to form a spanning tree. On this basis, we first derive an exact expression relating expectations under two different distributions, from which we proceed to derive lower and upper bounds on the approximation error.

In the development to follow, we will use the notation and perspective of the TRP algorithm. However, it should be noted that like Theorems 5.4.1 and 5.4.3, our analysis of the approximation error is again algorithm-independent. That is, it applies to any local minimum of the Bethe free energy, whether obtained by TRP/BP or an alternative minimization technique.

■ 5.5.1 Exact expression

Our treatment begins at a slightly more general level, before specializing to the case of marginal distributions and the TRP algorithm. Consider a function $f : \mathcal{X}^N \rightarrow \mathbb{R}$, and two distributions $p(\mathbf{x}; \widetilde{\theta})$ and $p(\mathbf{x}; \theta)$. Suppose that we wish to express the expectation $\mathbb{E}_{\widetilde{\theta}}[f(\mathbf{x})]$ in terms of an expectation over $p(\mathbf{x}; \theta)$. Using the exponential representation of equation (5.6), it is straightforward to show that

$$\mathbb{E}_{\widetilde{\theta}}[f(\mathbf{x})] \;=\; \mathbb{E}_{\theta}\left[\exp\left\{ \sum_{\alpha} (\widetilde{\theta} - \theta)_{\alpha} \phi_{\alpha}(\mathbf{x}) + \Phi(\theta) - \Phi(\widetilde{\theta}) \right\} \; f(\mathbf{x}) \right] \tag{5.45}$$

Note that this is a change of measure formula, where the exponentiated quantity can be viewed as the Radon-Nikodym derivative.

We now specialize equation (5.45) to the problem at hand. Let us denote the actual single node marginal $p(x_s = j; \theta^0)$ on the graph with cycles by

$$P_{s;j} \;\triangleq\; \mathbb{E}_{\theta^0}[\delta(x_s = j)] \;=\; \mathbb{E}_{\theta^*}[\delta(x_s = j)] \tag{5.46}$$

where $\delta(x_s = j)$ is the indicator function for node x_s to take value j. To derive equation (5.46), we have used the invariance property (i.e., $p(\mathbf{x}; \theta^0) = p(\mathbf{x}; \theta^*)$) of

Theorem 5.4.3. Assume that TRP (or BP) converges to some exponential parameter θ^*, with associated pseudomarginal vector T^*. In this case, Theorem 5.4.1 guarantees that the single-node elements of T^* can be interpreted as the following expectations:[13]

$$T^*_{s;j} \triangleq \mathbb{E}_{\Pi^i(\theta^*)}[\delta(x_s = j)] \tag{5.47}$$

We now make the assignments $\widetilde{\theta} = \theta^*$; $\theta = \Pi^i(\theta^*)$; and $f(\mathbf{x}) = \delta(x_s = j)$ in equation (5.45) and re-arrange to obtain

$$P_{s;j} - T^*_{s;j} = \mathbb{E}_{\Pi^i(\theta^*)}\Big[\big(\exp\big\{\sum_{\alpha \notin \mathcal{A}^i} \theta^*_\alpha \phi_\alpha(\mathbf{x}) - \Phi(\theta^*)\big\} - 1\big)\,\delta(x_s = j)\Big] \tag{5.48}$$

where we have used the fact that $\Phi(\Pi^i(\theta)) = 0$. Equation (5.48) is an exact expression for the error $(P_{s;j} - T^*_{s;j})$ in terms of an expectation over the tree-structured distribution $p(\mathbf{x}; \Pi^i(\theta^*))$. Note that equation (5.48) holds for all spanning tree indices $i \in \{0, \ldots, L-1\}$.

■ 5.5.2 Error bounds

It is important to observe that equation (5.48), though conceptually interesting, is of limited practical use. The problem stems from the presence of the residual term

$$r^i(\mathbf{x}) \triangleq \exp\{\sum_{\alpha \notin \mathcal{A}^i} \theta^*_\alpha \phi_\alpha(\mathbf{x})\}$$

within the expectation on the right-hand side. For most problems, computing the expectation of $r^i(\mathbf{x})$ will not be tractable, since it is a function of all nodes x_s incident with any edge removed to form spanning tree $\mathcal{T}^i$. Indeed, if the computation of equation (5.48) were easy for a particular graph, this would imply that we could compute the *actual* marginals, thereby obviating the need for an approximation technique such as BP/TRP.

This intractability motivates the idea of bounding the approximation error. In order to do so, we make use the bounds derived in Chapter 3. In particular, on the basis of Proposition 3.3.1, we can derive the following error bounds:

Theorem 5.5.1. Let θ^* be a consistent fixed point of TRP/BP, giving rise to approximate marginal distributions $T^*_{s;j}$, and let $P_{s;j}$ be the actual marginal distributions on the graph with cycles. Define the log error $E_{s;j} \triangleq \log T^*_{s;j} - \log P_{s;j}$. Then for each of the spanning trees $\mathcal{T}^i$, we have:
Lower bound:

$$E_{s;j} \;\leq\; D(\Pi^i(\theta^*) \,\|\, \theta^*) - \frac{1}{T^*_{s;j}} \sum_{\alpha \notin \mathcal{A}^i} \theta^*_\alpha \; \mathrm{cov}_{\Pi^i(\theta^*)}\{\delta(x_s = j),\, \phi_\alpha\} \tag{5.49}$$

[13]The tree-based consistency condition of Theorem 5.4.1 ensures that $T^*_{s;j} = \mathbb{E}_{\Pi^i(\theta^*)}[\delta(x_s = j)]$ independent of the choice of spanning tree index $i \in \{0, \ldots, L-1\}$.

Upper bound:

$$E_{s;j} \geq \log T^*_{s;j} - \log\left[1 - (1 - T^*_{s;j})\exp(\Delta)\right] \quad (5.50a)$$

$$\Delta \triangleq -D(\Pi^i(\theta^*) \,\|\, \theta^*) - \frac{1}{1 - T^*_{s;j}} \sum_{\alpha \notin \mathcal{A}^i} \theta^*_\alpha \operatorname{cov}_{\Pi^i(\theta^*)}\{\delta(x_s = j),\, \phi_\alpha\} \quad (5.50b)$$

Proof. The bounds of this theorem follow by appropriately applying Proposition 3.3.1 from Chapter 3. We first make the identifications $\widetilde{\theta} = \theta^*$ and $\theta = \mathcal{I}^i(\Pi^i(\theta^*))$, and then set $f(\mathbf{x}) = \delta(x_s = j)$, a choice which satisfies the assumptions of Proposition 3.3.1. Equation (5.49) then follows by applying Proposition 3.3.1, followed by some algebraic manipulation. The lower bound follows via the same argument applied to $f(\mathbf{x}) = 1 - \delta(x_s = j)$, which also satisfies the restrictions of Proposition 3.3.1. □

A number of remarks about Theorem 5.5.1 are in order. For practical purposes, the primary consideration is the cost of computing these lower and upper bounds. The summations appearing in equations (5.49) and (5.50) are tractable. In particular, each of the covariances can be calculated by taking expectations over tree-structured distributions, and their weighted summation is even simpler. On the other hand, within the KL divergence $D(\Pi^i(\theta^*) \,\|\, \theta^*)$ lurks a negative log partition function $-\Phi(\theta^*)$ associated with the graph with cycles. In general, computing this quantity is as costly as performing inference on the original graph. To obtain computable bounds, we require an upper bound on the log partition function. In Chapter 7, we derive a set of such upper bounds, which allow us to compute bounds of the form in Theorem 5.5.1.

On the conceptual side, Theorem 5.5.1 highlights three factors that control the accuracy of the TRP/BP approximation. For the sake of concreteness, consider the upper bound of equation (5.49).

(a) the KL divergence $D(\Pi^i(\theta^*) \,\|\, \theta^*)$ measures the discrepancy between the tree-structured distribution $p(\mathbf{x}; \Pi^i(\theta^*))$ and the distribution $p(\mathbf{x}; \theta^*)$ on the graph with cycles. It will be small when the distribution $p(\mathbf{x}; \theta^*)$ is well-approximated by a tree. This term reflects the empirical finding that BP performs well on graphs that are approximately tree-like (e.g., graphs with fairly long cycles).

(b) the covariance terms in the second summation measure the strength of the interaction, as measured under the tree distribution $p(\mathbf{x}; \Pi^i(\theta))$, between the delta function at node s and clique potential ϕ_α. When the removed clique potential ϕ_α interacts only weakly with the delta function, then this covariance term will be small and so have little effect.

(c) the weight θ^*_α on each covariance term measures the strength of the clique potentials $\{\phi_\alpha\}$ that were removed to form the spanning tree.

A number of extensions to the bounds presented in Theorem 5.5.1 are possible. First of all, it is worthwhile emphasizing that Theorem 5.5.1 provides L bounds on the

single-node marginal $P_{s;j}$ — one for each of the L spanning trees used in the algorithm.[14] This allows us to choose the *tightest* of all the spanning tree bounds for a given index $(s;j)$. Point (b) above suggests that one should expect tighter bounds when using a tree formed by removing edges "relatively far" away from node s of interest. Indeed, the covariance between $\delta(x_s = j)$ and the removed clique potential ϕ_α captures this notion in a precise fashion.

Secondly, equation (5.45) as well as Proposition 3.3.1 both hold for arbitrary choices of the function $f : \mathcal{X}^N \to [0, 1]$. Different choices will allow us to derive bounds on the error of other approximate marginals. For instance, making the choice of function $f(\mathbf{x}) = \delta(x_s = j)\ \delta(x_t = k)$ will lead to bounds on the pairwise marginal $P_{st;jk}$. Thirdly, note that the bounds of Theorem 5.5.1 are first-order, because they account for the interaction between the function f and clique potentials ϕ_α only up to first order (i.e., $\text{cov}_\theta\{f, \phi_\alpha\}$). On the basis of equation (5.48), it is possible to derive stronger bounds by including higher order terms (as in [e.g., 123]), though with an associated price of increased computation. A thorough analysis of various bounds and the inherent tradeoffs is open for future work.

■ 5.5.3 Illustrative examples of bounds

The tightness of the bounds given in Theorem 5.5.1 varies, depending on the graph topology, the choice of clique potentials, and the choice of spanning tree. In this section, we give some simple numerical illustrations. In all cases, we use the results of Chapter 7 to compute an upper bound on the log partition function $\Phi(\theta^*)$, so that the results of Theorem 5.5.1 are actually computable. Observe that bounds of Theorem 5.5.1 can be transformed into lower and upper bounds on the exact marginals. Specifically, a lower bound (respectively upper bound) on the log error $E_{s;1} = \log T^*_{s;1} - \log P_{s;1}$ gives rise to an upper bound (respectively lower bound) on the actual marginal via the relation $P_{s;1} = T^*_{s;1}\ \exp\{-E_{s;1}\}$.

Varying the clique potentials

We first consider the choice of clique potentials. Figure 5.13 illustrates the behavior of TRP and the corresponding error bounds for a binary-valued process on the 3×3 grid shown in panel (a) for different settings of clique potentials. Shown in panels (b) through (d) are the actual marginals $P_{s;1} = p(x_s = 1; \theta^*)$ compared to the TRP approximations $T^*_{s;1}$ plotted on a node-by-node basis. We have also used the TRP/BP fixed point to compute lower and upper bounds on $P_{s;1}$; these bounds on the actual marginal are plotted in panels (b) through (d) as well.

Panel (b) illustrates the case of weak potentials, so that TRP /BP leads to the very accurate approximation of the exact marginals $P_{s;1}$. The gap between the corresponding lower and upper bounds on the exact marginals is narrow, which assures us that the

[14] More generally, results of the form of Theorem 5.5.1 hold for any acyclic subgraph embedded within the graph, not just the spanning trees used to implement the algorithm.

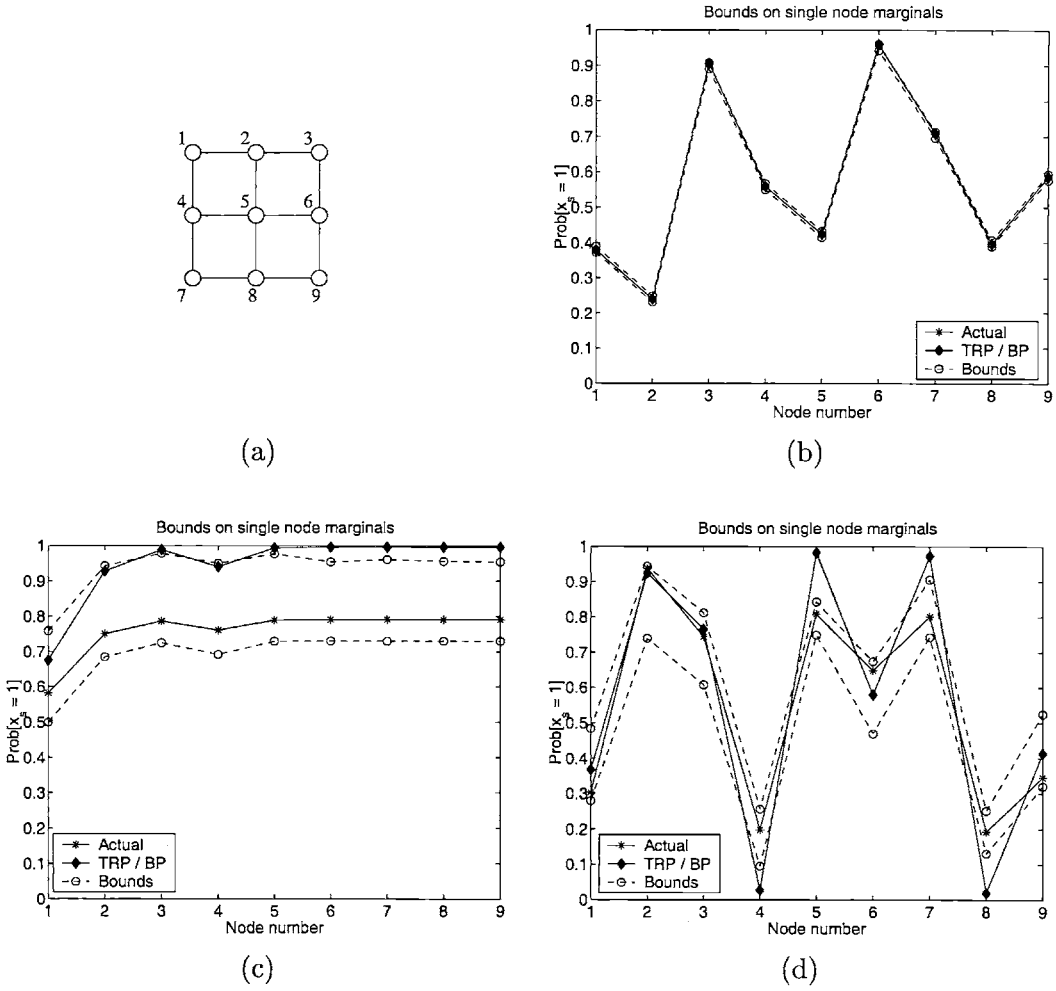


Figure 5.13. Behavior of the bounds of Theorem 5.5.1 for the 3×3 grid shown in panel (a) under various settings of clique potentials. Panels (b) through (d) show the actual marginals $P_{s;1}$ versus the TRP approximations $T^*_{s;1}$, as well as upper and lower bounds on the exact marginals. (b) For weak potentials, TRP gives a good approximation, and the gap between the lower and upper bounds on the exact marginals is very narrow. (c) For strong attractive potentials, the approximation is poor, and the gap becomes relatively large. (d) Similarly, the approximation is also poor for strong mixed potentials. Note how for certain nodes in (c) and (d), the TRP/BP approximation lies above the upper bounds on the actual marginal $P_{s;1}$.

approximation is excellent.

Panel (c), in contrast, displays the more interesting choice of strong attractive clique potentials, for which TRP/BP approximations tend to be skewed towards an extreme value (one in this case). The gap between the upper and lower bounds on the exact marginals is large in comparison to those shown in panel (b). Despite the relative

looseness of the bounds, note how the TRP/BP approximate marginals $T^*_{s;1}$ *exceed* the upper bounds for certain nodes (in particular, nodes 5 through 9). Consequently, the error bounds inform us that the BP approximation is very inaccurate for these nodes.

Panel (d) displays the case of strong mixed potentials, where the TRP/BP approximation is again inaccurate. Once more, the TRP/BP approximation lies outside the window of bounds for the actual marginal for certain nodes (e.g., nodes 4,5,7,8).

Choice of spanning tree for bounds

As mentioned earlier, bounds of the form in Theorem 5.5.1 hold for any spanning tree (or more generally forest) embedded within the graph $\mathcal{G}$. Here we show that the choice of spanning tree can also make a significant difference in the tightness of the bounds. Shown in panels (a) and (b) of Figure 5.14 are the actual marginals $P_{s;1}$ and TRP/BP

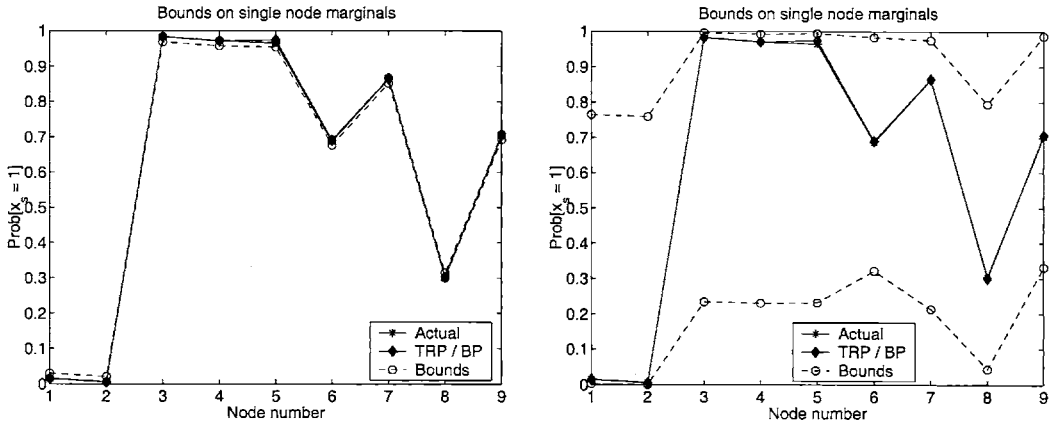


Figure 5.14. The narrowness of the bounds can depend strongly on the choice of spanning tree. Both panels show the exact marginals $P_{s;1}$ compared to the TRP/BP approximations $T^*_{s;1}$ on a node-by-node basis, as well as lower and upper bounds on the actual marginals. (a) The maximum weight spanning tree in the graph yields very narrow bounds. (b) The minimum weight spanning tree yields very poor bounds.

approximations $T^*_{s;1}$ for a particular binary problem on the 3×3 grid. Note that the TRP/BP approximation is very good in this case.

As in the previous section, we also used the TRP/BP fixed point to calculate upper and lower bounds on the actual marginal $P_{s;1}$; here we investigated the effect of varying the spanning tree used to compute the bound. Our choice of spanning tree was based on the following heuristic. We first computed the minimal exponential parameter γ^* (as in equation (5.31)) corresponding to the overcomplete representation $p(\mathbf{x}; \theta^*)$. We then computed the maximum and minimum weight spanning trees with Kruskal's algorithm [107,116], using $|\gamma^*_{st}|$ as the weight on edge (s,t). Panels (a) and (b) show the lower and upper bounds on the exact marginals obtained from the TRP solution using

the maximum and minimum weight spanning trees, respectively. The disparity between the two sets of bounds is striking. Further work should address techniques for choosing spanning trees that yield relatively tight bounds.

■ 5.6 Discussion

The contributions of this chapter are both practical and theoretical. On the practical side, we have introduced a class of iterative algorithms for approximate estimation of stochastic processes on graphs with cycles. Common to algorithms of this class is the operation of reparameterizing distributions defined on embedded trees. We showed, for example, that belief propagation (BP) is a special case of such an algorithm and derived a "message-free" implementation of BP that requires less computational storage than the traditional message-passing formulation. More generally, we considered updates that involve global operations over spanning trees of the graph. The convergence properties of these tree-based reparameterization (TRP) algorithms were shown to be superior to those of BP. There is great freedom in the detailed specification of TRP algorithms — in particular, in the choice of spanning trees (where the only constraint is that each edge in the graph with cycles belong to at least one tree). This freedom suggests open questions of a graph-theoretic nature: how to choose a set of spanning trees so as to make the bounds of Section 5.5 as tight as possible, to guarantee convergence, or to optimize the rate of convergence. One possibility is to choose spanning trees randomly from the full graph, a task for which there exist well-known algorithms [e.g., 141].

The reparameterization framework also led to a number of important theoretical results. In particular, we obtained a new and intuitive characterization of the fixed points of any TRP algorithm (including BP as a special case) in terms of consistency conditions over any tree embedded within the graph with cycles. We also proved a result which, though obvious from a reparameterization perspective, is nonetheless fundamental — namely, any TRP algorithm does not alter the full distribution on the graph with cycles. This invariance has a number of important consequences. In particular, in conjunction with the fixed point characterization, this invariance enabled us to derive an exact expression for the error between the TRP/BP approximations, and the exact marginals on an arbitrary graph with cycles. We also derived upper and lower bounds on this error, which illuminate the conditions governing performance of such approximation methods. In conjunction with the results of Chapter 7, these error bounds are computable, and thus provide valuable information on the performance of TRP/BP.

The theoretical results of this chapter followed very naturally from the perspective of tree-based reparameterization. However, it should be noted that most of these results — most importantly the characterization and invariance of fixed points, and associated error analysis — are, in fact, *algorithm-independent*. That is, the same results apply to any local minimum of the Bethe free energy, regardless of the algorithm [e.g., 175,

181] used to find it. Moreover, this chapter focused exclusively on reparameterization algorithms which involved only singleton and pairwise cliques. However, as we will see in Chapter 6, the ideas and results from this chapter can be extended to more advanced approximation techniques that either operate over larger cliques [e.g., 114, 180], or make use of more complex approximating structures [131]. We shall extend the same fixed point characterization, invariance and error analysis to these higher-order reparameterization algorithms.

Chapter 6

Exploiting higher-order structure for approximate estimation

6.1 Introduction

The focus of Chapter 5 was one of the most widely-studied algorithms for the approximate computation of marginal distributions — namely, the belief propagation (BP) or sum-product algorithm [e.g., 3, 130, 147, 173, 180]. It is well-documented that the performance of BP varies considerably, depending both on the graph topology and the settings of the potentials. It is therefore desirable to develop principled methods for improving the approximation.

In this chapter, we present a framework for developing and analyzing a large class of more advanced algorithms for approximate inference. Each approximation is specified by a subgraph of the original graph (known as the core structure) and a corresponding set of residual cliques, such that the union of the core and the residual terms covers the clique set of the full graph. This framework is quite general, in that it includes a large number of approximations, including the Bethe free energy associated with belief propagation, as well as more advanced methods such as Kikuchi approximations [180], and the structured approximations of Minka [131]. Although techniques that exploit more structure than the Bethe free energy entail a higher computational cost, the hope is that they lead to better approximations to the actual marginal distributions.

Worthy of note is that the notion of reparameterization from Chapter 5 carries over to these more advanced approximations in a very natural way. As a consequence, many of the important results from Chapter 5 also have analogs for these more advanced methods. For instance, a central result of Chapter 5 was that TRP/BP updates do not alter the distribution on the full graph with cycles. This invariance had a number of important consequences, perhaps the most important of which being its role in characterizing the approximation error. All of the approximations that we consider in this chapter satisfy a generalized form of this invariance. As with our work in Chapter 5, this invariance allows us to derive an exact expression for the approximation error, as well as upper and lower bounds. By recourse to the results of Chapter 7, these bounds are computable. Indeed, we shall provide examples for which the error analysis provide valuable information for assessing when the use of a more advanced technique is

appropriate.

■ 6.1.1 Variational formulation

The goal of this chapter is to compute approximations to the single node marginals of a given distribution $p(\mathbf{x})$. I.e., we want to compute:

$$p(x_s) = \sum_{\mathbf{x}' \in \mathcal{X}^N;\, x'_s = x_s} p(\mathbf{x}') \tag{6.1}$$

Despite its simple formulation, this problem is difficult because the number of terms in the summation ($\mathcal{O}(m^N)$) explodes exponentially in the number N of nodes.

As in Chapter 5, the variational problem that underlies the approximations that we shall consider is that of minimizing the Kullback-Leibler (KL) divergence between $p(\mathbf{x})$ and some approximating distribution $q(\mathbf{x})$. For discrete-valued random vectors $\mathbf{x}$, this divergence is given by:

$$D(q \,\|\, p) \;=\; \sum_{\mathbf{x} \in \mathcal{X}^N} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})} \tag{6.2}$$

It is well-known [41] that the KL divergence $D(q \,\|\, p)$ is non-negative, and equal to zero if and only if $q = p$. Therefore, if we actually performed this unconstrained minimization, we would simply recover the quantity $p(\mathbf{x})$ which, as a vector with m^N elements, is so large as to be impossible to store or manipulate. Therefore, this initial try does not bring us much closer to computing local marginal distributions.

While this direct approach is not helpful, other approximate formulations turn out to be fruitful. The Bethe free energy [180] is a particular approximation to the KL divergence. It depends on a set of *pseudomarginals* $\vec{\mathbf{Q}} = \{Q_s, Q_{st}\}$, which are required to be locally consistent (i.e., $\sum_{x'_t} Q(x_s, x'_t) = Q_s(x_s)$). We use these pseudomarginals to specify a distribution q on the graph:

$$q(\mathbf{x}) \;=\; \frac{1}{Z(\vec{\mathbf{Q}})} \prod_{s \in \mathcal{V}} Q_s(x_s) \prod_{(s,t) \in \mathcal{E}} \frac{Q_{st}(x_s, x_t)}{Q_s(x_s) Q_t(x_t)} \tag{6.3}$$

If $\mathcal{G}$ is a tree, then not only are we guaranteed that the pseudomarginals $\vec{\mathbf{Q}}$ are globally consistent (i.e., they are a valid set of marginals for some distribution over $\mathcal{G}$), but in fact they correspond to the marginals of q. In this case, the associated partition function $Z(\vec{\mathbf{Q}})$ is equal to one. On the other hand, if $\mathcal{G}$ is not tree-structured, the $\vec{\mathbf{Q}}$ may not satisfy global consistency; moreover, even if they do satisfy this property, they may not be the correct marginals associated with q. Nevertheless, it is the pseudomarginals $\vec{\mathbf{Q}}$ on which the Bethe free energy focuses.

The Bethe free energy arises from substituting the factorization of q given in equation (6.3) into the KL divergence $D(q \,\|\, p)$. There are a few catches: in performing this substitution, we neglect the fact that for a graph with cycles the partition function

$Z(\vec{\mathbf{Q}})$ is *not* equal to one (as it would be for a tree); and we assume that the pseudo-marginals $\{Q_s, Q_{st}\}$ are, in fact, the exact marginals of q. Therefore, the Bethe free energy is, in general, only an approximation to the KL divergence. Since minimizing the KL divergence would yield the true distribution p, the hope is that minimizing the Bethe free energy, as an approximation to this divergence, will yield minimizing arguments $\vec{\mathbf{Q}}^*$ that are approximations to the true marginals. As shown by Yedidia et al. [180], belief propagation is one particular algorithm for attempting to solve this minimization problem.

The approaches described in this chapter are based on this same guiding principle. In particular, we approximate the KL divergence of equation (6.2) using cost functions that depend only on (relatively) local pseudomarginal functions. The aim is then to minimize these cost functions so as to obtain approximations to the marginals of p.

■ 6.1.2 Related work

A number of improvements to the Bethe free energy have been proposed in previous work. First of all, Yedidia et al. [180] developed extensions based on Kikuchi approximations [114] from statistical physics. Note that the representation of q given in equation (6.3) depends only on single node Q_s and pairwise pseudomarginals Q_{st}; as a result, the Bethe free energy is a function only of these local quantities. Kikuchi approximations extend the Bethe free energy in a natural way by including higher order terms (i.e., marginals over larger subsets of nodes). Yedidia et al. developed a message-passing algorithm, analogous to BP, for minimizing such Kikuchi approximations, and found empirically that Kikuchi approximations typically lead to more accurate estimates of the marginal distributions. Secondly, several researchers have observed that belief propagation corresponds to updating a fully factorized approximation [e.g., 77,131,132,147]. Based on this observation, Minka [131] proposed extensions to belief propagation that entail updating distributions with more complex structure (e.g., a distribution induced by a tree). He also proposed an algorithm, which he called expectation-propagation, for updating such distributions. An ancillary contribution of this chapter is to show how both the Kikuchi approach of Yedidia et al. [180] and the expectation-propagation of Minka [131] can be formulated within a common framework.

■ 6.1.3 Overview of the chapter

This chapter is organized as follows. Section 6.2 describes the key elements of the approximations to the KL divergence (i.e., the cost functions) to be considered in this chapter. In Section 6.3, we develop properties of these cost functions, including conditions that govern when and how they are exact representations of the KL divergence. We illustrate these properties with a number of examples. The focus of Section 6.4 is not the cost functions themselves, but rather their optimizing arguments. It is these arguments that are of primary interest for the purposes of approximate inference. The key result of Section 6.4 is an *invariance* satisfied by the local minima of any of the approximations to the KL divergence considered in this chapter. This invariance, in

fact, is algorithm-independent, in that it holds for any local minimum regardless of the particular algorithm used to find it. We also provide a general set of updates, specified either in terms of message-passing or reparameterization, to try to minimize these cost functions subject to appropriate marginalization constraints. Particular versions of this algorithm are closely related to either the generalized belief propagation updates of Yedidia et al. [180], or the expectation-propagation updates of Minka [131]. The generalized message-passing updates have the interesting property that each of the iterates (and not just its fixed points) corresponds to a different reparameterization of the original distribution. In Section 6.5, we exploit the invariance to derive an exact expression for the error between the approximate and true marginals, as well as lower and upper bounds on this error. In Section 6.6, we illustrate properties of the approximations of this chapter by applying them to simple problems. We conclude with a discussion in Section 6.7.

■ 6.2 Elements of the approximations

In this section, we describe in detail the following four key elements of our framework for approximations:

(a) the *core structure*, which is given by a subgraph of $\mathcal{G}$ (or more generally, of a triangulated version of $\mathcal{G}$) over which an approximating distribution is optimized.

(b) a set of *residual elements* — namely, cliques that are in $\mathcal{G}$ but are not included in the core structure.

(c) a set of *auxiliary distributions* defined on augmented subgraphs formed by adjoining additional cliques to the core structure.

(d) a set of *marginalization operators* that enforce constraints between the auxiliary distributions and the core distributions

■ 6.2.1 Basic definitions and notation

In this section, we introduce the basic definitions and notation required for subsequent developments. The development of this section presupposes familiarity with the graph-theoretic concepts and notation introduced in Section 2.1.1.

Given a graph $\mathcal{G}$, it will often be useful to consider a triangulated version. Although this triangulated version is not unique in general, we assume throughout this chapter that a particular triangulated version $\widetilde{\mathcal{G}}$ is chosen and fixed. The set of cliques of $\widetilde{\mathcal{G}}$ will be denoted by $\widetilde{\mathbf{C}}$; this is (in general) a superset of the set of cliques of $\mathcal{G}$, denoted by $\mathbf{C}$.

Let $p(\mathbf{x})$ denote the distribution whose marginals we would like to approximate; this *target distribution* is defined as a product of compatibility functions ψ_C on the cliques

of $\mathcal{G}$ as follows:

$$p(\mathbf{x}) \triangleq \frac{1}{Z(p)} \prod_{C \in \mathbf{C}} \psi_C(\mathbf{x}_C) \tag{6.4}$$

In addition to this target distribution, there exist other distributions that can be formed as a product of compatibility functions ψ_C for C belonging to a *subset* of the full clique set $\mathbf{C}$. We shall use the notation $P_{(\cdot)}$ to refer to distributions constructed in this manner. In contrast, we reserve the notation $Q_{(\cdot)}$ to refer to approximating distributions. In particular, the marginal distributions of a given $Q_{(\cdot)}$ will represent approximations to the marginal distributions of the target distribution p.

■ 6.2.2 Core structures and distributions

The primary ingredient of any of the approximations developed in this chapter is the core structure, over which an approximating distribution $Q_{\mathbf{A}}$ will be defined. A *core structure* is specified by a subset $\mathbf{A}$ of the cliques $\widetilde{\mathbf{C}}$ of the triangulated version $\widetilde{\mathcal{G}}$. We make the following important assumption about the subgraph $\widetilde{\mathcal{G}}(\mathbf{A})$ of $\widetilde{\mathcal{G}}$ induced by the subset $\mathbf{A}$:

Assumption 6.2.1. The graph $\widetilde{\mathcal{G}}(\mathbf{A})$ induced by core structure $\mathbf{A}$ must be triangulated.

Example 6.2.1. To illustrate Assumption 6.2.1, consider the 3×3 grid illustrated in Figure 6.1(a). One possible triangulated version $\widetilde{\mathcal{G}}$ is shown in Figure 6.1(b). Shown in panel (c) is the graph $\widetilde{\mathcal{G}}(\mathbf{A})$ induced by the set of edges $\mathbf{A} = \mathcal{E}/\{(1,4),(4,7),(3,6)(6,9)\}$; it is a tree, and therefore satisfies the triangulation criterion of Assumption 6.2.1. In contrast, panel (d) shows the graph $\widetilde{\mathcal{G}}(\mathbf{A})$ induced by $\mathbf{A} = \mathcal{E}/\{(4,7),(3,6)(6,9)\}$, which fails Assumption 6.2.1.

Let $Q_{\mathbf{A}}(\mathbf{x})$ denote a distribution defined by potential functions on the cliques in the core set $\mathbf{A}$. The significance of Assumption 6.2.1 is in guaranteeing, via the junction tree representation, that $Q_{\mathbf{A}}(\mathbf{x})$ factorizes as a product of local marginal distributions. Since the induced graph $\widetilde{\mathcal{G}}(\mathbf{A})$ is triangulated, it has an associated junction tree [121]. Let $\mathbf{C}_{\max}(\mathbf{A})$ denote the set of maximal cliques in this junction tree, and let $\mathbf{C}_{\text{sep}}(\mathbf{A})$ be the corresponding set of separator sets. (See Section 2.1.5 for background on the junction tree representation). We then have the factorized representation

$$Q_{\mathbf{A}}(\mathbf{x}) = \frac{\prod_{C \in \mathbf{C}_{\max}(\mathbf{A})} Q(\mathbf{x}_C)}{\prod_{C \in \mathbf{C}_{\text{sep}}(\mathbf{A})} Q(\mathbf{x}_C)} \tag{6.5}$$

of the *core approximating distribution* (CAD). As we will see, this local product representation is the key to being able to optimize efficiently the choice of approximating distribution $Q_{\mathbf{A}}$. The implicit assumption here is that in contrast to the target distribution on the graph $\mathcal{G}$, the junction tree representation of $Q_{\mathbf{A}}$ over $\widetilde{\mathcal{G}}(\mathbf{A})$ is manageable (i.e., the maximal cliques are small enough so that exact inference is tractable).

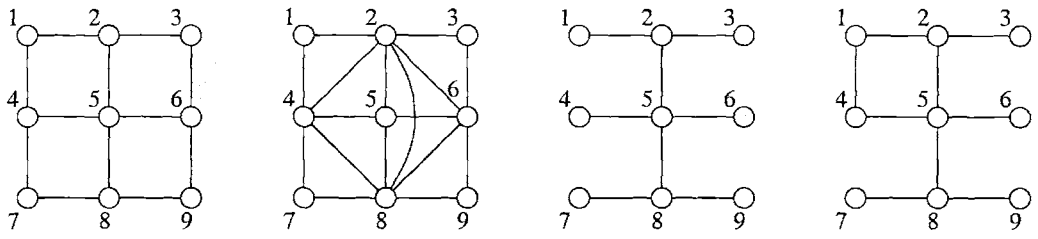


Figure 6.1. Illustration of triangulated and non-triangulated induced graphs $\widetilde{\mathcal{G}}(\mathbf{A})$. (a) Original graph $\mathcal{G}$ is the 3×3 grid. (b) One possible triangulated version $\widetilde{\mathcal{G}}$ of $\mathcal{G}$. Note the two 4-cliques $\{2,4,5,8\}$ and $\{2,5,6,8\}$ at the center of the graph. (c) The graph $\widetilde{\mathcal{G}}(\mathbf{A})$ induced by the core structure $\mathbf{A} = \mathcal{E}/\{(1,4),(4,7),(3,6)(6,9)\}$. It is a tree, and therefore triangulated. (d) The graph $\mathcal{G}(\mathbf{A})$ induced by the core structure $\mathbf{A} = \mathcal{E}/\{(4,7),(3,6)(6,9)\}$. It is no longer triangulated, since the 4-cycle $1-2-5-4-1$ lacks a chord. This problem can be rectified by adding the 3-clique $\{1,2,4\}$ that appears in $\widetilde{\mathcal{G}}$.

Similarly, we let $P_{\mathbf{A}}$ be a distribution formed over the core structure by a product over those compatibility functions of the target distribution of equation (6.4) contained within the core:

$$P_{\mathbf{A}}(\mathbf{x}) \propto \prod_{C \in \mathbf{A} \cap \mathbf{C}} \psi_C(\mathbf{x}) \tag{6.6}$$

Herein we refer to this distribution as the *core of the target distribution* (CTD). Note that the single node marginals associated with the CTD will, in general, be different than the single node marginals of the target distribution.

We illustrate these definitions with a few examples:

Example 6.2.2. Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with pairwise maximal cliques, in which case the set of all cliques $\mathbf{C}$ is equal to $\mathcal{V} \cup \mathcal{E}$.

(a) Let $\mathbf{A} = \mathcal{V}$, so that $\widetilde{\mathcal{G}}(\mathbf{A})$ is a completely disconnected graph. In this case, both the CAD and CTD are fully factorized:

$$\begin{aligned} Q_{\mathbf{A}}(\mathbf{x}) &= \prod_{s \in \mathcal{V}} Q(x_s) \\ P_{\mathbf{A}}(\mathbf{x}) &\propto \prod_{s \in \mathcal{V}} \psi_s(x_s) \end{aligned}$$

As we have mentioned and will be made explicit in Example 6.3.2, this choice of a fully factorized CAD corresponds to belief propagation.

(b) Let $\mathbf{A} = \mathcal{V} \cup \mathcal{E}(\mathcal{T})$ where $\mathcal{E}(\mathcal{T}) \subset \mathcal{E}$ is the edge set of an embedded tree. In this case, $\widetilde{\mathcal{G}}(\mathbf{A})$ corresponds to this particular embedded tree. The CAD is tree-structured, and factorizes as a product of marginals over the nodes and edges in $\mathcal{E}(\mathcal{T})$:

$$Q_{\mathbf{A}}(\mathbf{x}) = \prod_{s \in \mathcal{V}} Q(x_s) \prod_{(s,t) \in \mathcal{E}(\mathcal{T})} \frac{Q(x_s, x_t)}{Q(x_s)Q(x_t)}$$

The CTD also factorizes into a product of vertex and edge terms:

$$P_{\mathbf{A}}(\mathbf{x}) \propto \prod_{s \in \mathcal{V}} \psi_s(x_s) \prod_{(s,t) \in \mathcal{E}(\mathcal{T})} \psi_{st}(x_s, x_t)$$

■ 6.2.3 Residual partition

The second key element is the *residual set* — namely, the subset $\mathbf{C}/(\mathbf{A} \cap \mathbf{C}) \subset \mathbf{C}$ of cliques in $\mathbf{C}$ *not* covered by elements of the core set $\mathbf{A}$. (Note that since $\mathbf{A}$ is a subset of the clique set $\widetilde{\mathbf{C}}$ of the triangulated version $\widetilde{\mathcal{G}}$, it may not be a subset of $\mathbf{C}$.) We use the notation Δ to denote a particular subset of those cliques in the residual set.

Definition 6.2.1. Let $\mathbf{R}$ denote a partition of the residual set into a collection of subsets $\{\Delta_a\}$ (which need not be disjoint). I.e., the union $\cup_a \Delta_a$ is equal to the residual set. Such a decomposition is called a *residual partition.*

Example 6.2.3. We illustrate with a continuation of Example 6.2.2:

(a) In Example 6.2.2(a), the vertex set was chosen as the core structure ($\mathbf{A} = \mathcal{V}$). In this case, the residual set is given by $\mathbf{C}/\mathcal{V} = \mathcal{E}$ — that is, the set of all edges in the graph. These edges can be partitioned into subsets Δ in a variety of ways. The simplest choice is for each Δ to be a single edge $(s,t) \in \mathcal{E}$. The union $\mathbf{A} \cup \mathbf{R} = \mathcal{V} \cup \mathcal{E}$ covers the set of all cliques $\mathbf{C}$.

(b) Consider again Example 6.2.2(b), in which $\mathbf{A} = \mathcal{V} \cup \mathcal{E}(\mathcal{T})$ for some embedded tree $\mathcal{T}$. Here the residual set corresponds to those edges in $\mathcal{G}$ but not included in the tree (i.e., the set of edges $\mathcal{E}/\mathcal{E}(\mathcal{T})$). Again, the simplest partition of this residual set is into single edge terms (i.e., $\Delta = (s,t)$). The union $\mathbf{A} \cup \mathbf{R} = \mathcal{V} \cup \mathcal{E}(\mathcal{T}) \cup (\mathcal{E}/\mathcal{E}(\mathcal{T}))$ covers the clique set $\mathbf{C}$.

Figure 6.2 illustrates a particular case of this decomposition. Shown in (a) is the original graph $\mathcal{G}$ — here a 3×3 grid. Panel (b) shows the spanning tree induced by $\mathbf{A} = \mathcal{V} \cup \mathcal{E}(\mathcal{T})$, whereas panel (c) shows the residual set of edges in $\mathcal{E}$, but not in $\mathcal{E}(\mathcal{T})$.

Note that the core and residual structures in these examples satisfy an important property that we will always impose:

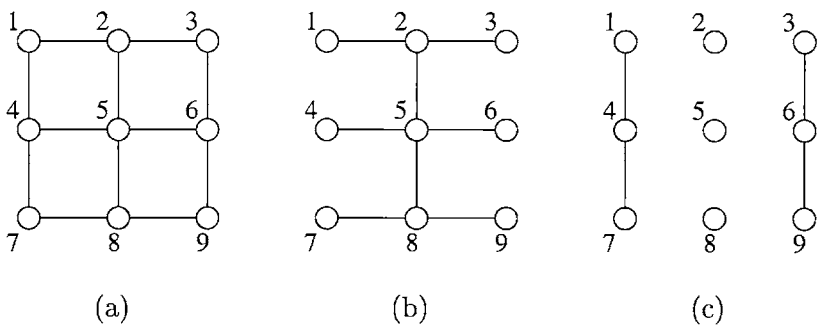


Figure 6.2. Illustration of core and residual structures. (a) Original graph (3×3 grid). (b) Spanning tree core ($\mathbf{A} = \mathcal{V} \cup \mathcal{E}(\mathcal{T})$). (c) Residual set $\mathcal{E}/\mathcal{E}(\mathcal{T})$ of edges not covered by the spanning tree.

Assumption 6.2.2. The union of the cliques in the core and residual sets covers the clique set (i.e., $\mathbf{A} \cup \mathbf{R} \supseteq \mathbf{C}$).

Note that the set of cliques $\mathbf{C}$ can be a strict subset of $\mathbf{A} \cup \mathbf{R}$, since the core structure $\mathbf{A}$ can include cliques in $\widetilde{\mathbf{C}}/\mathbf{C}$. Assumption 6.2.2 will play an important role in later analysis.

■ 6.2.4 Auxiliary distributions

In equations (6.5) and (6.6), we defined two distributions $Q_{\mathbf{A}}$ and $P_{\mathbf{A}}$ that were structured according to the core set $\mathbf{A}$. Similarly, given any element $\Delta \in \mathbf{R}$, it will be useful to define distributions structured according to the augmented set $\mathbf{A} \cup \Delta$. As an analog to $P_{\mathbf{A}}$, we define the distribution

$$P_{\mathbf{A} \cup \Delta}(\mathbf{x}) \propto \prod_{C \in \mathbf{A} \cup \Delta} \psi_C(\mathbf{x}) \tag{6.7}$$

as a normalized product of compatibility functions over cliques in the augmented set $\mathbf{A} \cup \Delta$.

In a similar fashion, we let $Q_{\mathbf{A} \cup \Delta}$ be an approximating distribution structured according to the cliques in the augmented set. To give an explicit expression for $Q_{\mathbf{A} \cup \Delta}$ is a bit more subtle: in particular, it requires that we consider a triangulated version of $\mathcal{G}(\mathbf{A} \cup \Delta)$.

Definition 6.2.2. For each $\Delta \in \mathbf{R}$, an *augmented residual set* $\widetilde{\Delta} \supseteq \Delta$ is a subset of cliques in $\widetilde{\mathbf{C}}$ such that the induced graph $\widetilde{\mathcal{G}}(\mathbf{A} \cup \widetilde{\Delta})$ is triangulated.

Although the choice of this augmented residual set is not necessarily unique, we shall assume that a particular choice is made and fixed.

Given an augmented residual set, we can exploit the junction tree representation of Section 2.1.5 to decompose the auxiliary distribution as follows:

$$Q_{\mathbf{A} \cup \Delta}(\mathbf{x}) = \frac{\prod_{C \in \mathbf{C}_{\max}(\mathbf{A} \cup \widetilde{\Delta})} \widetilde{Q}(\mathbf{x}_C)}{\prod_{C \in \mathbf{C}_{\text{sep}}(\mathbf{A} \cup \widetilde{\Delta})} \widetilde{Q}(\mathbf{x}_C)} \tag{6.8}$$

Here we use the notation $\widetilde{Q}$ to distinguish these local marginals from those in the definition of the core distribution $Q_{\mathbf{A}}$ in equation (6.5).

To illustrate, we continue with Example 6.2.3:

Example 6.2.4.

(a) In Example 6.2.3(a), the core set is $\mathbf{A} = \mathcal{V}$; the residual set given by the edge set $\mathcal{E}$; and the residual set is partitioned into individual edges (i.e., $\Delta = (u, v) \in \mathcal{E}$). In this case, the augmented graph $\mathbf{A} \cup \Delta = \mathcal{V} \cup (u, v)$ remains triangulated, so that there is no need to augment Δ. The auxiliary distributions are given by:

$$P_{\mathbf{A} \cup \Delta}(\mathbf{x}) \propto \psi_{uv}(x_u, x_v) \prod_{s \in \mathcal{V}} \psi_s(x_s) \tag{6.9a}$$

$$Q_{\mathbf{A} \cup \Delta}(\mathbf{x}) = \frac{\widetilde{Q}(x_u, x_v)}{\widetilde{Q}(x_u)\widetilde{Q}(x_v)} \prod_{s \in \mathcal{V}} \widetilde{Q}(x_s) \tag{6.9b}$$

(b) In Example 6.2.3(b), the core set is $\mathbf{A} = \mathcal{V} \cup \mathcal{E}(\mathcal{T})$, where $\mathcal{E}(\mathcal{T})$ is the edge set of an embedded tree $\mathcal{T}$. (See Figure 6.2(b)). As in (a), we partition the residual set $\mathcal{E}/\mathcal{E}(\mathcal{T})$ into individual edges ($\Delta = (u, v)$). Consider the augmented set

$$\mathbf{A} \cup \Delta = \mathcal{V} \cup \mathcal{E}(\mathcal{T}) \cup (u, v)$$

In this case, the auxiliary distribution $P_{\mathbf{A} \cup \Delta}$ has the form

$$P_{\mathbf{A} \cup \Delta}(\mathbf{x}) \propto \prod_{s \in \mathcal{V}} \psi_s(x_s) \prod_{(s,t) \in \mathcal{E}(\mathcal{T}) \cup \{(u,v)\}} \psi_{st}(x_s, x_t)$$

Now if $\mathcal{T}$ is a spanning tree, then adding an extra edge will add a cycle to the graph. In this case, since we assumed that $\mathcal{G}$ has pairwise maximal cliques, the augmented set will no longer be triangulated. We therefore need to augment Δ to form $\widetilde{\Delta}$ so that $\widetilde{\mathcal{G}}(\mathbf{A} \cup \widetilde{\Delta})$ is triangulated.

To provide a concrete illustration of this augmentation procedure, consider the 3×3 grid illustrated in Figure 6.2(a). Let us add edge $(1, 4)$ to the spanning tree shown in Figure 6.2(b); we have drawn the corresponding subgraph $\mathcal{G}(\mathbf{A} \cup (1, 4))$ in Figure 6.3(a). This subgraph is not triangulated, since the 4-cycle $1-2-5-4-1$ lacks a chord. Therefore, we form $\widetilde{\Delta}$ by adding the chord $(2, 4)$ to $\Delta = (1, 4)$ to obtain the triangulated subgraph shown in Figure 6.3(b).

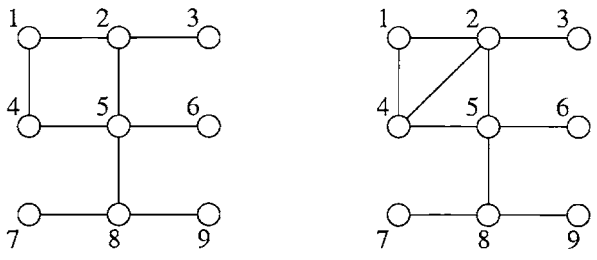


Figure 6.3. Augmented subgraph and its triangulation for 3×3 grid. (a) Augmented subgraph $\mathbf{A} \cup \Delta$ formed by adding a single edge $\Delta = (1, 4)$ to the spanning tree core set. (b) Triangulated version $\mathbf{A} \cup \tilde{\Delta}$ of the augmented subgraph. The edge $(2, 4)$ must be added to triangulate the graph.

Now the maximal cliques of this triangulated graph are given by

$$\{(124), (245), (23), (56), (58), (78), (89)\}$$

By the junction tree representation, we can decompose any auxiliary distribution $Q_{\mathbf{A} \cup \Delta}$ as:

$$Q_{\mathbf{A} \cup \Delta} = \frac{\widetilde{Q}_{124}\, \widetilde{Q}_{245}\, \widetilde{Q}_{23}\, \widetilde{Q}_{56}\, \widetilde{Q}_{58}\, \widetilde{Q}_{78}\, \widetilde{Q}_{89}}{\widetilde{Q}_{24}\, \widetilde{Q}_{2}\, (\widetilde{Q}_{5})^2\, (\widetilde{Q}_{8})^2}$$

where we have omitted the explicit dependence of Q on $\mathbf{x}$ for notational simplicity.

■ 6.2.5 Marginalization operators

Recall the definitions of the core approximating distribution (CAD) $Q_{\mathbf{A}}$ and auxiliary distribution $Q_{\mathbf{A} \cup \Delta}$:

$$Q_{\mathbf{A}}(\mathbf{x}) = \frac{\prod_{C \in \mathbf{C}_{\max}(\mathbf{A})} Q(\mathbf{x}_C)}{\prod_{C \in \mathbf{C}_{\text{sep}}(\mathbf{A})} Q(\mathbf{x}_C)} \tag{6.10a}$$

$$Q_{\mathbf{A} \cup \Delta} = \frac{\prod_{C \in \mathbf{C}_{\max}(\mathbf{A} \cup \tilde{\Delta})} \widetilde{Q}(\mathbf{x}_C)}{\prod_{C \in \mathbf{C}_{\text{sep}}(\mathbf{A} \cup \tilde{\Delta})} \widetilde{Q}(\mathbf{x}_C)} \tag{6.10b}$$

Note that both of these distributions are defined in terms local marginal distributions over subsets of cliques and separator sets. A key constraint is that the local marginals defining the auxiliary distribution must agree with those defining the core distribution, whenever they overlap. In this section, we define *marginalization operators* that will be used to enforce these constraints.

To be precise, for any $\Delta \in \mathbf{R}$, we define a set of clique pairs as follows:

$$\mathfrak{P}(\Delta) \triangleq \{ (\mathcal{C}, \mathcal{D}) \mid \mathcal{C} \in \mathbf{C}(\mathbf{A}); \;\; \mathcal{D} \in \mathbf{C}(\mathbf{A} \cup \widetilde{\Delta}) \quad \text{s.t } \mathcal{C} \subset \mathcal{D} \} \tag{6.11}$$

For any pair $(\mathcal{C}, \mathcal{D}) \in \mathfrak{P}(\Delta)$, let $Q_{\mathcal{C}}$ and $\widetilde{Q}_{\mathcal{D}}$ be the corresponding local marginals in the definitions of $Q_{\mathbf{A}}$ and $Q_{\mathbf{A} \cup \Delta}$ respectively. For any such pair, we require that

$$\sum_{\mathbf{x}'_{\mathcal{D}} \text{ s.t } \mathbf{x}'_{\mathcal{C}} = \mathbf{x}_{\mathcal{C}}} \widetilde{Q}_{\mathcal{D}}(\mathbf{x}'_{\mathcal{D}}) = Q_{\mathcal{C}}(\mathbf{x}_{\mathcal{C}})$$

I.e., the quantity $\widetilde{Q}_{\mathcal{D}}$, when marginalized down, agrees with $Q_{\mathcal{C}}$. We write this equation compactly as

$$\mathbb{M}(\widetilde{Q}_{\mathcal{D}}) = Q_{\mathcal{C}} \tag{6.12}$$

where $\mathbb{M}$ is a marginalization operator. We write $\mathbb{M}(Q_{\mathbf{A} \cup \Delta}) = Q_{\mathbf{A}}$ to mean that equation (6.12) holds for all pairs $(\mathcal{C}, \mathcal{D}) \in \mathfrak{P}(\Delta)$.

Example 6.2.5. We continue with Example 6.2.4 (a), in which $\mathbf{A} = \mathcal{V}$, and each residual term Δ consisted of a single edge (u, v). This gave rise to an auxiliary distribution of the form:

$$Q_{\mathbf{A} \cup \Delta}(\mathbf{x}) = \frac{\widetilde{Q}(x_u, x_v)}{\widetilde{Q}(x_u)\widetilde{Q}(x_v)} \prod_{s \in \mathcal{V}} \widetilde{Q}(x_s)$$

The marginalization condition $\mathbb{M}(Q_{\mathbf{A} \cup \Delta}) = Q_{\mathbf{A}}$ consists of the following constraints:

$$\sum_{x'_u} \widetilde{Q}(x'_u, x_v) = Q(x_v) \tag{6.13a}$$

$$\sum_{x'_v} \widetilde{Q}(x_u, x'_v) = Q(x_u) \tag{6.13b}$$

$$\widetilde{Q}(x_s) = Q(x_s) \quad \forall \;\; s \in \mathcal{V} \tag{6.13c}$$

Note that equations (6.13a) and (6.13b) are identical to the pairwise marginalization constraints enforced in standard belief propagation.

■ 6.3 Approximations to the Kullback-Leibler divergence

In this section, we use the formalism and machinery described in Section 6.2 to develop a variety of approximations to the Kullback-Leibler (KL) divergence. It is simplest to do so when the residual partition satisfies a certain property (to be defined) related to disjointness. Capturing Kikuchi approximations [114, 180] requires the use of non-disjoint residual partitions, which are conceptually similar but more complex in terms of notation.

Each of the approximations to be considered here is specified by a particular choice of the core set $\mathbf{A}$ and residual partition $\mathbf{R}$. The central quantity is a cost function $\mathcal{G}_{\mathbf{A};\mathbf{R}}(\vec{\mathbf{Q}})$ that depends a collection of approximating distributions $\vec{\mathbf{Q}}$. The notion of exactness is defined as follows:

Definition 6.3.1. The approximation is said to be *exact* if there exists a distribution q over $\mathcal{G}$ that marginalizes down to the local marginals defining $\vec{\mathbf{Q}}$ such that $\mathcal{G}_{\mathbf{A};\mathbf{R}}(\vec{\mathbf{Q}})$ is equal to the Kullback-Leibler divergence $D(q \,\|\, p)$ (aside from constants not dependent on $\vec{\mathbf{Q}}$ or q).

As we will see, the Bethe free energy corresponds to a special case of a $\mathcal{G}_{\mathbf{A};\mathbf{R}}(\vec{\mathbf{Q}})$ approximation, one which is exact for a tree-structured graph. Of course, the more general and interesting case will be when the function $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ is only an approximation to the Kullback-Leibler divergence.

■ 6.3.1 Disjoint and non-disjoint residual partitions

We first define the notion of disjoint and non-disjoint residual partitions. One might define the residual partition $\mathbf{R}$ to be pairwise disjoint if Δ_a and Δ_b are disjoint for all $\Delta_a, \Delta_b \in \mathbf{R}$. It turns out to be necessary to define disjointness at the level of the augmented residual sets $\widetilde{\Delta}$ specified in Definition 6.2.2. We denote the full collection of these augmented residual sets as follows:

$$\widetilde{\mathbf{R}} \triangleq \{\widetilde{\Delta} \mid \Delta \in \mathbf{R}\} \tag{6.14}$$

With this notation, we have:

Definition 6.3.2. A residual partition $\mathbf{R}$ is *pairwise disjoint* if $\widetilde{\Delta}_a \cap \widetilde{\Delta}_b = \emptyset$ for all distinct $\widetilde{\Delta}_a, \widetilde{\Delta}_b$ in the associated augmented residual set $\widetilde{\mathbf{R}}$. Otherwise, it is non-disjoint.

Example 6.3.1. To illustrate Definition 6.3.2, consider the 2-square graph $\mathcal{G}$ shown in Figure 6.4(a), as well as the triangulated version $\widetilde{\mathcal{G}}$ shown in panel (b). As the core structure, we choose the embedded spanning tree shown in panel (c). We partition the residual set into the two edge terms $\Delta_1 = (3,4)$ and $\Delta_2 = (5,6)$, which are disjoint. Adding the first term Δ_1 to the core $\mathbf{A}$ gives rise to the augmented structure $\mathbf{A} \cup \Delta_1$ shown in panel (d). Here we need augment Δ_1 to form $\widetilde{\Delta}_1 = \{\Delta_1, (1,4)\}$ in order to triangulate the graph. Similarly, adding Δ_2 yields the augmented structure shown in panel (e); here we need to form $\widetilde{\Delta}_2 = \{\Delta_2, (1,4), (3,4), (3,6)\}$ in order to triangulate the graph. Thus, $\widetilde{\Delta}_1 \cap \widetilde{\Delta}_2 = (1,4)$, so that the partition is not disjoint at the augmented level.

We will follow up Example 6.3.1 in Example 6.3.6 to demonstrate the necessity of defining disjointness at the level of the augmented residual sets.

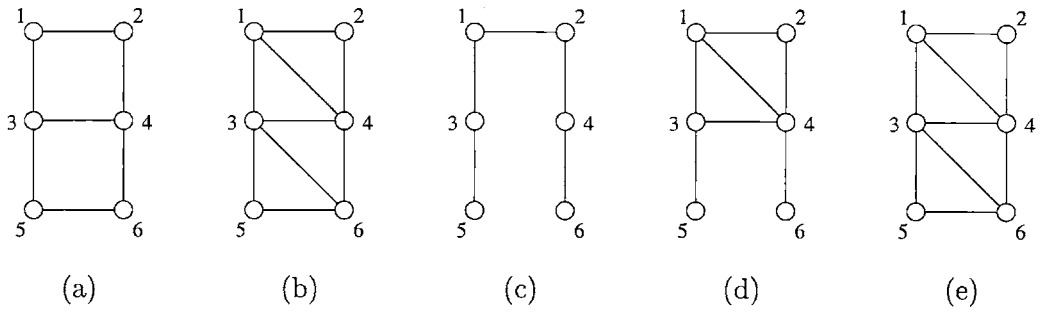


Figure 6.4. An example of disjointness at the Δ-level, but not at the $\widetilde{\Delta}$-level. (a) Original 2-square graph $\mathcal{G}$. (b) Triangulated version $\widetilde{\mathcal{G}}$. (c) Core structure of embedded spanning tree. (d) Augmented structure formed by adding $\Delta_1 = (3,4)$; edge $(1,4)$ is added to triangulate. (e) Augmented structure formed by adding edge $\Delta_2 = (5,6)$; here we must add edges $(3,6)$, $(3,4)$ and $(1,4)$ to triangulate.

■ 6.3.2 Approximation for a disjoint partition

In this section, we develop the basic form of the approximations for the case of a pairwise disjoint residual partition. (i.e., $\widetilde{\Delta}_a \cap \widetilde{\Delta}_b = \emptyset$ for all $\widetilde{\Delta}_a \neq \widetilde{\Delta}_b \in \widetilde{\mathbf{R}}$). Given such a residual partition $\mathbf{R}$ and a core set $\mathbf{A}$, we define:

$$\mathcal{G}_{\mathbf{A};\mathbf{R}}(\vec{\mathbf{Q}}) = D(Q_{\mathbf{A}} \parallel P_{\mathbf{A}}) + \sum_{\Delta \in \mathbf{R}} \Big\{ D(Q_{\mathbf{A} \cup \Delta} \parallel P_{\mathbf{A} \cup \Delta}) - D(Q_{\mathbf{A}} \parallel P_{\mathbf{A}}) \Big\} \tag{6.15}$$

The function $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ depends on the collection of distributions

$$\vec{\mathbf{Q}} \triangleq Q_{\mathbf{A}} \cup \{ Q_{\mathbf{A} \cup \Delta} \mid \Delta \in \mathbf{R} \}$$

where the core approximating distribution $Q_{\mathbf{A}}$ was defined in equation (6.5); and the auxiliary distributions $Q_{\mathbf{A} \cup \Delta}$ are defined in equation (6.8). The variational problem of interest is the following:

$$\begin{cases} \min \mathcal{G}_{\mathbf{A};\mathbf{R}}(\vec{\mathbf{Q}}) \\ \text{s.t} \quad \mathbb{M}(Q_{\mathbf{A} \cup \Delta}) = Q_{\mathbf{A}} \qquad \forall \Delta \in \mathbf{R} \end{cases} \tag{6.16}$$

To illustrate the case of a disjoint residual partition, we present a simple example that leads to the Bethe free energy [180] of belief propagation:

Example 6.3.2 (Bethe free energy). Consider the set-up of Example 6.2.5, where $\mathbf{A} = \mathcal{V}$; and the core and auxiliary distributions have the form:

$$\begin{aligned} Q_{\mathbf{A}}(\mathbf{x}) &= \prod_{s \in \mathcal{V}} Q(x_s) \\ Q_{\mathbf{A} \cup (u,v)}(\mathbf{x}) &= \frac{Q(x_u, x_v)}{Q(x_u)Q(x_v)} \prod_{s \in \mathcal{V}} Q(x_s) \\ P_{\mathbf{A}}(\mathbf{x}) &\propto \prod_{s \in \mathcal{V}} \psi_s(x_s) \\ P_{\mathbf{A} \cup (u,v)}(\mathbf{x}) &\propto \psi_{uv}(x_u, x_v) \prod_{s \in \mathcal{V}} \psi_s(x_s) \end{aligned}$$

Here we have dropped the distinction between Q and $\widetilde{Q}$ in defining $Q_{\mathbf{A} \cup (u,v)}$, since the marginalization constraints ensure that they are the same.

Substituting these relations into equation (6.15) yields, after some re-arrangements, the following cost function:

$$\begin{aligned} \mathcal{G}_{\mathbf{A};\mathbf{R}}(\vec{\mathbf{Q}}) = C + \sum_{s \in \mathcal{V}} \sum_{x_s} Q(x_s) \log \frac{Q(x_s)}{\psi_s(x_s)} \\ + \sum_{(s,t) \in \mathcal{E}} \sum_{x_s, x_t} Q(x_s, x_t) \log \Big[\frac{Q(x_s, x_t)}{Q(x_s)Q(x_t)} - \log \psi_{st}(x_s, x_t) \Big] \end{aligned} \quad (6.17)$$

where C is a constant independent of $\vec{\mathbf{Q}}$.

The first summation in equation (6.17), which arises from the KL divergence term $D(Q_{\mathbf{A}} \,\|\, P_{\mathbf{A}})$ in equation (6.15), can be viewed as a prior encouraging the fully factorized $Q_{\mathbf{A}}$ to be close to the fully factorized $P_{\mathbf{A}}$. In addition to this prior, each edge $(s,t) \in \mathcal{E}$ contributes a term to the second summation, which couples adjacent pairs of random variables x_s and x_t.

In this specific context, the variational problem (6.16) assumes the following form: minimize the cost functional of equation (6.17) as a function of $\{Q_s,\ Q_{st}\}$, subject to the marginalization constraints derived in Example 6.2.5:

$$\begin{aligned} \sum_{x'_s} Q(x'_s, x_t) &= Q(x_t) \\ \sum_{x'_t} Q(x_s, x'_t) &= Q(x_s) \end{aligned}$$

for all node pairs $(s,t) \in \mathcal{E}$.

The functional of equation (6.17), aside from the additive constant, is equivalent to the Bethe free energy [180]. Note that with the exception of tree-structured graphs,

the function $\mathcal{G}_{\mathbf{A};\mathbf{R}}(\vec{\mathbf{Q}}) \neq D(q \parallel p) + C$. Herein arises the primary source of error in the approximation.[1]

As shown by Yedidia et al. [180], the belief propagation (BP) algorithm is a particular technique for attempting to minimize the Bethe free energy subject to the marginalization constraints associated with variational problem (6.16). Overall, these relations illustrate that BP, as a technique for attempting to minimize the $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ functional with a fully factorized core set, can be viewed as a sequence of updates to a fully factorized distribution $Q_{\mathbf{A}}$. This fact has been pointed out by a number of researchers [e.g., 77, 131, 132, 147].

■ 6.3.3 Approximation for a non-disjoint partition

When the partition $\mathbf{R}$ is no longer pairwise disjoint, a minor modification to the cost function of equation (6.15) is required. In particular, for a non-disjoint partition, an unmodified $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ would count each element that appears in a non-empty intersection $\widetilde{\Delta}_a \cap \widetilde{\Delta}_b$ more than once. It is therefore necessary to subtract off terms corresponding to these intersections. If all the higher-order intersections $\widetilde{\Delta}_a \cap \widetilde{\Delta}_b \cap \widetilde{\Delta}_c$ are empty, then we are finished; otherwise, we need to add back in triplet terms. The basic principle at work here is that of *inclusion-exclusion* [168].

When there are at most pairwise intersections among elements $\widetilde{\Delta}$ of the augmented residual partition $\widetilde{\mathbf{R}}$, we define the following family of cost functions:

$$\mathcal{G}_{\mathbf{A};\mathbf{R}}(\vec{\mathbf{Q}}) = D(Q_{\mathbf{A}} \parallel P_{\mathbf{A}}) + \sum_{\Delta \in \mathbf{R}} \Big\{ D(Q_{\mathbf{A} \cup \Delta} \parallel P_{\mathbf{A} \cup \Delta}) - D(Q_{\mathbf{A}} \parallel P_{\mathbf{A}}) \Big\} - \sum_{\widetilde{\Delta}_a \cap \widetilde{\Delta}_b \neq \emptyset} \Big\{ D(Q_{\mathbf{A} \cup \Delta_a \cap \Delta_b} \parallel P_{\mathbf{A} \cup \Delta_a \cap \Delta_b}) - D(Q_{\mathbf{A}} \parallel P_{\mathbf{A}}) \Big\} \quad (6.18)$$

where the second sum ranges over all distinct pairs $\widetilde{\Delta}_a, \widetilde{\Delta}_b \in \widetilde{\mathbf{R}}$. For this non-disjoint partition, the notation $\vec{\mathbf{Q}}$ refers to the collection of distributions

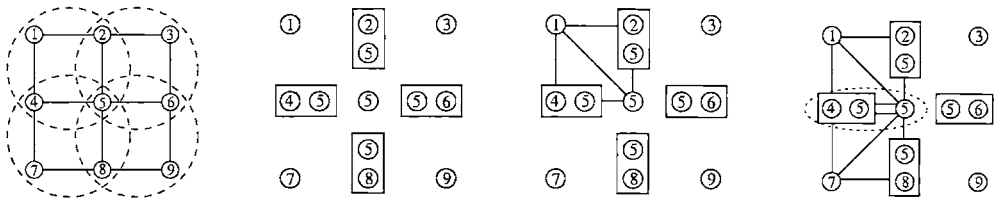
$$\vec{\mathbf{Q}} \triangleq Q_{\mathbf{A}} \cup \{ Q_{\mathbf{A} \cup \Delta} \mid \Delta \in \mathbf{R} \} \cup \{ Q_{\mathbf{A} \cup \Delta_a \cap \Delta_b} \mid \Delta_a, \Delta_b \in \mathbf{R}; \widetilde{\Delta}_a \cap \widetilde{\Delta}_b \neq \emptyset \} \quad (6.19)$$

It is clear that equation (6.18) can be further generalized to the case where higher-order intersections of residual sets are also non-empty. In particular, to include triplet terms, we would need to add back in terms involving $D(Q_{\mathbf{A} \cup \Delta_a \cap \Delta_b \cap \Delta_c} \parallel Q_{\mathbf{A}})$. In the interests of notational simplicity, we limit ourselves to at most pairwise intersections.

The associated variational problem is the following:

$$\begin{cases} \min \mathcal{G}_{\mathbf{A};\mathbf{R}}(\vec{\mathbf{Q}}) \\ \text{s.t} \quad \mathbb{M}(Q_{\mathbf{A} \cup \Delta_a \cap \Delta_b}) = Q_{\mathbf{A}} \qquad \forall\, \Delta_a \neq \Delta_b, \quad \text{s.t} \;\; \widetilde{\Delta}_a \cap \widetilde{\Delta}_b = \emptyset \\ \text{and} \quad \mathbb{M}(Q_{\mathbf{A} \cup \Delta}) = Q_{\mathbf{A}} \qquad \forall\, \Delta \in \mathbf{R} \end{cases} \quad (6.20)$$

[1] Another source of error is the possibility of obtaining local minima in solving the associated variational problem (6.16).



(a) (b) (c) (d)

Figure 6.5. 4-plaque Kikuchi approximation on a 3×3 grid. (a) 4-plaque clustering of original grid. (b) Fully disconnected core approximation on clustered variables. (c) Auxiliary structure formed by adding in the edges $\widetilde{\Delta}_1$ associated with the 4-plaque $\{1, 2, 4, 5\}$. (d) Let $\widetilde{\Delta}_2$ be the augmented residual set associated with the 4-plaque $\{4, 5, 7, 8\}$. Overlap between residual terms $\widetilde{\Delta}_1$ and $\widetilde{\Delta}_2$.

The following example illustrates a non-disjoint partition in the context of a Kikuchi approximation [114]:

Example 6.3.3 (Kikuchi approximation). We now consider a form of Kikuchi approximation known as 4-plaque clustering, as applied to a regular grid by Yedidia et al. [180]. In particular, the 4-plaques arise from clustering the nodes of a 3×3 grid into groups of four, as shown in Figure 6.5(a). The associated core structure is the fully disconnected graph shown in Figure 6.5(b), whose nodes are formed by particular clusters of the original graph. In particular, we retain all the 2-node intersections between adjacent 4-plaques (shown in rectangular boxes), the 1-node intersections of these intersections, plus any remaining singleton nodes. In Figure 6.5(b), node 5 is the single example of an intersection between the intersections (node pairs) of 4-plaques.

For this 3×3 grid, the residual set $\mathbf{R}$ consists of four terms, each of which corresponds to adding in the interactions associated with a particular 4-plaque. For instance, including the 4-plaque $\{1, 2, 4, 5\}$ is equivalent to adding in the set of edges

$$\Delta_1 = \{(1, 25), (1, 45), (5, 25), (5, 45)\} \tag{6.21}$$

where the notation (s, tu) denotes the edge between node s and the clustered node $\{t, u\}$. However, so that the induced graph is triangulated, we need to augment this residual set to $\widetilde{\Delta}_1 = \{\Delta_1, (1, 5)\}$; the resulting triangulated graph $\widetilde{\mathcal{G}}(\mathbf{A} \cup \widetilde{\Delta}_1)$ is shown in Figure 6.5(c). Here it should be understood that (for example) the edge between $\{4, 5\}$ and 5 means that the two random variables labeled with 5 are equated. The pairwise constraint ψ_{45} between x_4 and x_5 is incorporated when defining the cluster $\{4, 5\}$.

The partition is not disjoint, since for example the residual sets Δ_1 defined in equation (6.21) and the residual set

$$\Delta_2 \triangleq \{(5, 45), (5, 58), (7, 58), (7, 45)\} \tag{6.22}$$

associated with the 4-plaque $\{4, 5, 7, 8\}$ have a non-empty intersection $(5, 45)$. This non-empty intersection is represented by the dotted ellipse in Figure 6.5(d). For this reason, it is necessary to use the modified cost function of equation (6.18). This cost function is equivalent (aside from constants not dependent on $\vec{\mathbf{Q}}$) to the Kikuchi approximation used by Yedidia et al. [180].

Thus, like the Bethe free energy of belief propagation in Example (6.3.2), this Kikuchi approximation corresponds to using a fully factorized core distribution, albeit one defined on a graph formed by clustering nodes.

■ 6.3.4 Properties of the approximation

In what sense do the cost functions of equations (6.15) and (6.18) constitute approximations to the Kullback-Leibler divergence? Moreover, what factors govern their accuracy? In this section, we establish a number of properties that help to answer these questions.

Suppose that we are given a target distribution p of the form in equation (6.4), and that q is an approximation to p. We split the KL divergence between q and p into three terms:

$$D(q \parallel p) = -H(q) - \sum_{\mathbf{x} \in \mathcal{X}^N} q(\mathbf{x}) \sum_{C \in \mathbf{C}} \log \psi_C(\mathbf{x}) + \log Z(p) \tag{6.23}$$

Here $H(q) = -\sum_{\mathbf{x} \in \mathcal{X}^N} q(\mathbf{x}) \log q(\mathbf{x})$ is the entropy of q. Following the terminology of statistical physics [135], we shall refer to the second term as the *average energy*. The log partition function $\log Z(p)$ is a constant independent of q, so that we can ignore it. We shall show that the cost functions of equations (6.15) and (6.18) both treat the energy term exactly, whereas the treatment of the entropy term, in contrast, is usually approximate.

Lemma 6.3.1. If $\mathbf{A} \cup \mathbf{R} \supseteq \mathbf{C}$ (i.e., Assumption 6.2.2 holds), then the cost functions of equation (6.15) and equation (6.18) both capture the average energy term in equation (6.23) exactly.

Proof. We give the details of the proof for the disjoint residual partition of equation (6.15). Using the definition of $P_{\mathbf{A} \cup \Delta}$ in equation (6.7), it can be seen that each term $D(Q_{\mathbf{A} \cup \Delta} \parallel P_{\mathbf{A} \cup \Delta}) - D(Q_{\mathbf{A}} \parallel P_{\mathbf{A}})$ contributes energy terms of the form:

$$\sum_{\mathbf{x} \in \mathcal{X}^N} \sum_{C \in \Delta} Q(\mathbf{x}_C) \log \psi_C(\mathbf{x})$$

Including the terms contributed by $D(Q_{\mathbf{A}} \parallel P_{\mathbf{A}})$, we have:

$$\sum_{\mathbf{x} \in \mathcal{X}^N} \Big[\sum_{C \in \mathbf{A}} Q(\mathbf{x}_C) \log \psi_C(\mathbf{x}) + \sum_{\Delta \in \mathbf{R}} \sum_{C \in \Delta} Q(\mathbf{x}_C) \log \psi_C(\mathbf{x}) \Big] = \sum_{\mathbf{x} \in \mathcal{X}^N} \sum_{C \in \mathbf{C}} Q(\mathbf{x}_C) \log \psi_C(\mathbf{x})$$

where we have used Assumption 6.2.2 and the fact that the partition is pairwise disjoint. Thus, the average energy term is treated exactly.

The non-disjoint partition of equation (6.18) can be treated similarly. Here the second summation corrects the overcounting induced by the non-disjoint residual sets. □

While the energy is always captured exactly, the entropy term is usually treated in only an approximate manner. The nature of the approximation depends on the relation between the core structure $\mathbf{A}$ and partition $\mathbf{R}$, and the structure of the original graph. There are some straight-forward situations in which the cost function will be exact in the sense of Definition 6.3.1:

(a) for any graph, the approximation is exact whenever the core structure corresponds to the full graph (so that the residual set is empty).

(b) for any graph and core structure, the approximation is exact whenever the residual set is partitioned only into a single element Δ. For instance, using a spanning tree core on a single loop will always yield an exact result.

Neither of these cases are interesting, since such choices of $(\mathbf{A}; \mathbf{R})$ mean that computing the cost function $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ is as costly as computing the original KL divergence. However, there are tractable and non-trivial choices of $\mathbf{A}$ and $\mathbf{R}$ for which the cost function is still exact. Important cases include a fully factorized core approximation applied to tree-structured graph (to be discussed in the following paragraph), and generalizations of this case (which are covered by Proposition 6.3.1 below). The exactness, and more generally the relative accuracy of the $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ approximation, depends on an interaction between the structures of the core, residual set, and full graph.

Interestingly, the core structure need not cover any significant portion of the full clique set $\widetilde{\mathbf{C}}$ in order for the approximation to be exact. This property is best illustrated by Example 6.3.2, where the core structure is the vertex set $\mathcal{V}$, and the residual partition is formed of individual edges ($\Delta = (u, v)$). When the underlying graph $\mathcal{G}$ is tree-structured, the cost function of equation (6.17) is equivalent to the KL divergence. This exactness holds despite the gap between the core structure ($\mathcal{V}$) and the set of cliques of the full graph ($\mathcal{V} \cup \mathcal{E}$). The key property turns out to be whether or not the core structure and residual set cover the maximal cliques of a triangulated version of $\mathcal{G}$. With reference to our example, a tree is already triangulated and its maximal cliques correspond to the edges; hence, the set $\mathcal{V} \cup \mathcal{E}$ trivially contains all the maximal cliques.

This observation can be suitably generalized as follows:

Proposition 6.3.1. Let $\widetilde{\mathbf{C}}$ be the clique set of a triangulated version $\widetilde{\mathcal{G}}$ of the original graph. For a disjoint residual partition, the $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ function of equation (6.15) is equivalent to the KL divergence in the sense of Definition 6.3.1 if and only if $\mathbf{A} \cup \widetilde{\mathbf{R}} = \widetilde{\mathbf{C}}$.

Proof. If $\mathbf{A} \cup \widetilde{\mathbf{R}} = \widetilde{\mathbf{C}}$, then the function $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ will include a term for every maximal clique and separator set in a junction tree corresponding to $\widetilde{\mathbf{C}}$. Moreover, the disjointness of the residual partition ensures that it includes the correct number of such terms, as specified by the junction tree representation of $q(\mathbf{x})$.

Conversely, if $\mathbf{A} \cup \widetilde{\mathbf{R}}$ is a strict subset of $\widetilde{\mathbf{C}}$, then there is some maximal clique C^* in $\widetilde{\mathbf{C}}$ not covered by $\mathbf{A} \cup \widetilde{\mathbf{R}}$. By the junction tree representation of $q(\mathbf{x})$, the entropy $H(q)$ will be a function of the local marginal distribution $Q(\mathbf{x}_{C^*})$. This dependence will not be captured by the $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ function of equation (6.15); hence, it will in general be only an approximation to the KL divergence. □

■ 6.3.5 Illustrative examples

In this section, we consider a number of illustrative examples to develop further insight.

Example 6.3.4 (Exactness with a disjoint partition). To provide an illustration of Proposition 6.3.1, consider the simple 2-square graph $\mathcal{G}$ shown in Figure 6.6(a). Panel (b) shows a particular triangulated version $\widetilde{\mathcal{G}}$. As the core structure $\mathbf{A}$, we choose the edges (and vertices) corresponding to the spanning tree shown in Figure 6.6(c). We then partition the residual set into the two edges $\Delta_1 = (1,2)$ and $\Delta_2 = (5,6)$. With these choices, Minka [131] showed that his expectation-propagation algorithm would yield an exact result. We shall also establish an algorithm-independent exactness — in particular, by showing that in this case, the cost function $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ is equivalent to the Kullback-Leibler divergence. Therefore, the approximation is exact in the sense of Definition 6.3.1, and any technique for solving the associated variational problem (6.16) will yield the exact marginals of the target distribution p.

We first demonstrate exactness by recourse to Proposition 6.3.1. We begin by considering the augmented structures $\mathbf{A} \cup \Delta_i$, $i = 1, 2$. We form the augmented edge sets $\widetilde{\Delta}_1 = \{\Delta_1, (1,4)\}$ and $\widetilde{\Delta}_2 = \{\Delta_2, (3,6)\}$ so that the respective induced subgraphs $\widetilde{\mathcal{G}}(\mathbf{A} \cup \widetilde{\Delta}_i)$, $i = 1, 2$, illustrated in Figures 6.6(d) and (e) respectively, are triangulated. It is not difficult to see that the set $\mathbf{A} \cup \widetilde{\mathbf{R}}$ covers the clique set $\widetilde{\mathbf{C}}$ of the triangulated version $\widetilde{\mathcal{G}}$ shown panel (b). Therefore, Proposition 6.3.1 assures that the $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ function is an exact representation of the KL divergence in the sense of Definition 6.3.1.

It provides additional insight to demonstrate this exactness in a constructive fashion. According to the junction tree representation applied to the triangulated version $\widetilde{\mathcal{G}}$ of (b), any distribution $q(\mathbf{x})$ that is Markov with respect to $\mathcal{G}$ factorizes as:

$$q(\mathbf{x}) = \frac{Q_{124} Q_{134} Q_{346} Q_{356}}{Q_{14} Q_{34} Q_{36}} \tag{6.24}$$

where we have omitted the explicit dependence of the Q terms on $\mathbf{x}$ for notational simplicity. The terms in the numerator of equation (6.24) correspond to maximal cliques of a junction tree given by $\widetilde{\mathcal{G}}$, whereas the terms in the denominator correspond to separator sets.

Next, any core distribution over the tree shown in Figure 6.6(c) must factorize as

$$Q_{\mathbf{A}} = \frac{Q_{13} Q_{24} Q_{34} Q_{35} Q_{46}}{Q_3^2 Q_4^2} \tag{6.25}$$

Now consider the auxiliary distribution $Q_{\mathbf{A} \cup \Delta_1}$ defined on the graph in panel (d); by applying the junction tree representation to this triangulated graph, we are guaranteed

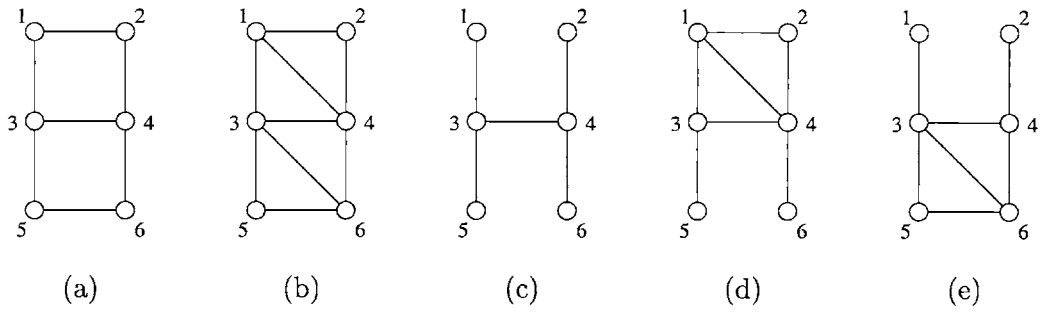


Figure 6.6. Non-trivial case where the $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ approximation is exact. (a) Original 2-square graph $\mathcal{G}$. (b) Triangulated version $\widetilde{\mathcal{G}}$. (c) Core structure is the spanning tree shown. Residual set is partitioned into $\Delta_1 = (1,2)$ and $\Delta_2 = (5,6)$; these sets are augmented to form $\widetilde{\Delta}_1$ and $\widetilde{\Delta}_2$. (e) Auxiliary structure $\mathbf{A} \cup \widetilde{\Delta}_1$. (e) Auxiliary structure $\mathbf{A} \cup \widetilde{\Delta}_2$.

that $Q_{\mathbf{A} \cup \Delta_1}$ factorizes as

$$Q_{\mathbf{A} \cup \Delta_1} = \frac{Q_{124} Q_{134} Q_{35} Q_{46}}{Q_{14} Q_3 Q_4} \tag{6.26}$$

Similarly, any auxiliary distribution $Q_{\mathbf{A} \cup \Delta_2}$ over the graph in Figure 6.6(e) factorizes as

$$Q_{\mathbf{A} \cup \Delta_2} = \frac{Q_{356} Q_{346} Q_{13} Q_{24}}{Q_{36} Q_3 Q_4} \tag{6.27}$$

Finally, we will show that the cost function $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ of equation (6.15) combines the terms $Q_{\mathbf{A}}$ and $Q_{\mathbf{A} \cup \Delta_i}$ in such a way so as to exactly represent the entropy of $q(\mathbf{x})$. In particular, using equation (6.15), the entropy terms of $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ are given by:

$$\sum_{\mathbf{x} \in \mathcal{X}^N} \left\{ Q_{\mathbf{A}}(\mathbf{x}) \log Q_{\mathbf{A}}(\mathbf{x}) + \sum_{i=1}^{2} Q_{\mathbf{A} \cup \Delta_i}(\mathbf{x}) \left[\log Q_{\mathbf{A} \cup \Delta_i}(\mathbf{x}) - \log Q_{\mathbf{A}}(\mathbf{x}) \right] \right\} \tag{6.28}$$

Substituting the representations of $Q_{\mathbf{A}}$ and $Q_{\mathbf{A} \cup \Delta_i}, i = 1, 2$ (given in equations (6.25), (6.26) and (6.27) respectively) into equation (6.28) yields, following some algebra, the following expression:

$$\sum_{\mathbf{x} \in \mathcal{X}^N} \Big\{ Q_{124}(\mathbf{x}) \log Q_{124}(\mathbf{x}) + Q_{134}(\mathbf{x}) \log Q_{134}(\mathbf{x}) + Q_{346}(\mathbf{x}) \log Q_{346}(\mathbf{x})$$

$$+ Q_{356}(\mathbf{x}) \log Q_{356}(\mathbf{x}) - Q_{34}(\mathbf{x}) \log Q_{14}(\mathbf{x}) - Q_{34}(\mathbf{x}) \log Q_{36}(\mathbf{x}) - Q_{36}(\mathbf{x}) \log Q_{14}(\mathbf{x}) \Big\}$$

By computing the negative entropy $-H(q)$ (where q is defined in equation (6.24)), we see that this is an exact representation of the (negative) entropy of q. Therefore, we have established in a direct and constructive manner that the cost function $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ is exact in the sense of Definition 6.3.1 for the graph of Figure 6.6(a).

It is also interesting to consider the Kikuchi approximation obtained by the 4-plaque clustering $\{1,2,3,4\}$ and $\{3,4,5,6\}$, in analogy to Example 6.3.3. To be precise, the core structure consists of the set of nodes $\{1,2,(34),5,6\}$, where (34) denotes the node formed by clustering 3 and 4 together, as illustrated in Figure 6.7(a). The residual set associated with $\{1,2,3,4\}$ is the set of edges $\Delta_1 = \{(1,2),(1,34),(2,34)\}$ as shown in panel (b), whereas the residual set for $\{3,4,5,6\}$ is formed of $\Delta_2 = \{(5,6),(5,34),(6,34)\}$, as shown in panel (c). Here the notation (s,tu) denotes an edge between node s and the

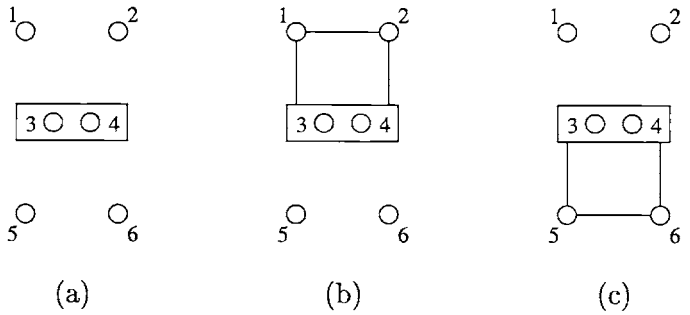


Figure 6.7. Kikuchi approximation on 2-square graph. (a) Fully disconnected core with clustered nodes. (b) Augmented set associated with 4-plaque $\{1,2,3,4\}$. (c) Augmented set associated with 4-plaque $\{3,4,5,6\}$.

clustered node (tu). It can be seen that these residual terms cover all maximal cliques of a triangulated version of the graph in a disjoint manner, so that Proposition 6.3.1 ensures that the Kikuchi approximation is also exact.

Example 6.3.5 (Non-exactness with a non-disjoint partition).
To follow up the previous example, we now illustrate non-exactness, established via Proposition 6.3.1, in the context of Kikuchi 4-plaque clustering applied to the 3×3 grid of Example 6.3.3. In this case, the union of the core structure and the edges Δ associated with the 4-plaques does *not* cover all the cliques of a triangulated version. In particular, there are two 4-cliques in the center of any triangulated version of the 3×3 grid (see Figure 6.1(b)). Neither of these 4-cliques are covered by any of the 4-plaques in this Kikuchi approximation. Therefore, the function $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ is, in general, only an approximation to the KL divergence.

Example 6.3.6 (Disjointness at augmented level). To demonstrate the necessity of defining the disjointness of a residual partition as in Definition 6.3.2, we continue with Example 6.3.1. That is, consider again the 2-square graph, which we have illustrated in Figure 6.8(a). As in Example 6.3.1, we choose as the core set the spanning tree shown

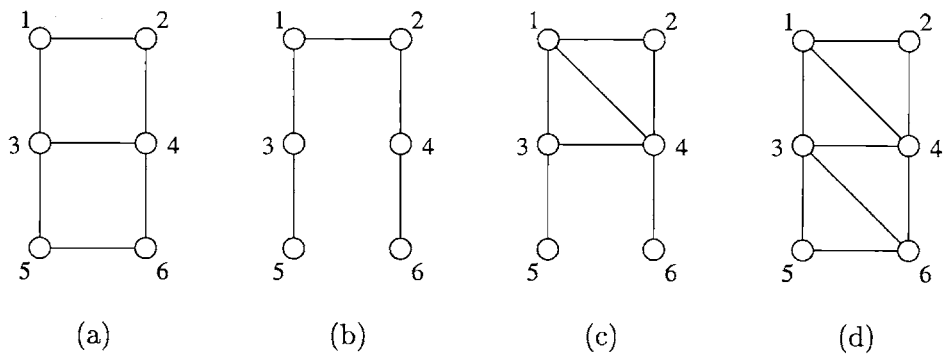


Figure 6.8. Necessity of defining disjointness at the level of the augmented residual sets. (a) Alternative spanning tree of $\mathcal{G}$ from Figure 6.6(a). (b) and (c): Auxiliary structures formed by adding the residual sets $\Delta_1 = (3,4)$ and $\Delta_2 = (5,6)$ respectively. Extra edges are added to form the corresponding augmented residual sets $\widetilde{\Delta}_1 = \{\Delta_1, (1,4)\}$ and $\widetilde{\Delta}_2 = \{\Delta_2, (1,4), (3,4), (3,6)\}$ to ensure triangulation. The partition is disjoint at the level of the Δ_i, but not at the level of the augmented $\widetilde{\Delta}_i$.

in Figure 6.8(b), with the corresponding residual partitions $\Delta_1 = (3,4)$ and $\Delta_2 = (5,6)$. This gives rise to the auxiliary structures shown in Figure 6.8(c) and (d) respectively. As discussed in Example 6.3.1, the partition $\{\Delta_1, \Delta_2\}$ is disjoint, whereas the augmented partition formed by $\widetilde{\Delta}_1 = \{(3,4), (1,4)\}$ and $\widetilde{\Delta}_2 = \{(5,6), (1,4), (3,4), (3,6)\}$ is no longer disjoint.

It can be seen that the union $\mathbf{A} \cup \widetilde{\mathbf{R}}$ covers the clique set $\widetilde{\mathbf{C}}$ of a triangulated version $\widetilde{\mathcal{G}}$. (In particular, use the triangulated version shown in Figure 6.8(c)). If disjointness were defined in terms of Δ_1 and Δ_2 (as opposed to Definition 6.3.2), then the residual partition $\mathbf{R} = \{\Delta_1, \Delta_2\}$ would be pairwise disjoint, and we would be led to use a cost function of the form in equation (6.15). Calculations similar to those in Example 6.3.4 show that the approximation is not exact with the cost function of equation (6.15). Therefore, applying Proposition 6.3.1 would suggest (misleadingly) that the resultant approximation is exact.

The apparent contradiction is resolved by noting that at the level of the augmented residual sets, the partition $\widetilde{\mathbf{R}} = \{\widetilde{\Delta}_1, \widetilde{\Delta}_2\}$ is not pairwise disjoint. Therefore, we should use the cost function of equation (6.18), which is applicable for non-disjoint partitions. It can be seen that in accordance with Proposition 6.3.1, this cost function is indeed exact in the sense of Definition 6.3.1.

Example 6.3.7 (Non-exactness with a disjoint partition).
We now consider a non-exact case for a disjoint residual partition; this example involves a very simple graph that nonetheless reveals the factors that control the accuracy of the approximation. In particular, consider the 5-node graph shown in Figure 6.9(a), with the core structure being the spanning tree shown in Figure 6.9(b). We partition the residual set into two edges $\Delta_1 = (1,4)$ and $\Delta_2 = (4,5)$. In order to assure triangulation

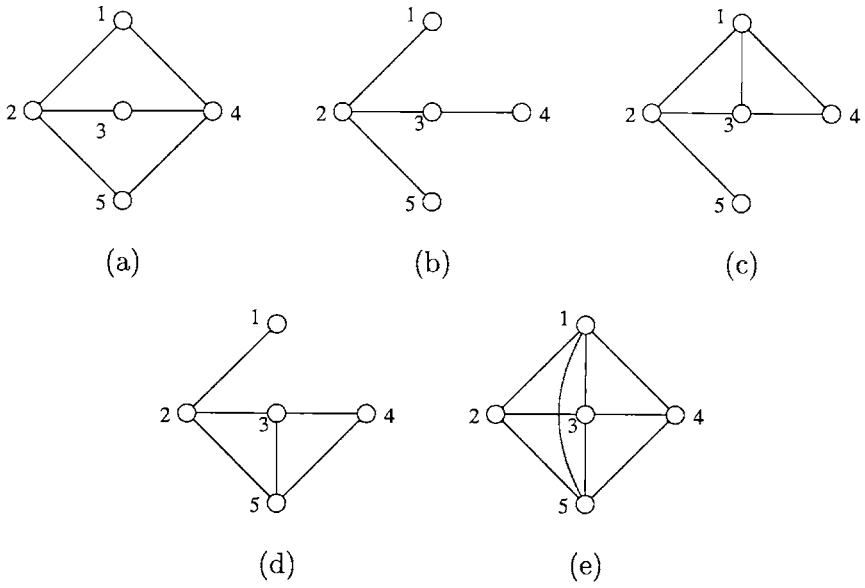


Figure 6.9. Non-exactness of the $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ approximation with a disjoint partition. (a) Original graph on 5 nodes. (b) Spanning tree as core structure. (c) Auxiliary structure formed by $\mathbf{A} \cup (1,4)$. (d) Auxiliary structure formed by $\mathbf{A} \cup (4,5)$. Extra edges (edges $(1,3)$ and $(3,5)$ in panels (c) and (d) respectively) are added so as triangulate these auxiliary structures. (e) Triangulated version of the original graph.

of the induced graphs $\widetilde{\mathcal{G}}(\mathbf{A} \cup \widetilde{\Delta}_i)$, we augment the Δ_i, $i = 1,2$ to $\widetilde{\Delta}_1 = \{\Delta_1, (1,3)\}$ and $\widetilde{\Delta}_2 = \{\Delta_2, (3,5)\}$ respectively. The resulting triangulated graphs $\widetilde{\mathcal{G}}(\mathbf{A} \cup \widetilde{\Delta}_i)$ are shown in Figures (c) and (d). From these triangulated graphs, it can be seen that the auxiliary structure of (c) treats the cliques $\{1,2,3\}$, $\{1,3,4\}$, and $(2,5)$ exactly, whereas (d) treats $\{2,3,5\}$, $\{3,4,5\}$, and $(1,2)$ exactly. The discrepancy with the exact model becomes clear upon considering a triangulated version of the original graph, as shown in (e). Here we see that it is necessary to consider (in addition to the previously listed 2 and 3-cliques) the 4-cliques $\{1,2,3,5\}$ and $\{1,3,4,5\}$, both of which are neglected by the auxiliary structures of (c) and (d). Since $\mathbf{A} \cup \widetilde{\mathbf{R}} \subset \widetilde{\mathbf{C}}$, Proposition 6.3.1 indicates that $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ will not be an exact representation of the KL divergence.

In essence, this $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ function assumes the existence of a distribution $q(\mathbf{x})$ whose higher order marginals satisfy certain factorization properties that may not hold — namely:

$$Q_{1235}(\mathbf{x}) = \frac{Q_{123}(\mathbf{x})Q_{235}(\mathbf{x})}{Q_{23}(\mathbf{x})}$$
$$Q_{1345}(\mathbf{x}) = \frac{Q_{134}(\mathbf{x})Q_{345}(\mathbf{x})}{Q_{34}(\mathbf{x})}$$

Therefore, the cost function $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ will not be exact in the sense of Definition 6.3.1.

6.4 Properties of optimal points

Our analysis in the previous section focused on properties of the cost functions $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ of equations (6.15) and (6.18). For the purposes of approximate inference, our interest lies not so much in properties of these cost functions themselves, but rather in the associated minimizing arguments $\vec{\mathbf{Q}}^*$ of the variational problems (6.16) and (6.20). Of course, our first order of business is to establish the existence of such minima (whether local or global). Having done so, we then turn to an investigation of the properties of such optimal points. Of primary interest here is the relation between such a point $\vec{\mathbf{Q}}^*$, and the true marginals P of the target distribution $p(\mathbf{x})$.

The key result of this section is a form of *invariance* satisfied by any local minimum $\vec{\mathbf{Q}}^*$. This property is a generalization of the invariance satisfied by the TRP/BP updates, as developed in Chapter 5. In later sections, we shall explore the consequences of this invariance, especially its use in characterizing the error between the approximate and exact marginals.

6.4.1 Existence of local minima

We begin by establishing that the variational problems (6.16) and (6.20) always have solutions. It is clear that the set of points that satisfy the marginalization constraints associated with these variational problems is always non-empty; in particular, the marginals corresponding to the target distribution $p(\mathbf{x})$ belong to the constraint set. Therefore, in order to prove the existence of a solution, it suffices to establish that the cost function $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ is bounded below.

Lemma 6.4.1. The cost function $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ is bounded below for all $\vec{\mathbf{Q}}$.

Proof. We assume without loss of generality that $\psi_C(\mathbf{x}_C) \leq 1$ for all $C \in \mathbf{C}$ and $\mathbf{x} \in \mathcal{X}^N$. (This assumption can be satisfied by rescaling the compatibility functions as necessary, which will only affect the normalization constant). We decompose the function $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ into entropy and average energy terms as follows:

$$\mathcal{G}_{\mathbf{A};\mathbf{R}}(\vec{\mathbf{Q}}) = (1 - |\mathbf{R}|)\, H(Q_{\mathbf{A}}) + \sum_{\Delta \in \mathbf{R}} H(Q_{\mathbf{A} \cup \Delta}) - \sum_{\mathbf{x} \in \mathcal{X}^N} \sum_{C \in \mathbf{C}} Q(\mathbf{x}_C) \log \psi_C(\mathbf{x}_C) + K$$

where $|\mathbf{R}|$ denotes the number of terms in the residual partition and K is a fixed finite constant independent of $\vec{\mathbf{Q}}$. For discrete random variables, entropy is bounded both above and below [41]; therefore, the term

$$(1 - |\mathbf{R}|)\, H(Q_{\mathbf{A}}) + \sum_{\Delta \in \mathbf{R}} H(Q_{\mathbf{A} \cup \Delta})$$

is bounded below. Since $-\log \psi_C(\mathbf{x}_C) \geq 0$ and $Q(\mathbf{x}_C) \geq 0$ for all $\mathbf{x} \in \mathcal{X}^N$, the second energy term is also bounded below.

A similar argument establishes that the cost function of problem (6.20) is bounded below. □

■ 6.4.2 Invariance of optimal points

As developed in Chapter 5, an important property of TRP (or the reparameterization form of BP) is that the distribution $p(\mathbf{x})$ on the graph remains invariant under the updates. In this section, we establish a generalized form of this invariance that applies to local minima $\vec{\mathbf{Q}}^*$ of the variational problems (6.16) and (6.20). In particular, we show that the collection of distributions $\vec{\mathbf{Q}}^*$ can be viewed as an alternative parameterization for the target distribution $p(\mathbf{x})$.

Theorem 6.4.1 (Invariance of local minima).
Any local minimum $\vec{\mathbf{Q}}^*$ of variational problem (6.16) is a reparameterization of the target distribution p in the following sense:

$$\log Q^*_{\mathbf{A}}(\mathbf{x}) + \sum_{\Delta \in \mathbf{R}} \left[\log Q^*_{\mathbf{A} \cup \Delta}(\mathbf{x}) - \log Q^*_{\mathbf{A}}(\mathbf{x}) \right] = \log p(\mathbf{x}) + K \tag{6.29}$$

where K is a constant independent of $\mathbf{x}$.

Similarly, any local minimum of variational problem (6.20) is a reparameterization in the following sense:

$$\begin{aligned} \log Q^*_{\mathbf{A}}(\mathbf{x}) + \sum_{\Delta \in \mathbf{R}} & \left[\log Q^*_{\mathbf{A} \cup \Delta}(\mathbf{x}) - \log Q^*_{\mathbf{A}}(\mathbf{x}) \right] \\ & - \sum_{\widetilde{\Delta}_a \cap \widetilde{\Delta}_b \neq \emptyset} \left[\log Q^*_{\mathbf{A} \cup \Delta_a \cap \Delta_b}(\mathbf{x}) - \log Q^*_{\mathbf{A}}(\mathbf{x}) \right] = \log p(\mathbf{x}) + K \end{aligned} \tag{6.30}$$

Proof. We provide a detailed proof of equation (6.29); the proof of equation (6.30) is extremely similar. To each marginalization constraint $\mathbb{M}(Q_{\mathbf{A} \cup \Delta}) = Q_{\mathbf{A}}$ of problem (6.16), we associate a Lagrange multiplier $\boldsymbol{\lambda}_\Delta$. To specify precisely the nature of $\boldsymbol{\lambda}_\Delta$, recall from equations (6.12) that the constraint $\mathbb{M}(Q_{\mathbf{A} \cup \Delta}) = Q_{\mathbf{A}}$ actually indexes a collection of marginalization constraints, one for each pair $(\mathcal{C}, \mathcal{D})$ in the set $\mathfrak{P}(\Delta)$ defined in equation (6.11). As a result, $\boldsymbol{\lambda}_\Delta$ is actually the collection

$$\boldsymbol{\lambda}_\Delta(\mathbf{x}) \triangleq \{ \lambda_\Delta^{\mathcal{C};\mathcal{D}}(\mathbf{x}_\mathcal{C}) \mid (\mathcal{C}, \mathcal{D}) \in \mathfrak{P}(\mathcal{G}) \} \tag{6.31}$$

of Lagrange multiplier functions, where $\lambda_\Delta^{\mathcal{C};\mathcal{D}}(\mathbf{x}_\mathcal{C})$ is associated with the constraint

$$\sum_{\mathbf{x}'_\mathcal{D} \text{ s.t } \mathbf{x}'_\mathcal{D} = \mathbf{x}_\mathcal{C}} \widetilde{Q}_\mathcal{D}(\mathbf{x}'_\mathcal{D}) = Q_\mathcal{C}(\mathbf{x}_\mathcal{C})$$

To simplify our notation, we define:

$$\begin{aligned} \boldsymbol{\lambda}_\Delta(\mathbf{x}) \bullet \left[Q_{\mathbf{A}}(\mathbf{x}) - \mathbb{M}(Q_{\mathbf{A} \cup \Delta}(\mathbf{x})) \right] &\triangleq \sum_{(\mathcal{C};\mathcal{D}) \in \mathfrak{P}(\Delta)} \lambda_\Delta^{\mathcal{C};\mathcal{D}}(\mathbf{x}_\mathcal{C}) \left[Q_\mathcal{C}(\mathbf{x}_\mathcal{C}) - \sum_{x_s;\, s \in \mathcal{D}/\mathcal{C}} \widetilde{Q}_\mathcal{D}(\mathbf{x}_\mathcal{D}) \right] \\ \boldsymbol{\lambda}_\Delta(\mathbf{x}) \bullet \mathbf{1}(\mathbf{x}) &\triangleq \sum_{(\mathcal{C};\mathcal{D}) \in \mathfrak{P}(\Delta)} \lambda_\Delta^{\mathcal{C};\mathcal{D}}(\mathbf{x}_\mathcal{C}) \end{aligned}$$

Moreover, we don't enforce the normalization constraints (e.g., $\sum_{\mathbf{x} \in \mathcal{X}^N} Q_{\mathbf{A}}(\mathbf{x}) = 1$) explicitly; instead, we use an arbitrary constant K (whose definition will change from line to line) to enforce normalization where appropriate.

Given this definition of the Lagrange multipliers $\boldsymbol{\lambda}_\Delta$, we then form the Lagrangian associated with variational problem (6.16):

$$\begin{aligned}
\mathcal{L}(\vec{\mathbf{Q}}; \boldsymbol{\lambda}) &= \mathcal{G}_{\mathbf{A};\mathbf{R}}(\vec{\mathbf{Q}}) + \sum_{\mathbf{x} \in \mathcal{X}^N} \sum_{\Delta \in \mathbf{R}} \boldsymbol{\lambda}_\Delta(\mathbf{x}) \bullet \left[Q_{\mathbf{A}}(\mathbf{x}) - \mathbb{M}(Q_{\mathbf{A} \cup \Delta})(\mathbf{x}) \right] \\
&= (1 - |\mathbf{R}|)\, D(Q_{\mathbf{A}} \parallel P_{\mathbf{A}}) + \sum_{\Delta \in \mathbf{R}} D(Q_{\mathbf{A} \cup \Delta} \parallel P_{\mathbf{A} \cup \Delta}) \\
&\qquad + \sum_{\mathbf{x} \in \mathcal{X}^N} \sum_{\Delta \in \mathbf{R}} \boldsymbol{\lambda}_\Delta(\mathbf{x}) \bullet \left[Q_{\mathbf{A}}(\mathbf{x}) - \mathbb{M}(Q_{\mathbf{A} \cup \Delta})(\mathbf{x}) \right]
\end{aligned}$$

From here, the crucial idea is that for any local minimum $\vec{\mathbf{Q}}$ of problem (6.16), we are guaranteed the existence of an associated collection of Lagrange multipliers $\boldsymbol{\lambda}^* = \{\boldsymbol{\lambda}^*_\Delta\}$ such that the Lagrangian stationary conditions

$$\nabla_{\vec{\mathbf{Q}}} \mathcal{L}(\vec{\mathbf{Q}}^*; \boldsymbol{\lambda}^*) = 0 \tag{6.33}$$

hold. The existence of these Lagrange multipliers follows because the marginalization constraints of problem (6.16) are all linear.[2] As a consequence, we can use the stationary condition of equation (6.33) to characterize any local minimum $\vec{\mathbf{Q}}^*$ of problem (6.16).

By taking derivatives of the Lagrangian with respect to $Q_{\mathbf{A}}$ and $Q_{\mathbf{A} \cup \Delta}$ and setting them to zero, we obtain a set of equations equivalent to equation (6.33):

$$(1 - |\mathbf{R}|)\, \log Q^*_{\mathbf{A}}(\mathbf{x}) = (1 - |\mathbf{R}|)\, \log P_{\mathbf{A}}(\mathbf{x}) - \sum_{\Delta \in \mathbf{R}} \boldsymbol{\lambda}^*_\Delta(\mathbf{x}) \bullet \mathbf{1}(\mathbf{x}) + K_{\text{core}} \tag{6.34a}$$

$$\log Q^*_{\mathbf{A} \cup \Delta}(\mathbf{x}) = \log P_{\mathbf{A} \cup \Delta}(\mathbf{x}) + \boldsymbol{\lambda}^*_\Delta(\mathbf{x}) \bullet \mathbf{1}(\mathbf{x}) + K_\Delta \tag{6.34b}$$

where K_{core} and K_Δ represent constants to ensure proper normalization of $Q_{\mathbf{A}}$ and $Q_{\mathbf{A} \cup \Delta}$ respectively.

We now add equation (6.34a) to $|\mathbf{R}|$ copies (one for each $\Delta \in \mathbf{R}$) of equation (6.20) to obtain:

$$(1 - |\mathbf{R}|)\, \log Q^*_{\mathbf{A}}(\mathbf{x}) + \sum_{\Delta \in \mathbf{R}} \log Q^*_{\mathbf{A} \cup \Delta}(\mathbf{x}) = (1 - |\mathbf{R}|)\, \log P_{\mathbf{A}}(\mathbf{x}) + \sum_{\Delta \in \mathbf{R}} \log P_{\mathbf{A} \cup \Delta}(\mathbf{x}) + K \tag{6.35}$$

where $K = K_{\text{core}} + \sum_{\Delta \in \mathbf{R}} K_\Delta$. Note how the Lagrange multipliers $\boldsymbol{\lambda}^*_\Delta$ themselves have cancelled out.

Finally, we observe that by construction, the RHS of equation (6.35) is closely related to the log of the target distribution $p(\mathbf{x}) \propto \prod_{C \in \mathbf{C}} \psi_C(\mathbf{x}_C)$. In particular, by an

[2] Local minima of constrained optimization problems with non-linear constraints don't necessarily have Lagrange multipliers; see Bertsekas [20] for a counterexample.

argument similar to the proof of Lemma 6.3.1, we write (omitting explicit dependence on $\mathbf{x}$ for notational simplicity):

$$\begin{aligned}(1-|\mathbf{R}|)\,\log P_{\mathbf{A}} + \sum_{\Delta\in\mathbf{R}} \log P_{\mathbf{A}\cup\Delta} + K &= (1-|\mathbf{R}|)\sum_{C\in\mathbf{A}} \log\psi_C + \sum_{\Delta\in\mathbf{R}}\sum_{C\in\mathbf{A}\cup\Delta} \log\psi_C + K \\ &= \sum_{C\in\mathbf{A}} \log\psi_C + \sum_{\Delta\in\mathbf{R}}\sum_{C\in\Delta} \log\psi_C + K \\ &= \sum_{C\in\mathbf{C}} \log\psi_C + K \\ &= \log p + K\end{aligned}$$

where K is an arbitrary constant that absorbs terms not dependent on $\mathbf{x}$ (i.e., its definition can change from line to line). This establishes equation (6.29). A similar argument can be used to prove equation (6.30).

□

■ 6.4.3 Generalized message-passing for minimization

Theorem 6.4.1 is a characterization of local minima that is independent of the technique used to find them. It is nonetheless interesting to consider iterative schemes that are generalizations of BP message-passing. In this section, we describe such an algorithm for problem (6.16); a similar algorithm can be developed for problem (6.20). This algorithm has the interesting property that all of its iterates (not just its fixed points) satisfy the invariance principle of Theorem 6.4.1.

The essential intuition lies in the Lagrangian conditions that were exploited to prove Theorem 6.4.1. In order to develop this intuition, we recall the notation used in the proof of Theorem 6.4.1:

$$\boldsymbol{\lambda}_\Delta(\mathbf{x}) \triangleq \{\, \lambda_\Delta^{C;\mathcal{D}}(\mathbf{x}_C) \mid (C,\mathcal{D}) \in \mathfrak{P}(\mathcal{G}) \,\} \tag{6.36a}$$

$$\boldsymbol{\lambda}_\Delta(\mathbf{x}) \bullet \mathbf{1}(\mathbf{x}) \triangleq \sum_{(C;\mathcal{D})\in\mathfrak{P}(\Delta)} \lambda_\Delta^{C;\mathcal{D}}(\mathbf{x}_C) \tag{6.36b}$$

where the set of clique pairs $\mathfrak{P}(\mathcal{G})$ was defined earlier in equation (6.11). Note that each $\lambda_\Delta^{C;\mathcal{D}}(\mathbf{x}_C)$ depends only on the subvector $\mathbf{x}_C$. Therefore, $\boldsymbol{\lambda}_\Delta(\mathbf{x}) \bullet \mathbf{1}(\mathbf{x})$ is defined in terms of an additive decomposition of relatively local functions. In fact, since for any pair $(C;\mathcal{D}) \in \mathfrak{P}(\Delta)$, the clique C is a member $\mathbf{C}(\mathbf{A})$, the function $\boldsymbol{\lambda}_\Delta(\mathbf{x}) \bullet \mathbf{1}(\mathbf{x})$ respects the structure of the core set $\mathbf{A}$. This local nature of $\boldsymbol{\lambda}_\Delta$ is crucial.

With this notation, the Lagrangian stationary conditions of equations (6.34a) and (6.34b) dictate that the approximating distributions $\{Q_{\mathbf{A}}, Q_{\mathbf{A}\cup\Delta}\}$ should be defined in terms of the Lagrange multipliers $\{\boldsymbol{\lambda}_\Delta\}$ as follows:

$$Q_{\mathbf{A}}(\mathbf{x}) = \kappa\; P_{\mathbf{A}}(\mathbf{x})\; \exp\Big\{\frac{1}{|\mathbf{R}|-1}\sum_{\Delta\in\mathbf{R}} \boldsymbol{\lambda}_\Delta(\mathbf{x}) \bullet \mathbf{1}(\mathbf{x})\Big\} \tag{6.37a}$$

$$Q_{\mathbf{A}\cup\Delta}(\mathbf{x}) = \kappa\; P_{\mathbf{A}\cup\Delta}(\mathbf{x})\; \exp\big\{\boldsymbol{\lambda}_\Delta(\mathbf{x}) \bullet \mathbf{1}(\mathbf{x})\big\} \tag{6.37b}$$

where κ denotes a normalization factor. The key is the marginalization constraint $\mathbb{M}(Q_{\mathbf{A}\cup\Delta}(\mathbf{x})) = Q_{\mathbf{A}}(\mathbf{x})$, which places restrictions on the Lagrange multipliers $\{\boldsymbol{\lambda}_\Delta\}$.

So as to define an algorithm that reduces to belief propagation, we perform a linear transformation of the Lagrange multipliers, thereby defining a set of (log) messages. In particular, the set of messages $\{\ \mathbf{M}_\Delta \mid \Delta \in \mathbf{R}\ \}$ is linked to the $\{\boldsymbol{\lambda}_\Delta\}$ via the invertible relations:

$$\log \mathbf{M}_\Delta(\mathbf{x}) = \frac{1}{|\mathbf{R}|-1}\sum_{\Upsilon\in\mathbf{R}/\Delta}\boldsymbol{\lambda}_\Upsilon(\mathbf{x}) - \left[\frac{|\mathbf{R}|-2}{|\mathbf{R}|-1}\right]\boldsymbol{\lambda}_\Delta(\mathbf{x}) \tag{6.38a}$$

$$\boldsymbol{\lambda}_\Delta(\mathbf{x}) = \sum_{\Upsilon\in\mathbf{R}/\Delta}\log \mathbf{M}_\Upsilon(\mathbf{x}) \tag{6.38b}$$

Each $\log \mathbf{M}_\Delta$ has the same structure as $\boldsymbol{\lambda}_\Delta$; that is, it decomposes into a sum of local functions on the cliques of $\mathbf{A}$.

As a multiplicative analog to $\boldsymbol{\lambda}_\Delta \bullet \mathbf{1}(\mathbf{x})$, define the product notation:

$$\mathbf{M}_\Delta(\mathbf{x}) \otimes \mathbf{1}(\mathbf{x}) \triangleq \prod_{(\mathcal{C};\mathcal{D})\in\mathfrak{P}(\Delta)} M_\Delta^{\mathcal{C};\mathcal{D}}(\mathbf{x}) \tag{6.39}$$

Then equations (6.37a) and (6.37b) can be re-written in terms of these messages as follows:

$$Q_{\mathbf{A}}(\mathbf{x}) = \kappa\ P_{\mathbf{A}}(\mathbf{x}) \prod_{\Delta\in\mathbf{R}} M_\Delta(\mathbf{x}) \otimes \mathbf{1}(\mathbf{x}) \tag{6.40a}$$

$$Q_{\mathbf{A}\cup\Delta}(\mathbf{x}) = \kappa\ P_{\mathbf{A}\cup\Delta}(\mathbf{x}) \prod_{\Upsilon\in\mathbf{R}/\Delta} M_\Upsilon(\mathbf{x}) \otimes \mathbf{1}(\mathbf{x}) \tag{6.40b}$$

where κ denotes a normalization factor (whose definition may be different from line to line).

We now need to update the messages so that the associated marginalization constraints $\mathbb{M}(Q_{\mathbf{A}\cup\Delta}) = Q_{\mathbf{A}}$ are satisfied; we do so with Algorithm 6.4.1. Although normalizing the messages in equation (6.41) is not strictly necessary, it tends to aid computational stability.

As with the reparameterization algorithms of Chapter 5, we can dispense entirely with the messages by reformulating the updates in a pure reparameterization form as follows:

$$Q_{\mathbf{A}}^{n+1}(\mathbf{x}) = \kappa\ P_{\mathbf{A}}(\mathbf{x}) \prod_{\Delta\in\mathbf{R}} \frac{\mathbb{M}\left[Q_{\mathbf{A}\cup\Delta}^n(\mathbf{x})\right]}{Q_{\mathbf{A}}^n(\mathbf{x})} \tag{6.42a}$$

$$Q_{\mathbf{A}\cup\Delta}^{n+1}(\mathbf{x}) = \kappa\ P_{\mathbf{A}\cup\Delta}(\mathbf{x}) \prod_{\Upsilon\in\mathbf{R}/\Delta} \frac{\mathbb{M}\left[Q_{\mathbf{A}\cup\Upsilon}^n(\mathbf{x})\right]}{Q_{\mathbf{A}}^n(\mathbf{x})} \tag{6.42b}$$

Moreover, it is clear by the construction of this generalized message-passing scheme that its fixed points satisfy the Lagrangian conditions associated with problem (6.16).

Algorithm 6.4.1 (Generalized message-passing: (GMP)).

1. Initialize messages M_{Δ}^{0} for all $\Delta \in \mathbf{R}$.
2. At iteration $n = 0, 1, \ldots,$ core and auxiliary distributions $Q_{\mathbf{A}}$ and $Q_{\mathbf{A} \cup \Delta}$ are specified in terms of the messages $\{\mathbf{M}_{\Delta}^{n}\}$ as in equations (6.40a) and (6.40b).
3. Update messages:

$$\mathbf{M}_{\Delta}^{n+1}(\mathbf{x}) = \kappa \; \mathbf{M}_{\Delta}^{n}(\mathbf{x}) \; \frac{\mathbb{M}\big[Q_{\mathbf{A} \cup \Delta}^{n}(\mathbf{x})\big]}{Q_{\mathbf{A}}^{n}(\mathbf{x})} \tag{6.41}$$

where κ denotes a normalization factor.

Therefore, its fixed points obey the invariance principle of Theorem 6.4.1. However, a much stronger statement can be made. By applying the same argument used in the proof of Theorem 6.4.1 to the representations of $Q_{\mathbf{A}}^{n}$ and $Q_{\mathbf{A} \cup \Delta}^{n}$ in equations (6.42a) and (6.42b), it can be seen that *all the iterates* of generalized message-passing — not just fixed points — satisfy the invariance principle of Theorem 6.4.1.

To gain intuition for this generalized message-passing, we now consider a few examples. We begin with the special case where the core structure is $\mathbf{A} = \mathcal{V}$, and the residual structure $\mathbf{R}$ is partitioned into individual edges $\Delta = (s, t)$. We show that in this case the message-passing of the GMP algorithm, modulo some minor notational differences, corresponds to belief propagation.

Example 6.4.1 (Belief propagation).
When $\mathbf{A} = \mathcal{V}$, then the CTD $P_{\mathbf{A}}(\mathbf{x})$ takes the form

$$P_{\mathbf{A}}(\mathbf{x}) \propto \prod_{s \in \mathcal{V}} \psi_s(x_s) \tag{6.43}$$

Similarly, since the cliques of $\mathbf{A}$ are simply vertices, each message $\mathbf{M}_{\Delta} \equiv \mathbf{M}_{uv}$ in the generalized message-passing of the GMP algorithm is a fully factorized quantity, which we write as follows:

$$\mathbf{M}_{uv}(\mathbf{x}) = \prod_{s \in \mathcal{V}} M_{uv;s}(x_s) \tag{6.44}$$

I.e., it has a term corresponding to each node $s \in \mathcal{V}$.

From the definitions of $Q_{\mathbf{A}}$ and $Q_{\mathbf{A} \cup \Delta}$ in equations (6.37a) and (6.37b) respectively, as well as the message update equation (6.41), it can be seen that when $\Delta = \{(u, v)\}$, then the message components $M_{uv;s}$ are constant for all nodes $s \neq u, v$. Thus, each $\mathbf{M}_{uv}(\mathbf{x})$ is actually a function of only x_u and x_v. As a consequence, the only actual messages sent to a node s are from its neighbors $\mathcal{N}(s) \triangleq \{\, t \in \mathcal{V} \mid (s, t) \in \mathcal{E} \,\}$.

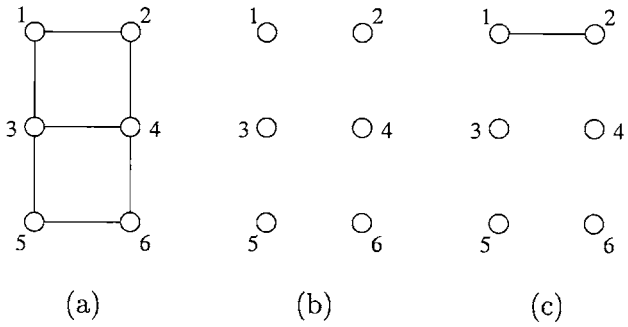


Figure 6.10. Illustration of structures involved in BP. (a) Original graph. (b) Fully disconnected core structure $\mathbf{A}$. (c) Augmented structure formed by a single edge (in this case edge $(1,2)$).

Figure 6.10 illustrates this type of decomposition; panel (a) shows the original graph $\mathcal{G}$. The CAD $Q_{\mathbf{A}}(\mathbf{x})$ is defined on the fully disconnected core structure shown in (b). The message $\mathbf{M}_{(1,2)}(\mathbf{x})$ is associated with the augmented structure obtained by adding edge $(1,2)$ to the core structure, as illustrated in panel (c). It is clear that adding edge $(1,2)$ will not affect the marginals at nodes $s \neq 1,2$.

Based on these properties, we can rewrite $Q_{\mathbf{A}}$, as defined in equation (6.42a), in the following way:

$$Q_{\mathbf{A}}(\mathbf{x}) = \kappa \prod_{s \in \mathcal{V}} \Big\{ \psi_s(x_s) \prod_{t \in \mathcal{N}(s)} M_{st;s}(x_s) \Big\} \tag{6.45}$$

Note that the term within curly braces in equation (6.45), modulo slightly altered notation, is precisely equivalent to the usual BP equation for the approximate single node marginal at node s (see equation (5.4) of Chapter 5).

Similarly, the distribution $P_{\mathbf{A} \cup (u,v)}$ has the form:

$$P_{\mathbf{A} \cup (u,v)}(\mathbf{x}); \propto \prod_{s \in \mathcal{V}} \psi_s(x_s)\, \psi_{uv}(x_u, x_v) \tag{6.46}$$

Using this form of $P_{\mathbf{A} \cup (u,v)}$ and the definition of $Q_{\mathbf{A} \cup (u,v)}$ from equation (6.37b), it can be shown that the auxiliary distribution $Q_{\mathbf{A} \cup (u,v)}(\mathbf{x})$ has the following structure:

$$Q_{\mathbf{A} \cup (u,v)}(\mathbf{x}) = \kappa \prod_{s \in \mathcal{V}} \psi_s(x_s)\, \psi_{uv}(x_u, x_v) \prod_{s \in \mathcal{N}(u)/s} M_{su;u}(x_u) \prod_{s \in \mathcal{N}(v)/s} M_{sv;v}(x_v) \tag{6.47}$$

If we isolate the components of equation (6.47) depending only on x_u and x_v, then we obtain

$$Q_{\mathbf{A} \cup (u,v)}(x_u, x_v) = \kappa\ \psi_u(x_u)\, \psi_v(x_v)\, \psi_{uv}(x_u, x_v) \prod_{s \in \mathcal{N}(u)/s} M_{su;u}(x_u) \prod_{s \in \mathcal{N}(v)/s} M_{sv;v}(x_v) \tag{6.48}$$

Again, equation (6.48), modulo minor differences in notation, is equivalent to the BP equations for the joint pairwise marginal (see, e.g., equation (5.21b) of Chapter 5).

Overall, we conclude that in the special case where $\mathbf{A} = \mathcal{V}$ and the residual set is partitioned into single edges, the generalized message-passing of the GMP algorithm is equivalent to belief propagation.

Example 6.4.2 (Flow of messages).
To illustrate the flow of messages in GMP, we now consider an example on a more complex core structure. In particular, we return to the 5-node graph shown of Example 6.3.7; we have redrawn it in Figure 6.11(a). As a core structure, we again use the spanning tree shown in Figure 6.11(b), and we choose the same residual sets $\Delta_1 = (1,3)$ and $\Delta_2 = (3,5)$, which gives rise to the (augmented) structures shown in panels (c) and (d) respectively.

If GMP is applied to this example, there are only two sets of messages — namely, $\{\mathbf{M}_{\Delta_i},\ i = 1,2\}$. Computing the CAD $Q_{\mathbf{A}}(\mathbf{x})$ requires knowledge of the CTD $P_{\mathbf{A}}(\mathbf{x})$, as well as both messages:

$$Q_{\mathbf{A}}(\mathbf{x}) = \kappa\ P_{\mathbf{A}}(\mathbf{x})\mathbf{M}_{\Delta_1}(\mathbf{x})\mathbf{M}_{\Delta_2}(\mathbf{x})$$

Accordingly, we can think of the messages $\{\mathbf{M}_{\Delta_i},\ i = 1,2\}$ being passed from the augmented structures $\{\mathbf{A} \cup \Delta_i,\ i = 1,2\}$ to the core structure, as illustrated by the diagonally upward arrows in Figure 6.11(e). Since the core structure is a tree, the messages are tree-structured quantities.

To compute the auxiliary distribution $Q_{\mathbf{A} \cup \Delta_1}$ requires knowledge of $P_{\mathbf{A} \cup \Delta_1}$, and also the message $\mathbf{M}_{\Delta_2}$. Accordingly, we think of the message $\mathbf{M}_{\Delta_2}$ as being passed from the structure $\mathbf{A} \cup \Delta_2$ to $\mathbf{A} \cup \Delta_1$, following the right-to-left arrow in Figure 6.11(e). Similarly, the message $\mathbf{M}_{\Delta_1}$ is passed from left to right — that is, from $\mathbf{A} \cup \Delta_1$ to $\mathbf{A} \cup \Delta_2$.

■ 6.4.4 Largest globally consistent substructure

Any optimal point $\vec{\mathbf{Q}}^*$ of problem (6.16) consists of a collection of distributions: the core approximating distribution $\{Q^*_{\mathbf{A}}\}$ is defined by the core structure $\widetilde{\mathcal{G}}(\mathbf{A})$, whereas the auxiliary distributions $\{Q^*_{\mathbf{A} \cup \Delta} \mid \Delta \in \mathbf{R}\}$ are defined on the augmented structures $\{\widetilde{\mathcal{G}}(\mathbf{A} \cup \widetilde{\Delta}) \mid \widetilde{\Delta} \in \widetilde{\mathbf{R}}\}$. Recall that the CAD $Q^*_{\mathbf{A}}$ is defined by a product of local marginals over the maximal cliques and separator sets of a junction tree corresponding to the triangulated $\widetilde{\mathcal{G}}(\mathbf{A})$:

$$Q^*_{\mathbf{A}}(\mathbf{x}) = \frac{\prod_{C \in \mathbf{C}_{\max}(\mathbf{A})} Q^*(\mathbf{x}_C)}{\prod_{C \in \mathbf{C}_{\text{sep}}(\mathbf{A})} Q^*(\mathbf{x}_C)} \tag{6.49}$$

The auxiliary distributions $Q^*_{\mathbf{A} \cup \Delta}$ are defined by similar products of local marginals, as in equation (6.10b).

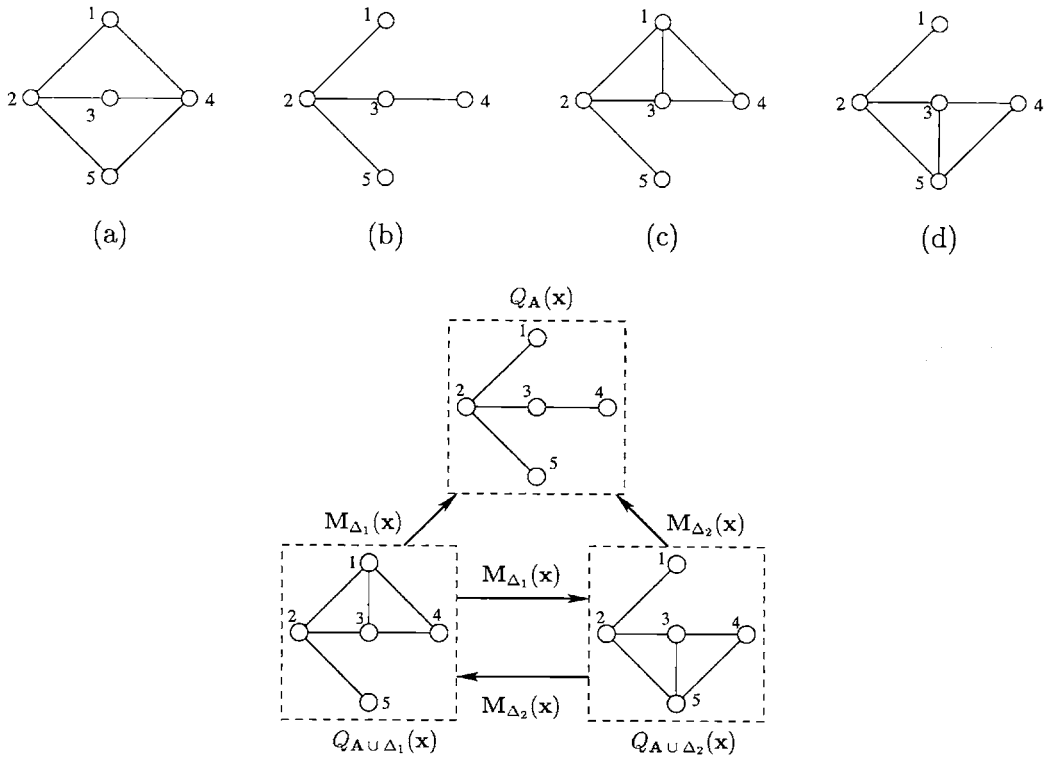


Figure 6.11. Illustration of message-passing flow for a simple 5-node graph. (a) Original graph. (b) Spanning tree core structure. (c) Augmented graph $\widetilde{\mathcal{G}}(\mathbf{A} \cup \widetilde{\Delta}_1)$. (d) Augmented graph $\widetilde{\mathcal{G}}(\mathbf{A} \cup \widetilde{\Delta}_2)$. (e) Flow of messages in GMP.

With respect to the single-node marginals Q_s^* of $Q_{\mathbf{A}}^*$, each of the auxiliary distributions are *globally consistent*, in the sense that they marginalize down appropriately:

$$\sum_{\mathbf{x}' \text{ s.t } x'_s = x_s} Q^*_{\mathbf{A}\cup\Delta}(\mathbf{x}') \;=\; Q^*(x_s) \tag{6.50}$$

These marginalization conditions are assured by the constraints associated with problem (6.16).

A natural question to ask is the following: what is the largest structure over which solutions $\vec{\mathbf{Q}}^*$ are globally consistent? We call this the largest globally consistent structure. It should be understood that we require *any* solution $\vec{\mathbf{Q}}^*$ to be consistent over this structure. This requirement excludes the possibility of artfully constructing problems (as we did in Section 5.4.6 for TRP/BP), such that fortuitous cancellations lead to consistency. That is, the global consistency of interest in this section is *generic*, in the

sense that it holds for all problem instances.

It is important to note that the largest globally consistent structure can be considerably larger than any of the augmented sets $\mathbf{A} \cup \Delta$, as demonstrated by the following example:

Example 6.4.3 (Spanning trees for belief propagation).
Recall the Bethe free energy and belief propagation (BP) as discussed in Examples 6.3.2 and 6.4.1. We found that the BP algorithm can be viewed as updating an approximat-

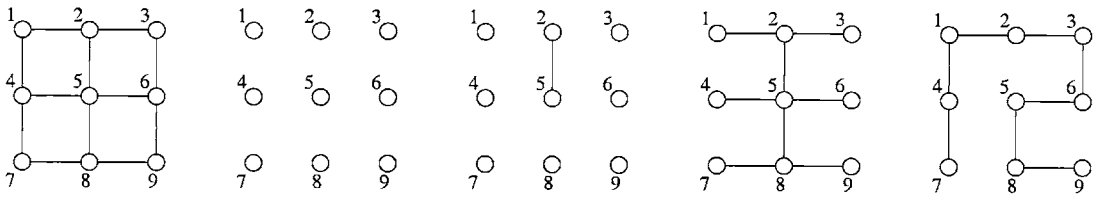


Figure 6.12. Spanning trees are the largest globally consistent substructure for BP. (a) Original 3×3 grid. (b) Fully disconnected graph is the core of BP. (c) Augmented structure $\mathbf{A} \cup \Delta_i$ formed by adding the single edge $(2,5)$. (d), (e) BP solution is guaranteed to be globally consistent over any embedded spanning tree.

ing distribution $Q_{\mathbf{A}}$ defined on a fully disconnected graph. The CAD is completely factorized $Q_{\mathbf{A}}(\mathbf{x}) = \prod_{s \in \mathcal{V}} Q_s(x_s)$. Residual terms for the usual form of BP correspond to single edges. Therefore, in addition to the single-node marginals of $Q_{\mathbf{A}}$, the collection of auxiliary distributions $Q_{\mathbf{A} \cup \Delta}$ gives rise to set of local pseudomarginals $Q_{st}(x_s, x_t)$, one for each edge $(s,t) \in \mathcal{E}$. As an illustrative example, Figure 6.12(a) shows a 3×3 grid, and panel (b) shows the corresponding fully disconnected core structure. The augmented structure $\mathbf{A} \cup \Delta$, corresponding to the addition of edge $(2,5)$, is shown in panel (c).

Upon convergence to a fixed point $\vec{\mathbf{Q}}^*$, each of the pseudomarginals Q^*_{st} for any $(s,t) \in \mathcal{E}$ is guaranteed to be consistent with single node marginals Q^*_s in the following sense:

$$\sum_{x'_t} Q^*_{st}(x_s, x'_t) = Q^*_s(x_s) \tag{6.51}$$

This is equivalent to the condition $\sum_{\mathbf{x}' \text{ s.t } x'_s = x_s} Q^*_{\mathbf{A} \cup (s,t)}(\mathbf{x}') = Q^*_s(x_s)$, so that the auxiliary distributions $Q^*_{\mathbf{A} \cup \Delta}$ are globally consistent with respect to the single-node marginals.

However, it can be seen that global consistency holds for much larger substructures. In particular, given the edge set $\mathcal{E}(\mathcal{T})$ of any tree embedded within $\mathcal{G}$, we construct a

distribution that respects its structure in the usual junction tree manner:

$$q_{\mathcal{T}}^*(\mathbf{x}) \triangleq \prod_{s \in \mathcal{V}} Q_s^*(x_s) \prod_{(s,t) \in \mathcal{E}(\mathcal{T})} \frac{Q_{st}^*(x_s, x_t)}{Q_s^*(x_s) Q_t^*(x_t)} \tag{6.52}$$

For any tree structure, local consistency in the sense of equation (6.51) is equivalent to global consistency [122] — viz.:

$$\sum_{\mathbf{x}' \text{ s.t } x'_s = x_s} q_{\mathcal{T}}^*(\mathbf{x}') = Q_s^*(x_s)$$

for all nodes $s \in \mathcal{V}$. This argument holds for *any* tree (or forest) embedded within the graph $\mathcal{G}$. Therefore, the largest globally consistent substructures are spanning trees; two particular spanning trees of the 3×3 grid in Figure 6.12(a) are illustrated in panels (d) and (e).

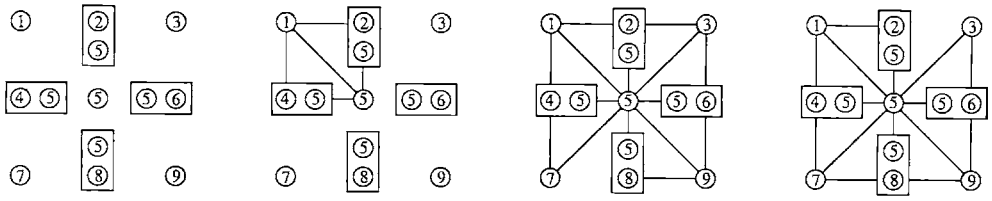
Although the BP solution $\vec{\mathbf{Q}}^*$ is globally consistent over any spanning tree, it will not (in general) be consistent over the full graph. In fact, as pointed out in Section 6.1.1, it is possible that there exists no distribution that, when marginalized, gives rise to the $\{Q_{st}^*\}$ on the full graph. I.e., the full set of $\{Q_{st}^*\}$ may fail to be globally consistent with *any* distribution whatsoever. See [57] for a simple example where this degeneracy arises. The spanning tree condition guarantees only that a subset of the $\{Q_{st}^*\}$ — namely, those corresponding to edges in a spanning tree — are globally consistent. It is for this reason that the terminology pseudomarginals is appropriate.

For the case of belief propagation, the set $\mathbf{A} \cup \mathbf{R}$ (which is equivalent to $\mathbf{A} \cup \widetilde{\mathbf{R}}$) is given by the union $\mathcal{V} \cup \mathcal{E} \subset \widetilde{\mathbf{C}}$. Spanning trees correspond to the largest triangulated substructure that can be formed with this particular subset of cliques (i.e., with vertices and edges). This property is actually quite general, as summarized by the following result.

Proposition 6.4.1 (Largest globally consistent substructure). Given an approximation specified by a core set $\mathbf{A}$ and (augmented) residual set $\widetilde{\mathbf{R}}$, the largest globally consistent substructures are given by the largest triangulated subgraphs that can be formed by joining together cliques (not necessarily maximal) from the set $\mathbf{A} \cup \widetilde{\mathbf{R}}$.

Proof. Given a triangulated subgraph formed of cliques in $\mathbf{A} \cup \widetilde{\mathbf{R}}$, we can always form a distribution on it by taking a combination of local marginals over the maximal cliques and separator sets, as specified by the junction tree representation. (See Section 2.1.5 for more details on the junction tree representation). The marginalization constraints associated with variational problems (6.16) and (6.20) assure that each of these marginal distributions are locally consistent. Local consistency implies global consistency for a triangulated graph [121][3], so that the distribution on the triangulated subgraph is

[3]Indeed, this is the essence of the junction tree representation: it specifies the necessary degree of local consistency that assures global consistency.



(a) (b) (c) (d)

Figure 6.13. Embedded 3-clique trees are the largest globally consistent substructures for this Kikuchi approximation. (a) Fully factorized core formed of clustered nodes. (b) Edges associated with augmented residual term $\widetilde{\Delta}$ for a particular 4-plaque $\{1, 2, 4, 5\}$. Largest globally consistent substructures are 3-clique trees (i.e., graphs of treewidth 2). Particular examples of such graphs are shown in (c) and (d).

globally consistent. If the subgraph is not triangulated, it will not be possible in a generic sense to form a locally consistent distribution that remains globally consistent. □

To illustrate Proposition 6.4.1, we now consider the largest consistent substructures for a particular Kikuchi approximation.

Example 6.4.4 (Largest consistent substructures for Kikuchi).
Recall the 4-plaque Kikuchi approximation applied to the 3×3 grid, as discussed in Example 6.3.3. This clustering procedure gives rise to the fully factorized core structure of clustered nodes, as illustrated in Figure 6.13(a). The augmented structures $\mathbf{A} \cup \widetilde{\Delta}$ correspond to the set of edges associated with a given 4-plaque, plus additional edges to triangulate. For the particular case of adding the 4-plaque $\{1, 2, 4, 5\}$, the corresponding augmented structure is illustrated in panel (b).

The analog of a spanning tree for this Kikuchi approximation is a clique tree formed by 3-cliques. For the 3×3 grid, two such 3-clique trees are shown in panels (c) and (d). Alternatively, these graphs can be said to have treewidth 2; an ordinary tree is formed of edges (i.e., 2-cliques), and has treewidth 1. See [17, 162] for more details on hypergraphs and the notion of treewidth.

Again, let $\vec{\mathbf{Q}}$ be a fixed point of the variational problem associated with this Kikuchi approximation. Now consider a distribution formed by taking products of the local marginals in $\vec{\mathbf{Q}}$ over maximal cliques and separator sets of one of these 3-clique trees. By a line of reasoning similar to Example 6.4.3, it can be shown that any such distribution will be globally consistent. Therefore, such 3-clique trees are the analogs for the Kikuchi approximation of spanning trees for the Bethe approximation. I.e., they are the largest globally consistent substructures for this Kikuchi approximation.

This notion of largest globally consistent substructure turns out to play an important role in our analysis of the approximation error.

■ 6.5 Characterization of the error

This section is devoted to analysis of the error between the marginals of the target distribution $p(\mathbf{x})$, and the approximate marginals of distributions in $\vec{\mathbf{Q}}^*$. As with our analysis of TRP/BP updates in Chapter 5, the invariance principle of Theorem 6.4.1 allows us to derive an exact expression for this error. This exact relation, though conceptually interesting, is of limited practical use. However, it does enable us to derive various bounds on the approximation error. With an upper bound on the log partition function (see Chapter 7), these bounds are computable, and hence provide valuable information about the performance of these approximation techniques. For concreteness, we focus our analysis on the case of single node marginals. However, the analysis that we describe can be extended easily to marginals defined over larger subsets of the vertex set.

At a local minimum $\vec{\mathbf{Q}}^*$, the approximate single node marginals Q_s^* are specified by the core distribution $Q_{\mathbf{A}}^*$; the desired single node marginals P_s are specified by the target distribution $p(\mathbf{x})$. The essential intuition of equation (6.29) is the following: the core distribution $Q_{\mathbf{A}}^*$ can be viewed as a perturbed version of the target distribution $p(\mathbf{x})$, where the perturbation is caused by the terms $\sum_{\Delta \in \mathbf{R}} [\log Q^*_{\mathbf{A} \cup \Delta} - \log Q^*_{\mathbf{A}}]$. Therefore, the approximate marginals Q_s^* will also be perturbed versions of the true marginals P_s.

■ 6.5.1 Reformulation in exponential parameters

To make this intuition precise, it is convenient to shift our analysis to an exponential representation. (See Section 2.2 for background on exponential representations and their properties.) In the analysis to follow, we let $\theta = \{\theta_\alpha \mid \alpha \in \mathcal{A}\}$ with associated potentials $\boldsymbol{\phi} = \{\phi_\alpha \mid \alpha \in \mathcal{A}\}$ represent a minimal exponential parameterization for the target distribution $p(\mathbf{x})$ defined on the graph $\mathcal{G}$. We make the following definitions:

(a) let $\bar{\theta}$ be the exponential parameter distribution of the target distribution; i.e., $p(\mathbf{x}) \equiv p(\mathbf{x}; \bar{\theta})$.

(b) For any subset $\mathbf{B} \subseteq \mathbf{C}$, let $\mathcal{A}(\mathbf{B}) \subset \mathcal{A}$ be the set of indices α associated with elements of $\mathbf{B}$. Also define

$$\boldsymbol{\phi}_{\mathbf{B}} \triangleq \{\phi_\alpha \mid \alpha \in \mathcal{A}(\mathbf{B})\} \tag{6.53a}$$

$$\theta_{\mathbf{B}} \bullet \boldsymbol{\phi}_{\mathbf{B}} \triangleq \sum_{\alpha \in \mathbf{B}} \theta_\alpha \phi_\alpha \tag{6.53b}$$

(c) for any subset $\mathbf{B} \subset \mathbf{C}$, let $\Pi^{\mathbf{B}}$ be the projection operator onto this structure. That is:

$$\Pi^{\mathbf{B}}(\theta) \triangleq \{\theta_\alpha \mid \alpha \in \mathbf{B}\} \equiv \theta_{\mathbf{B}}$$

As an example, we have $P_{\mathbf{A}} = p(\mathbf{x}; \Pi^{\mathbf{A}}(\bar{\theta}))$.

(d) Similarly, for any parameter vector $\theta_{\mathbf{B}} = \{ \theta_\alpha \mid \alpha \in \mathcal{A}(\mathbf{B}) \}$ define an injection operator to the full set $\mathcal{A}$ via

$$\mathcal{I}(\theta_{\mathbf{B}}) = \begin{cases} \theta_\alpha & \alpha \in \mathcal{A}(\mathbf{B}) \\ 0 & \text{otherwise} \end{cases}$$

(e) let $\theta_{\mathbf{A}}$ and $\theta_{\mathbf{A} \cup \Delta}$ be exponential parameters corresponding to the core distribution $Q_{\mathbf{A}}$ and auxiliary distributions $Q_{\mathbf{A} \cup \Delta}$ respectively.

To illustrate the use of the exponential representation, let $\theta^*_{\mathbf{A}}$ and $\theta^*_{\mathbf{A} \cup \Delta}$ be exponential parameters corresponding to $Q^*_{\mathbf{A}}$ and $Q^*_{\mathbf{A} \cup \Delta}$ respectively. Then equation (6.29) of Theorem 6.4.1 can be restated in the following manner:

$$\mathcal{I}(\theta^*_{\mathbf{A}}) + \sum_{\Delta \in \mathbf{R}} \left[\mathcal{I}(\theta^*_{\mathbf{A} \cup \Delta}) - \mathcal{I}(\theta^*_{\mathbf{A}}) \right] \;=\; \bar{\theta} \tag{6.54}$$

In the analysis to follow, we will not always be strict in our use of the injection operator; when it is omitted, it should be understood implicitly that vectors are augmented with zeroes where necessary so that binary operations make sense.

■ 6.5.2 Exact error expression

With this set-up, we can now give an exact characterization of the error between the approximate and true single node marginals. With $\delta(x_s = j)$ denoting the indicator function for the random variable x_s to assume value j, the exact marginals and approximate marginals are given, respectively, by the following expressions:

$$P_{s;j} \;=\; p(x_s = j; \bar{\theta}) \;=\; \mathbb{E}_{\bar{\theta}}[\delta(x_s = j)] \tag{6.55a}$$

$$Q^*_{s;j} \;=\; p(x_s = j; \theta^*_{\mathbf{A}}) \;=\; \mathbb{E}_{\theta^*_{\mathbf{A}}}[\delta(x_s = j)] \tag{6.55b}$$

Using equation (6.54), we now write an exact expression for the error in the single node marginals:

$$P_{s;j} - Q^*_{s;j} = \mathbb{E}_{\theta^*_{\mathbf{A}}} \left[\left\{ \exp\Big(\sum_{\Delta \in \mathbf{R}} \left[\theta^*_{\mathbf{A} \cup \Delta} - \theta^*_{\mathbf{A}} \right] \bullet \boldsymbol{\phi}_{\mathbf{A} \cup \Delta}(\mathbf{x}) + \Phi(\theta^*_{\mathbf{A}}) - \Phi(\bar{\theta}) \Big) - 1 \right\} f_{s;j}(\mathbf{x}) \right] \tag{6.56}$$

Although theoretically interesting, the practical use of equation (6.56) is limited, as it is impossible to exactly compute the LHS.[4] Although the expectation is taken over the core distribution $p(\mathbf{x}; \theta^*_{\mathbf{A}})$ (which is tractable by assumption), within the expectation is a set of residual terms — namely, $\exp\Big(\sum_{\Delta \in \mathbf{R}} \left[\theta^*_{\mathbf{A} \cup \Delta} - \theta^*_{\mathbf{A}} \right] \bullet \boldsymbol{\phi}_{\mathbf{A} \cup \Delta}(\mathbf{x}) \Big)$ — that

[4] Indeed, if we could exactly compute this error, then we would have no need for approximate inference algorithms in the first place.

includes a term from (at the very least) every edge in $\mathcal{G}$ not covered by the core structure $\mathbf{A}$. Consequently, actually computing this expectation is as difficult as performing exact inference on the graph $\mathcal{G}$. This intractability motivates the development of bounds on the error.

■ 6.5.3 Exact expression for larger substructures

In deriving equation (6.56), we used the fact that approximations to the single node marginals are given by $Q^*_{s;j} = \mathbb{E}_{\theta^*_{\mathbf{A}}}[\delta(x_s = j)]$. In fact, the formulation of the variational problem (6.16) guarantees an additional set of conditions — namely

$$Q^*_{s;j} = \mathbb{E}_{\theta^*_{\mathbf{A} \cup \Delta}}[\delta(x_s = j)] \qquad \text{for all } \Delta \in \mathbf{R}$$

This relation follows because by the constraints associated with variational problems (6.16) and (6.20), the auxiliary distribution $Q^*_{\mathbf{A} \cup \Delta}(\mathbf{x}) = p(\mathbf{x}; \theta^*_{\mathbf{A} \cup \Delta})$ must marginalize down to the core distribution $Q^*_{\mathbf{A}}$. Thus, we obtain an alternative set of expressions for the difference $P_{s;j} - Q^*_{s;j}$, given in terms of expectations under the distribution $p(\mathbf{x}; \theta^*_{\mathbf{A} \cup \Delta})$:

$$\mathbb{E}_{\theta^*_{\mathbf{A} \cup \Delta}}\left[\left\{\exp\left(\theta^*_{\mathbf{A}} \bullet \phi_{\mathbf{A}}(\mathbf{x}) + \sum_{\Upsilon \in \mathbf{R}/\Delta} [\theta^*_{\mathbf{A} \cup \Upsilon} - \theta^*_{\mathbf{A}}] \bullet \phi_{\mathbf{A} \cup \Upsilon}(\mathbf{x}) + \Phi(\theta^*_{\mathbf{A} \cup \Delta}) - \Phi(\bar{\theta})\right) - 1\right\} \delta(x_s = j)\right] \tag{6.57}$$

This expression is valid for each $\Delta \in \mathbf{R}$.

More generally, error expressions of the form in equation (6.57) are valid for any distribution (formed from elements of $\vec{\mathbf{Q}}^*$) over a substructure that is globally consistent. As we saw in Section 6.4.4, the largest globally consistent substructure can be considerably larger than the augmented structures $\mathbf{A} \cup \widetilde{\Delta}$. For example, in the context of belief propagation, these structures are given by spanning trees (see Example 6.4.3).

■ 6.5.4 Bounds on the error

As in Chapter 5, we now derive bounds on the log error in the single node marginals $[\log P_{s;j} - \log Q_{s;j}]$. The first set of bounds in equation (6.58) is based on the fact that the approximations $Q_{s;j}$ arise from a distribution $p(\mathbf{x}; \theta^*_{\mathbf{A}})$ that is a perturbed version of the target distribution $p(\mathbf{x}; \bar{\theta})$. As a result, we can apply Proposition 3.3.1 to bound the log difference in the marginals. The second set of bounds in equation (6.59) is based on the same idea applied to $p(\mathbf{x}; \theta^*_{\mathbf{A} \cup \Delta})$.

Theorem 6.5.1. Let $\theta^*_{\mathbf{A}}$ be a local minimum of variational problem (6.16) giving rise to approximate single node marginals $Q_{s;j}$. Let $P_{s;j}$ be the exact marginals corresponding to the target distribution $p(\mathbf{x}; \bar{\theta})$. Then we have the following upper bound on the error $E_{s;j} = \log Q_{s;j} - \log P_{s;j}$:

$$E_{s;j} \leq D(\theta^*_{\mathbf{A}} \,\|\, \bar{\theta}) - \frac{1}{Q_{s;j}} \sum_{\Delta \in \mathbf{R}} [\theta^*_{\mathbf{A} \cup \Delta} - \theta^*_{\mathbf{A}}] \bullet \mathrm{cov}_{\theta^*_{\mathbf{A}}}\{\phi_{\Delta}, \delta(x_s = j)\} \tag{6.58}$$

where

$$\left[\theta^*_{\mathbf{A}\cup\Delta} - \theta^*_{\mathbf{A}}\right] \bullet \mathrm{cov}_{\theta^*_{\mathbf{A}}}\{\phi_{\mathbf{A}\cup\Delta}, \delta(x_s = j)\} \triangleq \sum_{\alpha\in\mathcal{A}(\mathbf{A}\cup\Delta)} [\theta^*_{\mathbf{A}\cup\Delta} - \theta^*_{\mathbf{A}}]_\alpha \,\mathrm{cov}_{\theta^*_{\mathbf{A}}}\{\phi_\alpha, \delta(x_s = j)\}$$

Similarly, for each $\Delta \in \mathbf{R}$, we have the following upper bound:

$$E_{s;j} \leq D(\theta^*_{\mathbf{A}\cup\Delta} \,\|\, \bar{\theta}) - \frac{1}{Q_{s;j}}\Big[\theta^*_{\mathbf{A}}\,\mathrm{cov}_{\theta^*_{\mathbf{A}\cup\Delta}}\{\phi_{\mathbf{A}}(\mathbf{x}), \delta(x_s = j)\} + \sum_{\Upsilon\in\mathbf{R}/\Delta} \left[\theta^*_{\mathbf{A}\cup\Upsilon} - \theta^*_{\mathbf{A}}\right] \bullet \mathrm{cov}_{\theta^*_{\mathbf{A}\cup\Delta}}\{\phi_\Upsilon, \delta(x_s = j)\}\Big] \quad (6.59)$$

Proof. To obtain equation (6.58), we first make use of equation (6.54) to write:

$$\bar{\theta} - \mathcal{I}(\theta^*_{\mathbf{A}}) = \sum_{\Delta\in\mathbf{R}} [\mathcal{I}(\theta^*_{\mathbf{A}\cup\Delta}) - \mathcal{I}(\theta^*_{\mathbf{A}})]$$

We then apply Proposition 3.3.1 to the function $\delta(x_s = j)$, and the parameter vectors $\bar{\theta}$ and $\theta^*_{\mathbf{A}}$.

To obtain equation (6.59), we use a different re-arrangement of equation (6.54) — namely:

$$\bar{\theta} - \mathcal{I}(\theta^*_{\mathbf{A}\cup\Delta}) = \mathcal{I}(\theta^*_{\mathbf{A}}) + \sum_{\Upsilon\in\mathbf{R}/\Delta;\Upsilon\neq\Delta} [\mathcal{I}(\theta^*_{\mathbf{A}\cup\Upsilon}) - \mathcal{I}(\theta^*_{\mathbf{A}})]$$

We then apply Proposition 3.3.1 to the function $\delta(x_s = j)$, and the parameter vectors $\bar{\theta}$ and $\theta^*_{\mathbf{A}\cup\Delta}$. □

Note that as with the analysis of TRP fixed points in Chapter 5, similar arguments can be applied to derive lower bounds on the error $E_{s;j}$. We do not write out these bounds in an explicit form here, as they can be deduced from equations (6.58) and (6.59).

It is also worthwhile noting that equation (6.59) holds for all $\Delta \in \mathbf{R}$. Therefore, for a given node s and state j, we can, in principle, compute the bound of equation (6.58) for the core distribution, as well as the bound of equation (6.59) for each $\Delta \in \mathbf{R}$, and then choose the tightest of all possible bounds. A similar freedom of choice was associated with the TRP bounds of Chapter 5, where we were free to choose any spanning tree of the graph to compute a bound.

A caveat associated with Theorem 6.5.1: it is still necessary to upper bound the log partition function $\Phi(\bar{\theta})$, which appears as part of the KL divergence. Techniques for obtaining such upper bounds are described in Chapter 7.

■ 6.6 Empirical simulations

In this section, we illustrate some properties of the approximations considered in this chapter, as well as the error bounds on their performance (as derived in Section 6.5), for some simple examples.

■ 6.6.1 When to use an approximation with more complex structure?

We first consider the question of when to use an approximation with more complex structure. For many problems, TRP/BP fixed points provide accurate approximations to the marginals, in which case it is undesirable to incur the higher computational cost associated with a more structured approximation. Here we illustrate with a simple example that error bounds may be useful in this context.

We begin by forming a distribution $p(\mathbf{x}; \theta^*)$ for a binary valued vector $\mathbf{x}$ with a random choice of attractive potentials (as described in Section 2.2.1) on the 3×3 grid. Let $P_{s;1}$ denote the actual marginal of $p(\mathbf{x}; \theta^*)$ at node s. We then run TRP/BP on the problem, thereby obtaining a set of approximate single node marginals $T^*_{s;1}$. We also use this TRP/BP fixed point to compute lower and upper bounds on the actual marginals P_s, as described in Section 5.5.3. The actual marginals $P_{s;1}$, the TRP/BP approximations $T^*_{s;1}$ and these upper and lower bounds are all plotted in panel (b) of Figure 6.14. As discussed in Example 6.3.2, the core structure underlying the TRP/BP approximation is a fully disconnected graph; as a reminder, we have plotted this structure in panel (a).

The TRP/BP approximation in panel (b) is quite poor, as reflected by the relative looseness of the bounds. Note how the TRP/BP approximation lies beneath the lower bound on the actual marginals for certain nodes (e.g., node 5), so that we know that it is a poor approximation even without having to see the actual marginals themselves.

Given that the TRP/BP approximation is poor, we are motivated to try an approximation with a more complex core structure. Here we illustrate a $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ approximation using as the core structure the spanning tree illustrated in panel (c), and a set of residual terms formed by the 4 edges remaining in the 3×3 grid that are not included in this spanning tree. We run the generalized message-passing Algorithm 6.4.1 in order to find a fixed point of the associated variational problem. The resultant approximate single node marginals $Q^*_{s;1}$ are plotted in comparison the actual marginals $P_{s;1}$ in panel (d). Note that the approximation is excellent. As before, we can use the fixed point $\vec{\mathbf{Q}}^*$ and Theorem 6.5.1 to calculate upper and lower bounds on the actual marginals. In this case, these bounds are quite tight, which tells us the approximation is quite good.

■ 6.6.2 Choice of core structure

Another important question is the effect of varying the choice of core structure used in a $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ approximation. Here we investigate the effect of different choices of core structure of the same complexity — in this case, two different spanning trees of the same graph.

To investigate this question, we first formed a distribution $p(\mathbf{x}; \theta^*)$ for a binary valued vector $\mathbf{x}$ with a random choice of mixed potentials (as described in Section 2.2.1) on the fully connected graph on 9 nodes (K_9). We then computed the maximum and minimum weight spanning trees with Kruskal's algorithm [107, 116], using $|\theta^*_{st}|$ as the weight on edge (s, t). Using these spanning trees as the core structures, we then

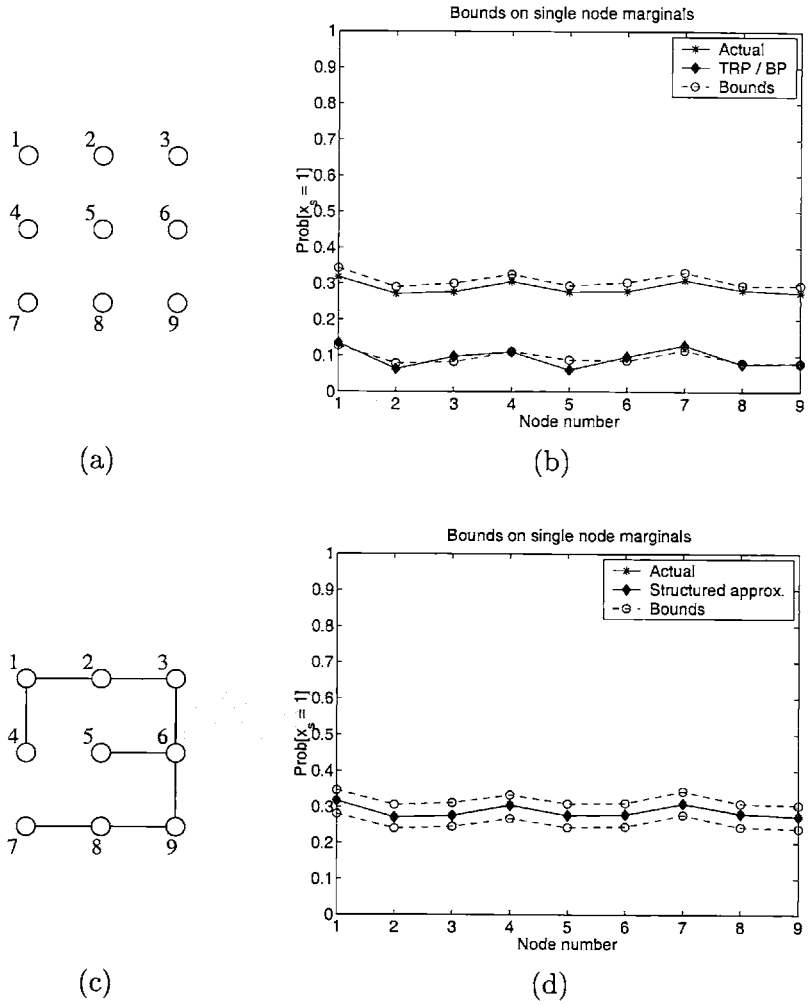


Figure 6.14. Changes in approximation accuracy and error bounds for TRP/BP compared to a $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ approximation using a spanning tree core on the 3×3 grid. (a) Fully disconnected core for BP/TRP. (b) Plots of actual $P_{s;1}$ and approximate single node marginals versus node number). Also shown are lower and upper bounds on the actual marginals, computed from the TRP fixed point. The TRP/BP approximation is quite poor; the bounds on the exact marginals are relatively loose. (c) Spanning tree core $\mathbf{A}$ for $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ approximation. (d) Plots of the actual and approximate marginals, as well as error bounds. The approximation using a spanning tree core is very accurate, as reflected by tighter bounds.

computed approximations $\vec{\mathbf{Q}}^*$ for the corresponding $\mathcal{G}_{\mathbf{A};\mathbf{R}}$ approximations, where each term in the residual set consisted of a single edge.

Panels (a) and (b) of Figure 6.15 show the results for the minimum and maximum weight spanning trees respectively. In each panel, we plot the approximate marginals $Q^*_{s;1}$ and the actual marginals $P_{s;1}$ versus node number, as well as upper and lower

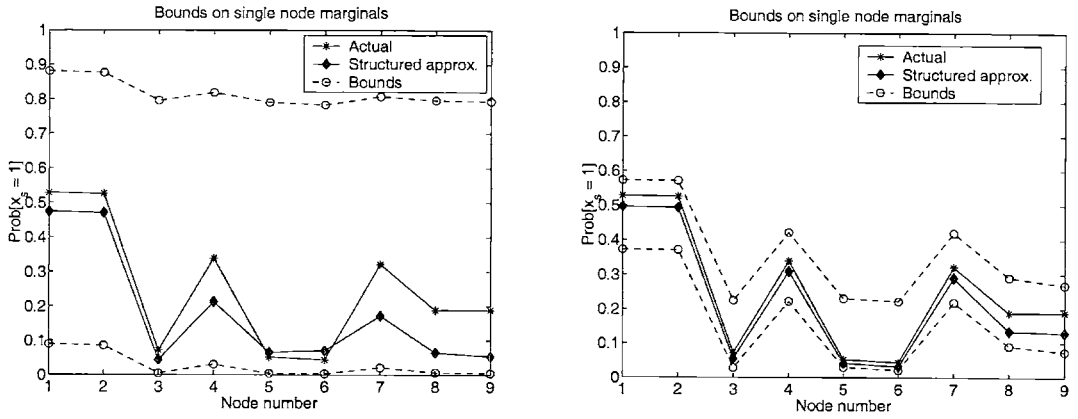


Figure 6.15. Approximations of the same problem based on different choice of spanning tree cores. Each panel shows the approximate marginals $Q^*_{s;1}$ and the actual marginals $P_{s;1}$ versus node number, as well as upper and lower bounds on the actual marginals computed using the fixed point $\vec{\mathbf{Q}}^*$. (a) Approximation based on minimum weight spanning tree is poor, and bounds are quite loose. (b) Approximation based on maximum weight spanning tree is better, and bounds are correspondingly tighter.

bounds on the actual marginal computed from the fixed points $\vec{\mathbf{Q}}^*$ using Theorem 6.5.1. The approximation in panel (a), based on the minimum weight spanning tree, is poor, and the corresponding bounds on the actual marginal are quite loose. In contrast, the approximation based on the maximum weight spanning tree, as shown in panel (b), is better; moreover, the bounds are now tighter.

Note that the cost associated with computing either set of approximations is equivalent, yet the quality of the resulting approximation varies substantially. Although we have simply used a reasonable heuristic here, this example illustrates that the choice of core structure, even when restricted to a fixed class (e.g., spanning trees), is important. We shall discuss this issue of choosing a core structure at more length in Chapter 8.

■ 6.7 Discussion

This chapter developed a unifying framework for a wide class of more advanced techniques for approximate inference (including BP as a special case). All of these techniques, like belief propagation and the Bethe free energy, are based on minimizing approximations to the Kullback-Leibler divergence. The minimizing arguments of these

variational problems are then taken as approximations to local marginal distributions. An ancillary contribution of this chapter is to unify two previous extensions to belief propagation: the Kikuchi approximations of Yedidia et al. [180], and the expectation-propagation updates of Minka [131].

The analysis of this chapter demonstrated that the idea of reparameterization, first introduced in Chapter 5, is more generally applicable to all of the approximations considered in this chapter. As a consequence, most of the significant results from Chapter 5 on tree-based reparameterization (or belief propagation) carry over in a natural way to the more advanced techniques analyzed in this chapter. In particular, we proved the existence of fixed points, and showed that they all satisfy a generalized form of the invariance principle from Chapter 5. Moreover, we developed a generalized message-passing (or reparameterization) algorithm for computing fixed points. Lastly, we analyzed the error that arises in using these approximations, and developed computable bounds on this error. Given the understanding and insight provided by this analysis, it is interesting to consider the application of these more advanced methods to large-scale problems to which BP has been successfully applied, including problems in image processing, artificial intelligence, and iterative decoding.

Chapter 7

Upper bounds based on convex combinations

7.1 Introduction

A fundamental quantity associated with any graph-structured distribution is the log partition function. With the exception of certain special cases, actually computing the log partition function, though a straightforward summation in principle, is NP-hard [39,46] due to the exponential number of terms. Therefore, an important problem is either to approximate or obtain bounds on the log partition function. There is a large literature on approximation algorithms for the log partition function [e.g., 102,139,140]. A related (and possibly more ambitious) goal is to obtain upper and lower bounds [e.g., 12,92,93,105,154,176]. The applicability of such bounds on the log partition function is wide; possible uses include approximate inference [e.g., 94,96], model fitting [e.g., 105], and large deviations analysis [e.g., 158] .

An important property of the log partition function is its convexity (see Section 2.2). Mean field theory [e.g., 105], as presented in Section 2.3.1, can be viewed as exploiting one property of a convex function: namely, that the first order tangent approximation is always an underestimate [20]. In Chapter 3, we exploited another fundamental property of convex functions — namely, Jensen's inequality [41] — in order to derive a new class of upper bounds applicable to an arbitrary undirected graphical model. These upper bounds were based on taking a particular convex combination of exponential parameter vectors.

In this chapter, we analyze this new class of bounds in more detail, focusing on the case where the exponential parameter vectors are drawn from some tractable class for which exact computations can be performed efficiently. The canonical example of such a tractable substructure is a tree embedded within the original graph. The weights in the convex combination are specified by a probability distribution $\vec{\mu}$ over the set of tractable substructures.

For any given log partition function, there is an entire family of upper bounds, indexed by the choice of exponential parameters as well as the probability distribution $\vec{\mu}$. It is therefore natural to consider the problem of optimizing both the choice of exponential parameters, and the distribution over tractable subgraphs, so as to obtain

the tightest possible bounds on the log partition function. This optimization problem turns out to have an interesting structure. Indeed, we prove that the problem is jointly convex in both the distribution $\vec{\mu}$ and the exponential parameter vectors, so that there is a unique optimal pair that yields the tightest possible upper bound. This uniqueness is in sharp contrast to mean field theory, where the associated optimization problem is well-known to suffer from multiple local minima, even for relatively simple problems [e.g., 92].

The line of analysis that we develop here is quite general, in that it applies to discrete random variables assuming an arbitrary number of states m, and in principle to arbitrary sizes of clique potentials. The only restriction that our analysis imposes on the approximating structures themselves is that they correspond to a triangulated graph. However, in order to bring our development into sharp focus, this chapter treats the special case of binary nodes ($m = 2$) and pairwise clique potentials; moreover, we assume that the set of tractable substructures, denoted by $\mathfrak{T}$, corresponds to the set of all spanning trees of the graph $\mathcal{G}$. Based on an understanding of this case, the modifications necessary to deal with higher state numbers ($m > 2$), larger clique sizes, and more complex approximating structures will be clear.

A major challenge to be overcome lies in the dimension of the problem. The length of $\vec{\mu}$ is equal to the number of spanning trees in the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Reasonably complex graphs tend have a very large number of spanning trees — one which grows prohibitively quickly in the number of nodes $N = |\mathcal{V}|$. The collection of exponential parameter vectors is even larger by an additional factor of $\mathcal{O}(N)$. Fortunately, we are able to sidestep this combinatorial explosion by applying the theory of Lagrangian duality [20]. Indeed, in the dual formulation, the entire collection of exponential parameters is replaced by a single vector of length $N + |\mathcal{E}|$. In addition, the dual function consists of a convex combination of entropy terms of tree-structured distributions, each of which can be decomposed as a sum of single node and edge terms. This local decomposition is the crucial property that permits efficient optimization of the bounds. In particular, for a fixed distribution over spanning trees of the graph, we develop a constrained Newton's method to optimize efficiently the choice of exponential parameters. Simultaneously optimizing both the choice of the exponential parameters and the distribution $\vec{\mu}$ over spanning trees requires a more intensive but computationally tractable algorithm with an inner and outer loop. Interestingly, steps in the outer loop correspond to solving a maximum weight spanning tree problem [107], which can be interpreted as finding the tree that best fits the current data [see 36].

This chapter is organized in the following manner. In Section 7.1.1, we introduce the notation and definitions required for analysis. In Section 7.1.2, we derive the basic form of the upper bounds to be studied in this chapter. The dual formulation of these bounds, which is essential in avoiding the combinatorial explosion described above, is developed in Section 7.2. Section 7.3 builds on this dual formulation by stating and characterizing the optimal form of the upper bounds, first for the case of a fixed distribution, and secondly when both the distribution $\vec{\mu}$ and the collection of exponential

parameter vectors are allowed to vary. Section 7.4 is devoted to more practical issues: we present algorithms for computing the optimal form of the bounds specified in Section 7.3. We also present the results of these techniques in application to bounding the log partition function of randomly generated distributions. We finish up in Section 7.5 with a summary and extensions to the work described here.

■ 7.1.1 Set-up

The analysis of this chapter makes heavy use of exponential representations of distributions, and the associated Legendre transform between exponential and mean parameters. The reader should consult Section 2.2 for the relevant background. In this section, we set up the notation necessary for subsequent analysis. So as to provide a relatively self-contained and more readable presentation, we duplicate some material from Section 3.3.3 of Chapter 3, in which upper bounds of this nature were first introduced.

Let $\mathfrak{T} = \mathfrak{T}(\mathcal{G})$ denote the set of all spanning trees of $\mathcal{G}$. We use the symbol $\mathcal{T}$ to refer to a spanning tree in $\mathfrak{T}$. The number $|\mathfrak{T}(\mathcal{G})|$ of spanning trees in a graph $\mathcal{G}$ is typically quite large; for instance, a well-known result of Cayley [168] states that the complete graph K_N on N nodes has N^{N-2} spanning trees. More generally, the number of spanning trees in a graph can be computed via the Matrix-Tree theorem [168].

We now define a probability distribution over the set of spanning trees $\mathfrak{T} = \mathfrak{T}(\mathcal{G})$:

$$\vec{\mu} = \{ \mu(\mathcal{T}), \mathcal{T} \in \mathfrak{T} \mid \mu(\mathcal{T}) \geq 0; \quad \sum_{\mathcal{T} \in \mathfrak{T}} \mu(\mathcal{T}) = 1 \} \tag{7.1}$$

The *support* of $\vec{\mu}$ is defined as

$$\mathrm{supp}(\vec{\mu}) \triangleq \{ \mathcal{T} \in \mathfrak{T} \mid \mu(\mathcal{T}) > 0 \} \tag{7.2}$$

In the sequel, it will also be of interest to consider the probability that a given edge $e \in \mathcal{E}$ appears in a spanning tree $\mathcal{T}$ chosen randomly under the distribution $\vec{\mu}$.

Definition 7.1.1. Given a distribution $\vec{\mu}$ over spanning trees, the *edge appearance probability* of an edge $e \in \mathcal{E}$ is defined as follows:

$$\mu_e \triangleq \mathbb{E}_{\vec{\mu}}\{\delta[e \in \mathcal{T}]\} = \Pr_{\vec{\mu}}\{e \in \mathcal{T}\} \tag{7.3}$$

where $\delta[e \in \mathcal{T}]$ is the indicator function for edge e to appear in tree $\mathcal{T}$. I.e., this is the probability that edge e belongs to a spanning tree chosen randomly under distribution $\vec{\mu}$.

For a random vector $\mathbf{x}$ taking values in $\{0,1\}^N$, let θ^* denote the minimal exponential parameter of a distribution $p(\mathbf{x};\theta^*)$ defined on the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$:

$$p(\mathbf{x};\theta^*) = \exp\Big\{ \sum_{s \in \mathcal{V}} \theta^*_s x_s + \sum_{(s,t) \in \mathcal{E}} \theta^*_{st} x_s\, x_t - \Phi(\theta^*) \Big\} \tag{7.4}$$

We refer to this quantity as the *target distribution.*

For each spanning tree $\mathcal{T} \in \mathfrak{T}$, let $\theta(\mathcal{T})$ be an exponential parameter vector of the same dimension as θ^* that respects the structure of $\mathcal{T}$. To be explicit, if $\mathcal{T}$ is defined by an edge set $\mathcal{E}(\mathcal{T}) \subset \mathcal{E}$, then $\theta(\mathcal{T})$ must have zeros in all elements corresponding to edges not in $\mathcal{E}(\mathcal{T})$. For a binary process, such a tree-structured distribution has the form:

$$p(\mathbf{x}; \theta(\mathcal{T})) = \exp\Big\{ \sum_{s \in \mathcal{V}} \theta(\mathcal{T})_s x_s + \sum_{(s,t) \in \mathcal{E}(\mathcal{T})} \theta(\mathcal{T})_{st}\, x_s x_t - \Phi(\theta(\mathcal{T}))\Big\} \tag{7.5}$$

Since any spanning tree on a connected graph with N nodes has $N-1$ edges, the parameter vector $\theta(\mathcal{T})$ has $d(\theta(\mathcal{T})) \triangleq 2N-1$ non-zero elements for a binary-valued process.

For compactness in notation, let

$$\boldsymbol{\theta} = \{\theta(\mathcal{T}) \mid \mathcal{T} \in \mathfrak{T}\} \tag{7.6}$$

denote the full collection of tree-structured exponential parameter vectors. This quantity can be viewed as a large vector with $[(2N-1)\,|\mathfrak{T}(\mathcal{G})|]$ non-zero elements. The notation $\theta(\mathcal{T})$ specifies those subelements of $\boldsymbol{\theta}$ corresponding to spanning tree $\mathcal{T}$.

■ 7.1.2 Basic form of bounds

The central idea is that of a convex combination of tree-structured parameter vectors:

Definition 7.1.2. Given a distribution $\vec{\mu}$ and a collection of exponential vectors $\boldsymbol{\theta}$, a *convex combination* of exponential parameter vectors is defined via the expectation:

$$\mathbb{E}_{\vec{\mu}}[\boldsymbol{\theta}] \triangleq \sum_{\mathcal{T} \in \mathfrak{T}} \mu(\mathcal{T})\theta(\mathcal{T}) \tag{7.7}$$

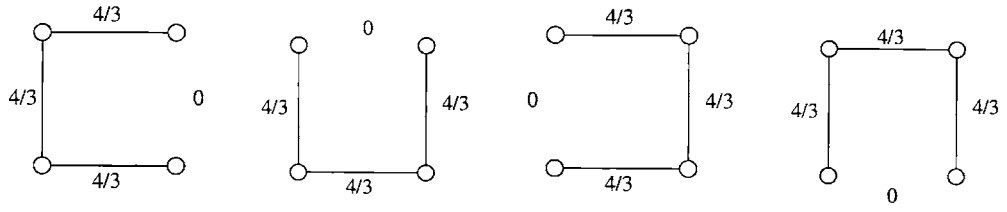
We are especially interested in sets of approximating points $\boldsymbol{\theta}$ for which there exists a convex combination that is equal to θ^*. Accordingly, we define the following set of pairs $(\boldsymbol{\theta}; \vec{\mu})$:

$$\mathcal{A}(\theta^*) \triangleq \Big\{ (\boldsymbol{\theta}; \vec{\mu}) \;\Big|\; \mathbb{E}_{\vec{\mu}}[\boldsymbol{\theta}] = \theta^* \Big\} \tag{7.8}$$

It is not difficult to see that $\mathcal{A}(\theta^*)$ is never empty.

Example 7.1.1. To illustrate these definitions, consider a binary distribution defined by a single loop on 4 nodes, as shown in Figure 7.1. Consider a target distribution of the form

$$p(\mathbf{x}; \theta^*) = \exp\{x_1x_2 + x_2x_3 + x_3x_4 + x_4x_1 - \Phi(\theta^*)\}$$



(a) $\theta(\mathcal{T}_1)$ (b) $\theta(\mathcal{T}_2)$ (c) $\theta(\mathcal{T}_3)$ (d) $\theta(\mathcal{T}_4)$

Figure 7.1. A convex combination of four distributions $p(\mathbf{x}; \theta(\mathcal{T}_i))$, each defined by a spanning tree $\mathcal{T}_i$, is used to approximate the target distribution $p(\mathbf{x}; \theta^*)$ on the single-cycle graph.

That is, the target distribution is specified by the minimal parameter $\theta^* = [0\ 0\ 0\ 0\ 1\ 1\ 1\ 1]$, where the zeros represent the fact that $\theta^*_s = 0$ for all $s \in \mathcal{V}$. The tractable class consists of the four possible spanning trees $\mathfrak{T} = \{\mathcal{T}_i \mid i = 1, \ldots, 4\}$ on a single cycle on four nodes. We define a set of associated exponential parameters $\boldsymbol{\theta} = \{\theta(\mathcal{T}_i)\}$ as follows:

$$\begin{aligned}
\theta(\mathcal{T}_1) &= (4/3)\,[0\ 0\ 0\ 0\ \ 1\ 1\ 1\ 0] \\
\theta(\mathcal{T}_2) &= (4/3)\,[0\ 0\ 0\ 0\ \ 1\ 1\ 0\ 1] \\
\theta(\mathcal{T}_3) &= (4/3)\,[0\ 0\ 0\ 0\ \ 1\ 0\ 1\ 1] \\
\theta(\mathcal{T}_4) &= (4/3)\,[0\ 0\ 0\ 0\ \ 0\ 1\ 1\ 1]
\end{aligned}$$

Finally, we choose $\mu(\mathcal{T}_i) = 1/4$ for all $\mathcal{T}_i \in \mathfrak{T}$. It is not difficult to check that this choice of a uniform distribution ensures that $\mathbb{E}_{\vec{\boldsymbol{\mu}}}[\boldsymbol{\theta}] = \theta^*$; that is, the specified pair $(\boldsymbol{\theta}; \vec{\boldsymbol{\mu}})$ belongs to $\mathcal{A}(\theta^*)$.

Recall from Lemma 2.2.1 that the log partition function Φ is convex as a function of θ. This property allows us to apply Jensen's inequality [41] to a convex combination specified by a pair $(\boldsymbol{\theta}, \vec{\boldsymbol{\mu}}) \in \mathcal{A}(\theta^*)$; doing so yields the following result:

Proposition 7.1.1. For any pair $(\boldsymbol{\theta}, \vec{\boldsymbol{\mu}}) \in \mathcal{A}(\theta^*)$, the following upper bound is valid:

$$\Phi(\theta^*) \leq \mathbb{E}_{\vec{\boldsymbol{\mu}}}[\Phi(\theta(\mathcal{T}))] \triangleq \sum_{\mathcal{T} \in \mathfrak{T}} \mu(\mathcal{T}) \Phi(\theta(\mathcal{T})) \tag{7.9}$$

Note that the bound of equation (7.9) is a function of both the distribution $\vec{\boldsymbol{\mu}}$ over spanning trees; and the collection of tree-structured exponential parameter vectors $\boldsymbol{\theta}$. The primary goal of this chapter is to optimize these choices so as to minimize the RHS of equation (7.9), thereby obtaining the tightest possible upper bound of the form in Proposition 7.1.1. We shall consider first the problem of optimizing the choice of $\boldsymbol{\theta}$ for a fixed $\vec{\boldsymbol{\mu}}$; and then the joint optimization of $\boldsymbol{\theta}$ and $\vec{\boldsymbol{\mu}}$. Despite the relatively simple form of equation (7.9), these optimization problems turn out to have a rich and interesting structure.

■ 7.2 Dual formulation with fixed $\bar{\mu}$

In this section, we develop a Lagrangian dual formulation of the bounds of equation (7.9). For a fixed distribution $\bar{\mu}$, consider the following constrained optimization problem:

$$\begin{cases} \min_{\theta} \mathbb{E}_{\bar{\mu}}[\Phi(\theta(\mathcal{T}))] \\ \text{s. t} \qquad \mathbb{E}_{\bar{\mu}}[\boldsymbol{\theta}] = \theta^* \end{cases} \tag{7.10}$$

With $\bar{\mu}$ fixed, the upper bound $\mathbb{E}_{\bar{\mu}}[\Phi(\theta(\mathcal{T})]$ is strictly convex as a function of $\boldsymbol{\theta}$, and the associated constraint is linear in $\boldsymbol{\theta}$.

We assume that $\bar{\mu}$ is chosen such that the associated edge appearance probabilities $\bar{\mu}_e = \Pr_{\bar{\mu}}\{e \in \mathcal{T}\}$ are all strictly positive. I.e., all edges $e \in \mathcal{E}$ appear in at least one tree $\mathcal{T} \in \text{supp}(\bar{\mu})$. This assumption is necessary to ensure that constraint set $\{ \boldsymbol{\theta} \mid (\boldsymbol{\theta}, \bar{\mu}) \in \mathcal{A}(\theta^*) \}$ is non-empty. By standard results in nonlinear programming [20], problem (7.10) has a unique global minimum, attained at $\widehat{\boldsymbol{\theta}} \equiv \widehat{\boldsymbol{\theta}}(\bar{\mu})$. In principle, a variety of methods could be used to solve the convex program (7.10) (see, e.g., [20]). However, an obvious concern is the dimension of the parameter vector $\boldsymbol{\theta}$; in particular, it is directly proportional to $|\mathfrak{T}|$, the number of spanning trees in $\mathcal{G}$, which is typically very large.

As we will show in this section, the theory of convex duality allows us to neatly avoid this combinatorial explosion. In particular, we show that the Lagrangian dual of problem (7.10) depends on a vector λ of length $N + |\mathcal{E}|$, which has the form:

$$\lambda = \{\lambda_s,\ s \in \mathcal{V}; \quad \lambda_{st},\ (s,t) \in \mathcal{E}\} \tag{7.11}$$

This vector can be viewed as a set of parameters defining the local marginal distributions of a binary process on the single nodes and edges of the original graph $\mathcal{G}$ as follows:

$$p(x_s; \lambda) \triangleq [1 - \lambda_s;\ \lambda_s]' \tag{7.12a}$$

$$p(x_s, x_t; \lambda) \triangleq \begin{pmatrix} [1 + \lambda_{st} - \lambda_s - \lambda_t] & \lambda_{st} - \lambda_t \\ \lambda_{st} - \lambda_s & \lambda_{st} \end{pmatrix} \tag{7.12b}$$

To ensure that these definitions make sense as marginal distributions (i.e., their elements lie between zero and one), the vector λ must belong to the following polytope:

$$\mathbb{L}(\mathcal{G}) = \{ \lambda \mid 0 \leq \lambda_{st} \leq \lambda_s \leq 1; \quad \lambda_s + \lambda_t \leq 1 + \lambda_{st} \quad \forall\, s \in \mathcal{V},\ (s,t) \in \mathcal{E} \} \tag{7.13}$$

Let $\widehat{\boldsymbol{\theta}} = \{ \widehat{\theta}(\mathcal{T}) \mid \mathcal{T} \in \mathfrak{T} \}$ denote the optimum of problem (7.10). The significance of λ is in specifying this optimum in a very compact fashion. For each tree $\mathcal{T} \in \mathfrak{T}$, let $\Pi^{\mathcal{T}}(\lambda)$ denote the projection of λ onto the spanning tree $\mathcal{T}$. Explicitly,

$$\Pi^{\mathcal{T}}(\lambda) \triangleq \{\lambda_s,\ \lambda_{st} \mid s \in \mathcal{V}; \quad (s,t) \in \mathcal{E}(\mathcal{T}) \} \tag{7.14}$$

consists only of those elements of λ belonging to single nodes, or elements of the edge set $\mathcal{E}(\mathcal{T}) \subset \mathcal{E}$ of the tree $\mathcal{T}$.

Via equation (7.12), any such vector $\Pi^{\mathcal{T}}(\lambda)$ defines a set of marginal distributions for each node $s \in \mathcal{V}$ and edge $(s,t) \in \mathcal{E}(\mathcal{T})$. These marginals provide an explicit construction of a distribution $p(\mathbf{x}; \Pi^{\mathcal{T}}(\lambda))$ via the usual factorization of tree-structured distributions implied by the junction tree representation (see Section 2.1.5) — viz.:

$$p(\mathbf{x}; \Pi^{\mathcal{T}}(\lambda)) \triangleq \prod_{s \in \mathcal{V}} p(x_s; \lambda) \prod_{(s,t) \in \mathcal{E}(\mathcal{T})} \frac{p(x_s, x_t; \lambda)}{p(x_s; \lambda)\, p(x_t; \lambda)} \tag{7.15}$$

The proof of Proposition 7.2.1 below shows that the optimal dual parameter $\widehat{\lambda}$ specifies the full collection of optimal exponential parameters $\widehat{\boldsymbol{\theta}}$ via the relation:

$$p(\mathbf{x}; \widehat{\theta}(\mathcal{T})) = p(\mathbf{x}; \Pi^{\mathcal{T}}(\widehat{\lambda})) \qquad \text{for all } \mathcal{T} \in \mathfrak{T} \tag{7.16}$$

That is, at the optimum, a single vector $\widehat{\lambda}$ of length $N + |\mathcal{E}|$ suffices to specify the full collection $\widehat{\boldsymbol{\theta}} = \{ \widehat{\theta}(\mathcal{T}) \mid \mathcal{T} \in \mathfrak{T} \}$. Consequently, the dual formulation reduces the problem dimension from the size of $\boldsymbol{\theta}$ (which is proportional to $|\mathfrak{T}|$) down to the dimension of λ (namely, $N + |\mathcal{E}|$). It is this massive reduction in the problem dimension that permits efficient optimization.

An insight that emerges from our analysis is that the collection of tree-structured distributions $\{ p(\mathbf{x}; \widehat{\theta}(\mathcal{T})) \mid \mathcal{T} \in \mathfrak{T} \}$ has the following remarkable property:

(a) For every $p(\mathbf{x}; \widehat{\theta}(\mathcal{T}))$, the single node marginal probability $p(x_s = 1; \widehat{\theta}(\mathcal{T}))$ is equal to the same constant $\widehat{\lambda}_s$, for all vertices $s \in \mathcal{V}$.

(b) For every tree-structured distribution $p(\mathbf{x}; \widehat{\theta}(\mathcal{T}))$ for which the tree $\mathcal{T}$ includes edge (s,t), the corresponding marginal probability $p(x_s = 1, x_t = 1; \widehat{\theta}(\mathcal{T}))$ is equal to the same constant $\widehat{\lambda}_{st}$.

These conditions are very similar to the consistency conditions satisfied by any fixed point of tree-based reparameterization (see Chapter 5). Not surprisingly then, the dual function of Proposition 7.2.1 has a very close relation with the Bethe free energy, as we point out in Section 7.3.4.

■ 7.2.1 Explicit form of dual function

We now state and derive the dual form of problem (7.10). Let $\Psi(\Pi^{\mathcal{T}}(\lambda))$ be the negative entropy of the tree-structured distribution $p(\mathbf{x}; \Pi^{\mathcal{T}}(\lambda))$ defined in equation (7.15), and recall the definition of the polytope $\mathbb{L}(\mathcal{G})$ given in equation (7.13).

Proposition 7.2.1 (Dual formulation). For a fixed weight vector $\bar{\boldsymbol{\mu}}$, we have the equivalent dual formulation of problem (7.10):

$$\min_{\boldsymbol{\theta} \text{ s.t } \mathbb{E}_{\bar{\mu}}[\boldsymbol{\theta}] = \theta^*} \mathbb{E}_{\bar{\boldsymbol{\mu}}}[\Phi(\theta(\mathcal{T}))] = \max_{\lambda \in \mathbb{L}(\mathcal{G})} \mathcal{Q}(\lambda; \bar{\boldsymbol{\mu}}; \theta^*) \tag{7.17}$$

where

$$\mathcal{Q}(\lambda; \bar{\mu}; \theta^*) \triangleq -\mathbb{E}_{\bar{\mu}}[\Psi(\Pi^{\mathcal{T}}(\lambda))] + \sum_{\alpha} \lambda_\alpha \theta^*_\alpha \tag{7.18}$$

Proof. For the fixed $\bar{\mu}$, we consider the Lagrangian dual function associated with problem (7.10):

$$\mathcal{Q}(\lambda; \bar{\mu}; \theta^*) \quad = \quad \inf_{\boldsymbol{\theta}} \left\{ \mathcal{L}(\boldsymbol{\theta}; \lambda; \bar{\mu}; \theta^*) \right\} \tag{7.19}$$

where the Lagrangian is defined by

$$\mathcal{L}(\boldsymbol{\theta}; \lambda; \bar{\mu}; \theta^*) = \mathbb{E}_{\bar{\mu}}[\Phi(\theta(\mathcal{T})] + \sum_{\alpha} \lambda_\alpha \{\theta^*_\alpha - \mathbb{E}_{\bar{\mu}}[\theta(\mathcal{T})_\alpha]\} \tag{7.20}$$

The function $\mathcal{Q}(\lambda; \bar{\mu}; \theta^*)$ is a function of the dual variables λ; in particular, λ_α is a Lagrange multiplier associated with the constraint $\theta^*_\alpha - \mathbb{E}_{\bar{\mu}}[\theta(\mathcal{T})_\alpha] = 0$. In addition to these constraints, each $\theta(\mathcal{T})$ is restricted to correspond to a tree-structured distribution, meaning that certain elements $\theta(\mathcal{T})_\beta$ must be zero. We enforce these zero constraints explicitly without Lagrange multipliers.

Now the Lagrangian is also strictly convex as a function of $\boldsymbol{\theta}$, so that the infimum of equation (7.19) is attained at some value $\widehat{\boldsymbol{\theta}} = \{\widehat{\theta}(\mathcal{T})\}$. By taking derivatives of the Lagrangian with respect to $\boldsymbol{\theta}$, we obtain the stationary conditions for the optimum:

$$\bar{\mu}(\mathcal{T})\{\mathbb{E}_{\widehat{\theta}(\mathcal{T})}[\phi_\alpha] - \widehat{\lambda}_\alpha\} = 0 \tag{7.21}$$

where

$$\phi_\alpha(x) = \begin{cases} x_s & \text{if } \alpha = s \in \mathcal{V} \\ x_s x_t & \text{if } \alpha = (s,t) \in \mathcal{E} \end{cases}$$

If $\bar{\mu}(\mathcal{T}) = 0$, then the approximating parameter $\theta(\mathcal{T})$ plays no role in the problem, so that we can simply ignore it. Otherwise, if $\bar{\mu}(\mathcal{T}) > 0$, equation (7.21) implies that for all indices $\alpha \in \mathcal{V} \cup \mathcal{E}(\mathcal{T})$, the Lagrange multipliers are connected to the optimal approximating parameters $\widehat{\theta}(\mathcal{T})$ via the relation:

$$\mathbb{E}_{\widehat{\theta}(\mathcal{T})}[\phi_\alpha] = \widehat{\lambda}_\alpha \tag{7.22}$$

Recall from Section 2.2.4 that the expectations $\mathbb{E}_{\widehat{\theta}(\mathcal{T})}[\phi_\alpha]$ define a set of mean parameters $\eta(\widehat{\theta}(\mathcal{T}))$ that are dually coupled via the Legendre transform to the exponential parameter $\widehat{\theta}(\mathcal{T})$. Therefore, equation (7.22) has two important implications:

(a) for all $\mathcal{T} \in \text{supp}(\bar{\mu})$ and nodes $s \in \mathcal{V}$, the mean parameters $\mathbb{E}_{\widehat{\theta}(\mathcal{T})}[x_s]$ are all equal to a common value $\widehat{\lambda}_s$

(b) similarly, for all $\mathcal{T} \in \text{supp}(\bar{\mu})$ that include edge (s,t), the mean parameters $\mathbb{E}_{\widehat{\theta}(\mathcal{T})}[x_s\, x_t]$ are all equal to a common value $\widehat{\lambda}_{st}$

Since $\widehat{\theta}(\mathcal{T})$ and $\Pi^{\mathcal{T}}(\widehat{\lambda})$ are coupled via the Legendre transform, they correspond to the exponential parameter and mean parameter respectively of the tree-structured distribution $p(\mathbf{x}; \widehat{\theta}(\mathcal{T})) \equiv p(\mathbf{x}; \Pi^{\mathcal{T}}(\lambda))$. Here, as is common in exponential families, we are using the exponential parameter $\widehat{\theta}(\mathcal{T})$ and the mean parameter $\Pi^{\mathcal{T}}(\lambda)$ interchangeably to index the same distribution.[1]

By the Legendre duality between the log partition function and the negative entropy function, we have the relation:

$$\Phi(\widehat{\theta}(\mathcal{T})) = \sum_{\alpha} \widehat{\theta}(\mathcal{T})_{\alpha} \widehat{\lambda}_{\alpha} - \Psi(\Pi^{\mathcal{T}}(\widehat{\lambda})) \tag{7.23}$$

where $\Psi(\Pi^{\mathcal{T}}(\widehat{\lambda}))$ is the negative entropy of $p(\mathbf{x}; \widehat{\theta}(\mathcal{T})) \equiv p(\mathbf{x}; \Pi^{\mathcal{T}}(\lambda))$. Substituting equation (7.23) into equation (7.20) yields an explicit expression for the Lagrangian dual function:

$$\mathcal{Q}(\widehat{\lambda}; \bar{\mu}; \theta^*) = -\mathbb{E}_{\bar{\mu}}[\Psi(\Pi^{\mathcal{T}}(\widehat{\lambda}))] + \sum_{\alpha} \widehat{\lambda}_{\alpha} \theta^*_{\alpha}$$

Since λ must correspond to a set of mean parameters valid for each node and edge, it is restricted to the polytope $\mathbb{L}(\mathcal{G})$ defined in equation (7.13). The cost function is strictly convex and the constraints are linear, so that strong duality holds [20]; therefore, the optimum dual value $\mathcal{Q}^*(\bar{\mu}; \theta^*) = \max_{\lambda \in \mathbb{L}(\mathcal{G})} \mathcal{Q}(\lambda; \bar{\mu}; \theta^*)$ is equivalent to the global minimum of the primal problem (7.10). □

■ 7.2.2 Characterization of optimal points

In this section, we characterize both the optimizing argument $\widehat{\lambda}$ of the dual problem in Proposition 7.2.1, as well as the corresponding optimum $\widehat{\boldsymbol{\theta}}$ of the original primal problem (7.10). We begin by showing that for finite θ^*, the optimal $\widehat{\lambda}$ always occurs at interior points of the constraint set $\mathbb{L}(\mathcal{G})$. Next, we provide an explicit construction of these optima, specified in terms of θ^*, and the edge appearance probabilities $\bar{\mu}_e$ associated with the distribution $\bar{\mu}$.

Lemma 7.2.1. If $\|\theta^*\| < \infty$, the optimum $\widehat{\lambda}$ of problem (7.17) is always attained at an interior point of $\mathbb{L}(\mathcal{G})$.

Proof. Consider the dual function of equation (7.18) as a function of λ and θ^*. Observe that the function $-\mathbb{E}_{\bar{\mu}}[\Psi(\Pi^{\mathcal{T}}(\lambda))]$ is strictly concave. Therefore, from the form of equation (7.18), the optimum $\widehat{\lambda} = \widehat{\lambda}(\theta^*)$ and θ^* can be put into one-to-one correspondence via the invertible and continuously differentiable Legendre transform [151]. The

[1] Strictly speaking, we should write $p(\mathbf{x}; \Lambda^{-1}(\Pi^{\mathcal{T}}(\lambda)))$ to mean $p(\mathbf{x}; \Pi^{\mathcal{T}}(\lambda))$, where Λ^{-1} is the inverse Legendre mapping from mean parameters to exponential parameters. (See Section 2.2 for more details).

domains of $\widehat{\lambda}$ and θ^* are $\mathbb{L}(\mathcal{G})$ and $\mathbb{R}^{N+|\mathcal{E}|}$ respectively. Therefore, extreme points of the polyhedron $\mathbb{L}(\mathcal{G})$ are attained only as $\|\theta^*\| \to \infty$. □

The significance of Lemma 7.2.1 is in allowing us to characterize optima of the dual function in equation (7.18) in terms of ordinary gradient conditions. That is, it obviates the need to consider Lagrange multipliers associated with the constraints defining $\mathbb{L}(\mathcal{G})$ in equation (7.13).

Proposition 7.2.2 (Characterization of optimum). For any $\|\theta^*\| < \infty$, the optimal pair $\widehat{\boldsymbol{\theta}}$ and $\widehat{\lambda}$ are characterized by the relations:

$$\theta^*_s = \mathbb{E}_{\bar{\mu}}[\widehat{\theta}(\mathcal{T})_s] = \log\left[\frac{\widehat{\lambda}_s}{(1-\widehat{\lambda}_s)}\right] + \sum_{t \in \mathcal{N}(s)} \bar{\mu}_{st} \log\left[\frac{(\widehat{\lambda}_s - \widehat{\lambda}_{st})}{(1+\widehat{\lambda}_{st} - \widehat{\lambda}_s - \widehat{\lambda}_t)}\right] \quad (7.24a)$$

$$\theta^*_{st} = \bar{\mu}_{st}\widehat{\theta}_{st} = \bar{\mu}_{st} \log\left[\frac{(\widehat{\lambda}_{st})\,(1+\widehat{\lambda}_{st} - \widehat{\lambda}_s - \widehat{\lambda}_t)}{(\widehat{\lambda}_s - \widehat{\lambda}_{st})(\widehat{\lambda}_t - \widehat{\lambda}_{st})}\right] \quad (7.24b)$$

where $\mathcal{N}(s) = \{\, t \in \mathcal{V} \mid (s,t) \in \mathcal{E} \,\}$ is the set of neighbors of node s in $\mathcal{G}$.

Proof. See Appendix D.1. □

Equations (7.24a) and (7.24b) can be viewed as an alternative and more explicit statement of the fact that $p(\mathbf{x}; \widehat{\theta}(\mathcal{T})) = p(\mathbf{x}; \Pi^{\mathcal{T}}(\widehat{\lambda}))$ for all spanning trees $\mathcal{T} \in \mathfrak{T}$. An important implication of equation (7.24b) is that that the optimal exponential edge parameters $\widehat{\theta}(\mathcal{T})_e$ are equal for all spanning trees $\mathcal{T} \in \mathfrak{T}$. From equation (7.24b) and our assumption that $\bar{\mu}_e > 0$ for all $e \in \mathcal{E}$, this common value is given by $\theta^*_e/\bar{\mu}_e$. As a consequence, the only remaining degree of freedom is in the single node parameters $\widehat{\theta}(\mathcal{T})_s$.

Example 7.2.1 (Single loop). To illustrate Proposition 7.2.2, we return to the single loop of Example 7.1.1, which has four spanning trees in total. Each edge in the graph appears in 3 of these 4 spanning trees. As a result, the edge appearance probabilities under the uniform distribution (i.e., $\mu(\mathcal{T}_i) = 1/4$ for all $i = 1, \ldots, 4$) are given by $\mu_e = 3/4$ for all edges $e \in \mathcal{E}$. From equation (7.24b) and the fact that

$$\theta^* = [0\ 0\ 0\ 0\ 1\ 1\ 1\ 1]^T,$$

the optimal exponential edge parameters are given by $\widehat{\theta}_e = \theta^*_e\,/\,(3/4) = 4/3$ for all edges. We now must choose the single node parameters $\widehat{\theta}(\mathcal{T}_i)_s$ of each spanning tree $\mathcal{T}_i$ to ensure that the mean parameters are all equal as well. Doing so numerically yields optimal solutions of the form:

$$\widehat{\theta}(\mathcal{T}_1) = [-a \;\; -a \;\; +a \;\; +a \;\; (4/3)\;\; (4/3)\;\; (4/3)\;\; 0]$$

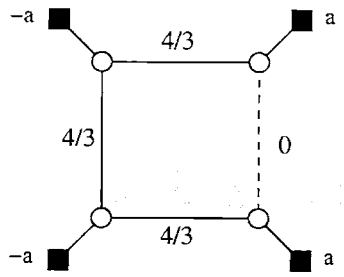


Figure 7.2. Illustration of optimal exponential tree parameter $\widehat{\theta}(\mathcal{T}_1)$ on a single loop. All edges are assigned weight (4/3) and single node parameters are specified in terms of $a \approx 0.5043$. Other optimal solutions $\widehat{\theta}(\mathcal{T}_i)$ are rotated versions of this one.

where $a \approx 0.5043$. Figure 7.2 gives a graphical illustration of the structure of this solution. Other optimal solutions $\widehat{\theta}(\mathcal{T}_i)$, $i = 2, \ldots, 4$ are obtained by rotating $\widehat{\theta}(\mathcal{T}_1)$. With this specification of optimal solutions, it can be verified that

$$\mathbb{E}_{\bar{\mu}}[\widehat{\theta}(\mathcal{T})] = (1/4) \sum_{\mathcal{T}_i} \widehat{\theta}(\mathcal{T}_i) = \theta^*$$

as required.

■ 7.2.3 Decomposition of entropy terms

A major advantage of the dual formulation of Proposition 7.2.1 is the attendant reduction in the dimension of the problem — namely, from the $\mathcal{O}(N\,|\mathfrak{T}(\mathcal{G})|)$ vector $\boldsymbol{\theta}$ to the $(N + |\mathcal{E}|)$-dimensional vector λ. However, a remaining concern with the formulation of Proposition 7.2.1 is the apparent need to calculate an entropy term for all structures $\mathcal{T} \in \text{supp}(\bar{\mu})$. Fortunately, this problem can also be circumvented by using the fact that for a tree-structured distribution, the negative entropy decomposes into a sum of node and edge terms.

In particular, using the form in $p(\mathbf{x}; \Pi^{\mathcal{T}}(\lambda))$ in equation (7.15), we can write

$$\begin{aligned} \Psi(\Pi^{\mathcal{T}}(\lambda)) &= \sum_{\mathbf{x} \in \mathcal{X}^N} p(\mathbf{x}; \Pi^{\mathcal{T}}(\lambda)) \log p(\mathbf{x}; \Pi^{\mathcal{T}}(\lambda)) \\ &= -\sum_{s \in \mathcal{V}} H_s(\lambda) + \sum_{(s,t) \in \mathcal{E}(\mathcal{T})} I_{st}(\lambda) \end{aligned} \tag{7.25}$$

where

$$H_s(\lambda) = -\lambda_s \log \lambda_s - (1 - \lambda_s) \log(1 - \lambda_s) \tag{7.26a}$$

$$\begin{aligned} H_{st}(\lambda) = &-\lambda_{st} \log \lambda_{st} - (1 + \lambda_{st} - \lambda_s - \lambda_t) \log(1 + \lambda_{st} - \lambda_s - \lambda_t) \\ &-(\lambda_s - \lambda_{st}) \log(\lambda_s - \lambda_{st}) - (\lambda_t - \lambda_{st}) \log(\lambda_t - \lambda_{st}) \end{aligned} \tag{7.26b}$$

$$I_{st}(\lambda) = H_s(\lambda) + H_t(\lambda) - H_{st}(\lambda) \tag{7.26c}$$

are the entropy of the single node distribution $p(x_s;\lambda)$, the joint marginal entropy of $p(x_s,x_t;\lambda)$, and the mutual information between x_s and x_t under $p(x_s,x_t;\lambda)$ respectively.

Using the decomposition of equation (7.25), we obtain:

$$\begin{aligned}\mathbb{E}_{\vec{\mu}}[\Psi(\Pi^{\mathcal{T}}(\lambda))] &= \sum_{\mathcal{T}\in\mathfrak{T}}\mu(\mathcal{T})\left\{-\sum_{s\in\mathcal{V}}H_s(\lambda)+\sum_{(s,t)\in\mathcal{E}(\mathcal{T})}I_{st}(\lambda)\right\}\\ &= -\sum_{s\in\mathcal{V}}H_s(\lambda)+\sum_{(s,t)\in\mathcal{E}}\mu_{st}I_{st}(\lambda) \qquad (7.27)\end{aligned}$$

where $\mu_{st} = \Pr_{\vec{\mu}}\{(s,t)\in\mathcal{T}\}$ is the edge appearance probability defined in Definition 7.1.1.

As a consequence, the optimal value of the upper bound depends on the distribution $\vec{\mu}$ *only* via the vector of edge appearance probabilities $\boldsymbol{\mu_e} = \{\mu_e \mid e\in\mathcal{E}\}$. In principle, this result allows us to consider optimizing the choice of distribution $\vec{\mu}$ over all spanning trees by appropriately adjusting the vector $\boldsymbol{\mu_e}$. The potential reduction in complexity is significant, since the vector $\boldsymbol{\mu_e}$ has only $|\mathcal{E}|$ entries, as opposed to the $|\mathfrak{T}(\mathcal{G})|$ entries of $\vec{\mu}$.

■ 7.2.4 Spanning tree polytope

Any procedure for adjusting the elements of $\boldsymbol{\mu_e}$ needs to ensure that they still correspond to the edge appearance probabilities of a valid distribution $\vec{\mu}$ over spanning trees. I.e., they must belong to the set

$$\mathbb{T}(\mathcal{G}) = \{\boldsymbol{\mu_e} \mid \mu_e = \mathbb{E}_{\vec{\mu}}\{\delta[e\in\mathcal{T}]\} \text{ for some } \vec{\mu};\quad \forall\, e\in\mathcal{E}\} \qquad (7.28)$$

where $\delta[e\in\mathcal{T}]$ is an indicator function for edge e to belong to spanning tree $\mathcal{T}$. We use this function to define the *spanning tree incidence* vector $\boldsymbol{\nu}(\mathcal{T})$. For a given spanning tree $\mathcal{T}$, the quantity $\boldsymbol{\nu}(\mathcal{T})$ is a binary-valued vector of length $|\mathcal{E}|$ with elements

$$\boldsymbol{\nu}(\mathcal{T})_e = \delta[e\in\mathcal{T}] \qquad (7.29)$$

With this definition, we can rewrite the equations $\mu_e = \mathbb{E}_{\vec{\mu}}\{\delta[e\in\mathcal{T}]\}$ that define membership in $\mathbb{T}(\mathcal{G})$ in a vector form as follows:

$$\boldsymbol{\mu_e} = \sum_{\mathcal{T}\in\mathfrak{T}}\mu(\mathcal{T})\boldsymbol{\nu}(\mathcal{T}) \qquad (7.30)$$

Equation (7.30) shows that the set $\mathbb{T}(\mathcal{G})$ is the convex hull of the set of spanning tree incidence vectors $\{\boldsymbol{\nu}(\mathcal{T}) \mid \mathcal{T}\in\mathfrak{T}\}$. For this reason, we refer to $\mathbb{T}(\mathcal{G})$ as the *spanning tree polytope*.

By standard results on polyhedra [20], the set $\mathbb{T}(\mathcal{G})$ must have an equivalent characterization in terms of a set of linear inequalities. Fortunately, the spanning tree polytope

is a well-studied object in combinatorial optimization and matroid theory [e.g., 34, 58]. The following lemma, based on a result of Edmonds [58], provides such a characterization of $\mathbb{T}(\mathcal{G})$ in terms of linear relations:

Lemma 7.2.2 (Spanning tree polytope). Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and any subset $A \subseteq \mathcal{E}$, define the rank of A as $r(A) = v(A) - c(A)$, where $v(A)$ is the number of vertices covered by edges in A, and $c(A)$ is the number of connected components of the subgraph induced by A. Then the spanning tree polytope $\mathbb{T}(\mathcal{G})$ is characterized by the following linear relations:

$$\sum_{e \in A} \mu_e \leq r(A) \quad \forall \quad A \subset \mathcal{E} \tag{7.31a}$$

$$\sum_{e \in \mathcal{E}} \mu_e = N - 1 \tag{7.31b}$$

$$\mu_e \geq 0 \quad \forall \quad e \in \mathcal{E} \tag{7.31c}$$

In order to gain intuition for the constraints in equation (7.31), we consider some particular cases. The necessity of the non-negativity constraints in equation (7.31c) is clear, since the μ_e correspond to edge appearance probabilities. The corresponding upper bounds $\mu_e \leq 1$ are obtained by choosing a single edge set $A = \{e\}$. In this case, we have $v(A) = 2$ and $c(A) = 1$, so that $r(A) = 1$. Equation (7.31a) thus reduces to the constraint $\mu_e \leq 1$. Next, equation (7.31b) can be deduced with the following reasoning. Let $\vec{\mu} = \{\mu(\mathcal{T})\}$ be the distribution giving rise to the edge appearance probabilities $\boldsymbol{\mu_e}$. Then

$$\begin{aligned} \sum_{e \in \mathcal{E}} \mu_e &= \sum_{e \in \mathcal{E}} \sum_{\mathcal{T} \in \mathfrak{T}} \mu(\mathcal{T}) \boldsymbol{\nu}(\mathcal{T})_e \\ &= \sum_{\mathcal{T} \in \mathfrak{T}} \mu(\mathcal{T}) \sum_{e \in \mathcal{E}} \boldsymbol{\nu}(\mathcal{T})_e \\ &= N - 1 \end{aligned}$$

where we have used the fact that $\sum_{e \in \mathcal{E}} \boldsymbol{\nu}(\mathcal{T})_e = N - 1$ (since any spanning tree $\mathcal{T}$ on N nodes has $N - 1$ edges); and the fact that $\sum_{\mathcal{T}} \mu(\mathcal{T}) = 1$.

Note that equation (7.31a) captures a large number of linear inequalities — one for each subset A of the edge set. For certain choices of A, such inequalities capture more subtle constraints that arise from the particulars of graph structure. For instance, consider a connected graph in which the edge c is a bridge. I.e. removing the edge c breaks the graph into two components. A simple example of such a graph is illustrated in Figure 7.3. Clearly, any spanning tree $\mathcal{T}$ of $\mathcal{G}$ must include the edge c, which implies that $\mu_c = 1$ for any valid spanning tree distribution over this graph.

This constraint $\mu_c = 1$ is captured by setting $A = \mathcal{E}/c$ in equation (7.31a). With this choice of A, equation (7.31a) indicates that

$$\sum_{e \in \mathcal{E}/c} \mu_e \leq r(\mathcal{E}/c) = N - 2$$

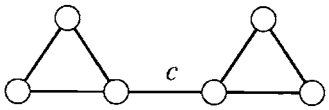


Figure 7.3. A simple example of a graph with a bridge. Removing edge c breaks the graph into two components, and hence must have edge appearance probability $\mu_c = 1$ in any distribution over spanning trees.

At the same time, from equation (7.31b), we also have that $\sum_{e \in \mathcal{E}/c} \mu_e + \mu_c = N - 1$. These two equations, along with $\mu_c \leq 1$, imply that $\mu_c = 1$ as necessary.

Note that there is a constraint of the form in equation (7.31a) for each edge subset $A \subset \mathcal{E}$. Some of these constraints turn out to be redundant; indeed, it suffices to impose bounds of the form in equation (7.31a) only for a certain collection of subsets A of the edge set [142]. To characterize these subsets, recall the following definitions from the background in Section 2.1.1. Given a subset $S \subseteq \mathcal{V}$ of the vertex set, the node-induced subgraph $\mathcal{G}[S]$ is the subgraph of $\mathcal{G}$ induced by S. That is, $\mathcal{G}[S] = (S, \mathcal{E}[S])$ where

$$\mathcal{E}[S] \triangleq \{ (s,t) \in \mathcal{E} \mid s,t \in S \}$$

A *cut vertex* or *cut node* in a graph $\mathcal{G}$ is a member of $\mathcal{V}$ whose removal from $\mathcal{G}$ increases the number of components.

With this terminology, we can now define the relevant collection of edge subsets:

Definition 7.2.1. A subset $A \subset \mathcal{E}$ is a *critical subset* means that:

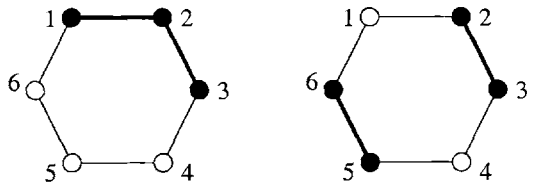
(a) A corresponds to the edge set $\mathcal{E}[S]$ of the induced graph $\mathcal{G}[S]$, for some subset $S \subseteq \mathcal{V}$ of the vertex set, and

(b) the induced graph $\mathcal{G}[S]$ is connected and contain no cut nodes.

Any singleton set $\{e\} \subset \mathcal{E}$ is also critical.

An important result in polyhedral combinatorics [see 142] asserts that a polytope of the form of $\mathbb{T}(\mathcal{G})$ can be characterized using constraints of equation (7.31a) only for critical subsets $A \subseteq \mathcal{E}$. The number of such critical subsets is at most 2^N, corresponding to the edge sets $\mathcal{E}[S]$ of the graphs induced by all 2^N possible subsets S of the vertex set $\mathcal{V}$. Since $|\mathcal{E}| \geq N$ for any connected graph with cycles, this may be a substantial reduction relative to the total number of edge subsets ($2^{|\mathcal{E}|}$). However, it is still an intractable number of constraints in general.

Example 7.2.2. In certain cases, condition (b) of Definition 7.2.1 can lead to a substantial reduction in the number of critical subsets (relative to 2^N). For a single loop, consider any node-induced subgraph $\mathcal{G}[S]$ with $3 \leq |S| < N$. It can be seen such a graph either has a cut node (Figure 7.4(a)), or is not connected (Figure 7.4(b)). Therefore,



(a) (b)

Figure 7.4. For a single loop, each node-induced subgraph $\mathcal{G}[S]$ with $3 \leq |S| < N$ either has cut node, or is not connected. (a) With $S = \{1, 2, 3\}$, node 2 is a cut node in the induced subgraph $\mathcal{G}[S]$. (b) With $S = \{2, 3, 5, 6\}$, the induced subgraph $\mathcal{G}[S]$ is not connected.

the only critical subsets for a single loop are the singleton edge sets $\{e\}$, and the full edge set $\mathcal{E}$. Consequently, the conditions $0 \leq \mu_e \leq 1$ and $\sum_{e \in \mathcal{E}} \mu_e = N - 1$ are sufficient to characterize $\mathbb{T}(\mathcal{G})$ for a single loop.

■ 7.3 Jointly optimal upper bounds

In this section, we begin by specifying the form of the optimal upper bounds on the log partition function $\Phi(\theta^*)$, where the optimization takes place both over the dual variables λ and the set of edge appearance probabilities $\boldsymbol{\mu_e}$. We then turn to characterization of the optimizing arguments $(\widehat{\lambda}, \widehat{\boldsymbol{\mu}_e})$. Finally, we point out some connections between the cost function central to our bounds, and the Bethe free energy of statistical physics [180].

■ 7.3.1 Optimal upper bounds on $\Phi(\theta^*)$

The key insight of Section 7.2.3 is that the expectation $\mathbb{E}_{\vec{\mu}}[\Psi(\Pi^T(\lambda))]$ depends on $\vec{\mu}$ only via the *edge appearance probabilities* $\boldsymbol{\mu_e}$ (see Definition 7.1.1). Thus, using the decomposition of entropy terms given in equation (7.27), we can express the function $-\mathcal{Q}(\lambda; \vec{\mu}; \theta^*)$ of equation (7.18) as follows:

$$\mathcal{F}(\lambda; \boldsymbol{\mu_e}; \theta^*) = -\sum_{s \in \mathcal{V}} H_s(\lambda) + \sum_{(s,t) \in \mathcal{E}} \mu_{st} I_{st}(\lambda) - \sum_{\alpha} \lambda_\alpha \theta^*_\alpha \tag{7.32}$$

where the entropy H_s and mutual information I_{st} are defined in equations (7.26a) and (7.26c) respectively. All of our upper bounds will be expressed in terms of this function evaluated at particular values of $\lambda \in \mathbb{L}(\mathcal{G})$ (as defined in equation (7.13)) and $\boldsymbol{\mu_e} \in \mathbb{T}(\mathcal{G})$ (as defined in equation (7.28)).

Theorem 7.3.1 (Optimal upper bounds).

(a) For an arbitrary $\bar{\boldsymbol{\mu}}_{\boldsymbol{e}} \in \mathbb{T}(\mathcal{G})$, the log partition function is bounded above as follows:

$$\Phi(\theta^*) \leq -\min_{\lambda \in \mathbb{L}(\mathcal{G})} \left\{ \mathcal{F}(\lambda; \bar{\boldsymbol{\mu}}_{\boldsymbol{e}}; \theta^*) \right\} \tag{7.33}$$

This minimum is attained at a unique $\widehat{\lambda} \equiv \widehat{\lambda}(\bar{\mu}_e) \in \mathbb{L}(\mathcal{G})$.

(b) In addition, we have an upper bound, jointly optimal over both λ and μ_e, of the form:

$$\Phi(\theta^*) \leq -\max_{\mu_e \in \mathbb{T}(\mathcal{G})} \mathcal{H}(\mu_e; \theta^*) \tag{7.34}$$

where

$$\mathcal{H}(\mu_e; \theta^*) \triangleq \min_{\lambda \in \mathbb{L}(\mathcal{G})} \{\mathcal{F}(\lambda; \mu_e; \theta^*)\} \tag{7.35}$$

Moreover, there is a unique pair $(\widehat{\lambda}(\widehat{\mu_e}); \widehat{\mu_e}) \in \mathbb{L}(\mathcal{G}) \times \mathbb{T}(\mathcal{G})$ that attains this tightest possible upper bound of equation (7.34).

Proof. (a) For each $\mu_e \in \mathbb{T}(\mathcal{G})$, there exists a corresponding distribution $\vec{\mu}$ that realizes the edge appearance probabilities μ_e. Therefore, the function $\mathcal{F}(\lambda; \mu_e; \theta^*)$ is equivalent to a function of the form $-\mathcal{Q}(\lambda; \vec{\mu}; \theta^*) = \mathbb{E}_{\vec{\mu}}[\Psi(\Pi^{\mathcal{T}}(\lambda))] - \sum_\alpha \lambda_\alpha \theta^*_\alpha$ for some distribution $\vec{\mu}$ over spanning trees $\mathcal{T} \in \mathfrak{T}$. Based on this relation, the upper bound of equation (7.33) follows from Proposition 7.2.1. The negative entropy $\Psi(\Pi^{\mathcal{T}}(\lambda))$ is strictly concave as a function of λ, so that $\mathcal{F}(\lambda; \mu_e; \theta^*) = -\mathcal{Q}(\lambda; \vec{\mu}; \theta^*)$ is strictly convex as a function of λ. Consequently, the associated minimization problem (with linear constraints on λ) has a unique global minimum $\widehat{\lambda}(\mu_e)$.

(b) The bound of equation (7.33) holds for all $\mu_e \in \mathbb{T}(\mathcal{G})$, from which equation (7.34) follows. Observe that $\mathcal{F}(\lambda; \mu_e; \theta^*)$ is linear in μ_e. Therefore, $\mathcal{H}(\mu_e; \theta^*)$ is the minimum of a collection of linear functions, and so is concave as a function of μ_e [20]. Consequently, $\mathcal{H}(\mu_e; \theta^*)$ has a unique global maximum $\widehat{\mu_e}$, at which the optimal value of the upper bound in equation (7.34) is attained. The corresponding optimal λ given by $\widehat{\lambda} \equiv \widehat{\lambda}(\widehat{\mu_e})$. □

We illustrate Theorem 7.3.1 by following up the single loop case of Example 7.2.1.

Example 7.3.1. Consider a single loop on four nodes, as shown in Figure 7.5(a).

(a) We begin with the choice of exponential parameter

$$\theta^* = [0\ 0\ 0\ 0\ 1\ 1\ 1\ 1]^T,$$

as in Example 7.2.1, and fix uniform edge appearance probabilities $\mu_e = (3/4)$ for all edges $e \in \mathcal{E}$. The optimal mean parameter for the bound of Theorem 7.3.1(a) can be calculated as

$$\widehat{\lambda}(\mu_e) = [b_1\ b_1\ b_1\ b_1 \quad b_2\ b_2\ b_2\ b_2]^T,$$

where $b_1 \approx 0.8195$ and $b_2 \approx 0.7069$. This yields the optimal upper bound of $-\mathcal{F}(\widehat{\lambda}; 3/4; \theta^*) \approx 4.6422$ on the true log partition function $\Phi(\theta^*) \approx 4.6252$.

By symmetry of the problem, it can be inferred that the uniform choice of edge appearance probabilities is indeed optimal in the sense of Theorem 7.3.1(b).

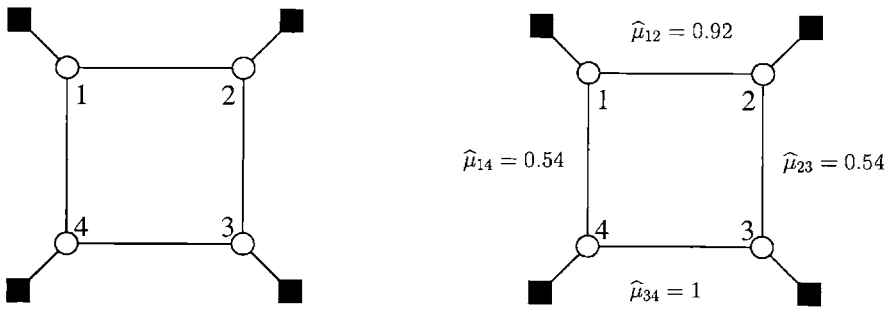


Figure 7.5. Illustration of optimality conditions on a single loop. (a) Single loop. (b) Optimal edge appearance probabilities for $\theta^* = [0\ 0\ 0\ 0\ 1\ 1\ 1\ 3]^T$.

(b) Now consider the same single loop, but the non-symmetric choice of exponential parameter

$$\theta^* = [0\ 0\ 0\ 0\ 1\ 1\ 1\ 3]^T,$$

If we choose uniform (3/4) edge appearance probabilities, then we obtain an upper bound $-\mathcal{F}(\widehat{\lambda}; 3/4; \theta^*) \approx 6.3451$, optimal in the sense of Theorem 7.3.1(a), on the log partition function $\Phi(\theta^*) \approx 6.3326$.

Given the inhomogeneous nature of θ^*, it is appropriate to consider joint optimization over both λ and $\boldsymbol{\mu}_e$, as dictated by Theorem 7.3.1(b). Performing this optimization using Algorithm 7.4.2 of Section 7.4, we obtain the following optimal edge appearance probabilities:

$$\widehat{\boldsymbol{\mu}_e} \approx [0.92 \quad 0.54 \quad 0.54 \quad 1] \tag{7.36}$$

Note that the optimum assigns edge appearance probability of one to the edge with largest weight (i.e., the single edge with weight 3). As a result, this edge must appear in any spanning tree in the support of the optimizing distribution $\widehat{\boldsymbol{\mu}}$. This set of edge appearance probabilities, combined with the associated $\widehat{\lambda}(\widehat{\boldsymbol{\mu}_e})$, yields the upper bound $\mathcal{H}(\widehat{\boldsymbol{\mu}_e}; \theta^*) \approx 6.3387$ on the true log partition function $\Phi(\theta^*) \approx 6.3326$. This upper bound is tighter than the previous bound (≈ 6.3451) based on uniform edge appearance probabilities.

■ 7.3.2 Alternative proof

In this section, we present an alternative proof of part (a) of Theorem 7.3.1. It provides a perspective different from that of convex combinations in the exponential domain.

We begin with the variational representation of $\Phi(\theta^*)$, as guaranteed by the Legendre transform (see Section 2.2.4):

$$\Phi(\theta^*) = \max_{\eta \in \mathbb{M}(\mathcal{G})} \left\{ \eta^T \theta^* - \Psi(\eta) \right\} \tag{7.37}$$

where $\mathbb{M}(\mathcal{G})$ is the set of valid mean parameters. (I.e., $\mathbb{M}(\mathcal{G})$ is given by the range $Ra(\Lambda)$ of the Legendre mapping).

By the convexity of the negative entropy Ψ, for any tree-structured set of mean parameters η^{tree}, we have

$$\Psi(\eta) \;\geq\; \Psi(\eta^{\text{tree}}) + \sum_{\alpha} \theta_{\alpha}^{\text{tree}} (\eta_{\alpha} - \eta_{\alpha}^{\text{tree}}) \tag{7.38}$$

Here we have used the fact that $\frac{\partial \Psi}{\partial \eta_{\alpha}}(\eta^{\text{tree}}) = \theta_{\alpha}^{\text{tree}}$. For a fixed tree $\mathcal{T}$ specified by edge set $\mathcal{E}(\mathcal{T}) \subset \mathcal{E}$, the lower bound of equation (7.38) is tightest, in fact, for the *moment-matched* tree distribution

$$\eta^{\text{tree}} = \Pi^{\mathcal{T}}(\eta) = \{\eta_s, \;\; \eta_{st} \;\mid\; s \in \mathcal{V}; \quad (s,t) \in \mathcal{E}(\mathcal{T}) \} \tag{7.39}$$

This fact is most easily seen by noting that the difference between the LHS and RHS of equation (7.38) is equivalent to the KL divergence $D(\eta \parallel \eta^{\text{tree}})$ between the distributions $p(\mathbf{x}; \eta)$ and $p(\mathbf{x}; \eta^{\text{tree}})$. (See equation (2.32) for this dual representation of the KL divergence). Therefore, the problem of maximizing the lower bound on the RHS of equation (7.38) is equivalent to minimizing this KL divergence, which corresponds to an I-projection onto the e-flat manifold of tree-structured distributions. (See Section 2.2.7 for details on these notions from information geometry). Therefore, the optimal tree parameter is given by the moment matching procedure specified in equation (7.39).

For this choice of η^{tree}, equation (7.38) reduces to the simpler form

$$\Psi(\eta) \;\geq\; \Psi(\Pi^{\mathcal{T}}(\eta)) \tag{7.40}$$

since the mean parameters of η and η^{tree} are equal for indices corresponding to single nodes or edges in the tree, and $\theta_{\alpha}^{\text{tree}} = 0$ for all other indices (corresponding to edges not in the tree). This is a statement of the fact that any distribution on a graph has lower[2] entropy than a moment-matched distribution structured according to any of its spanning trees.

Since equation (7.40) holds for any spanning tree $\mathcal{T}$, we can consider taking a convex combination of such bounds, each weighted by some $\mu(\mathcal{T}) \geq 0$. This yields the weighted lower bound:

$$\Psi(\eta) \;\geq\; \sum_{\mathcal{T}} \mu(\mathcal{T}) \Psi(\Pi^{\mathcal{T}}(\eta)) \;=\; \mathbb{E}_{\vec{\mu}}[\Psi(\Pi^{\mathcal{T}}(\eta))] \tag{7.41}$$

[2]Remember that Ψ is *negative* entropy.

Finally, applying the bound of equation (7.41) to the original variational formulation of equation (7.37) yields:

$$\begin{aligned}\Phi(\theta^*) &\leq \max_{\eta \in \mathbb{M}(\mathcal{G})} \{\eta^T\theta^* - \mathbb{E}_{\vec{\mu}}[\Psi(\Pi^T(\eta))]\} && \text{(7.42a)}\\ &\leq \max_{\eta \in \mathbb{L}(\mathcal{G})} \{\eta^T\theta^* - \mathbb{E}_{\vec{\mu}}[\Psi(\Pi^T(\eta))]\} && \text{(7.42b)}\\ &= -\min_{\eta \in \mathbb{L}(\mathcal{G})} \mathcal{F}(\eta; \boldsymbol{\mu}_e; \theta^*) && \text{(7.42c)}\end{aligned}$$

where equation (7.42a) follows from the bound of equation (7.41); and equation (7.42b) follows because the set of tree-consistent mean parameters $\mathbb{L}(\mathcal{G})$ is a superset of the set $\mathbb{M}(\mathcal{G})$ of globally $\mathcal{G}$-consistent mean parameters. Finally, equation (7.42c) is obtained in a straightforward manner by decomposing the negative entropies into node and edge terms, and using the definition of $\mathcal{F}$ in equation (7.32). This final equation (7.42c) is equivalent to equation (7.33) in the statement of Theorem 7.3.1(a).

This alternative proof shows more directly why moment-matched tree distributions arise in the optimal form of the bounds. However, it does not make clear the links to our original starting point — namely, that of taking convex combinations of exponential parameters. From this alternative derivative, it is easily seen how to generalize the bounds to distributions defined by subgraphs of higher treewidth. Indeed, a lower bound analogous to that of equation (7.38) can be derived for any triangulated subgraph of the original graph. A similar argument will establish that moment-matching again yields the optimal form of the bound, as in equation (7.40). Finally, we can use a weighted combination of negative entropies, thereby yielding upper bounds analogous to that of equation (7.42).

■ 7.3.3 Characterization of joint optima

In this section, we provide a number of results characterizing the joint optima $(\widehat{\lambda}(\widehat{\boldsymbol{\mu}_e}), \widehat{\boldsymbol{\mu}_e})$ of Theorem 7.3.1(b). In particular, we show that these optima can be characterized in terms of a balancing of mutual information terms $I_{st}(\widehat{\lambda}(\widehat{\boldsymbol{\mu}_e}))$ on each edge (s,t) of the graph. Moreover, we develop a geometric relation between this mutual information vector, and spanning tree incidence vectors $\boldsymbol{\nu}(\mathcal{T})$. Finally, these results lead to a minimax result that has a game-theoretic flavor [170].

Our first result is a characterization of the mutual information terms $I_{st}(\widehat{\lambda}(\widehat{\boldsymbol{\mu}_e}))$, which arise from the optimal pair $(\widehat{\lambda}(\widehat{\boldsymbol{\mu}_e}), \widehat{\boldsymbol{\mu}_e})$.

Proposition 7.3.1 (Characterization of joint optima). Let I_{st} be the mutual information as defined in equation (7.26c).

(a) There exist real numbers $\xi_0 > 0$ and $\xi(A) \geq 0$ such that for each $(s,t) \in \mathcal{E}$, the optimum $(\widehat{\lambda}(\widehat{\boldsymbol{\mu}_e}), \widehat{\boldsymbol{\mu}_e})$ is characterized by the following conditions:

$$I_{st}(\widehat{\lambda}(\widehat{\boldsymbol{\mu}_e})) = \xi_0 + \sum_{A \ni (s,t)} \xi(A) \qquad \text{(7.43)}$$

The sum $\sum_{A \ni (s,t)}$ ranges over all critical subsets A that include edge (s,t). Moreover, for each critical subset A, we have $\xi(A) > 0$ only if the constraint $\sum_{e \in A} \mu_e \leq r(A)$ of equation (7.31a) is met with equality.

(b) Letting $\langle\ ,\ \rangle$ denote the Euclidean inner product, we have:

$$\langle I(\widehat{\lambda}(\widehat{\boldsymbol{\mu}_e})),\ \boldsymbol{\nu}(\mathcal{T}) - \widehat{\boldsymbol{\mu}_e} \rangle \leq 0 \qquad \forall \quad \mathcal{T} \in \mathfrak{T} \tag{7.44}$$

for the incidence vector $\boldsymbol{\nu}(\mathcal{T})$ of any spanning tree $\mathcal{T} \in \mathfrak{T}$.

Since $\widehat{\boldsymbol{\mu}_e} \in \mathbb{T}(\mathcal{G})$, there exists a distribution over spanning trees $\widehat{\boldsymbol{\mu}}$ that gives rise to this set of edge appearance probabilities. Inequality (7.44) holds with equality for all spanning trees $\mathcal{T}$ in the support of $\widehat{\boldsymbol{\mu}}$ (i.e., spanning trees for which $\widehat{\mu}(\mathcal{T}) > 0$).

Proof. See Appendix D.2. □

Equation (7.43) of Proposition 7.3.1(a) specifies that at the optimum $(\widehat{\lambda}, \widehat{\boldsymbol{\mu}_e})$, the mutual information on each edge (s,t) of the graph (i.e., between the random variables x_s and x_t) is balanced in a certain sense. The mutual information on edge (s,t) consists of a baseline amount $\xi_0 \geq 0$, to which we add varying amounts of additional information (i.e., $\sum_{A \ni (s,t)} \xi(A) \geq 0$), depending on how many critical sets A corresponding to an active constraint involve edge (s,t).

Example 7.3.2 (Optimal information terms for a single loop). The conditions of Proposition 7.3.1(a) take a particularly simple form for a single loop, where the only critical subsets A are formed of single edges. In this case, we have

$$I_{st}(\widehat{\lambda}(\widehat{\boldsymbol{\mu}_e})) = \begin{cases} \xi_0 & \text{if } \widehat{\mu}_{st} < 1 \\ \xi_0 + \xi[(s,t)] & \text{if } \widehat{\mu}_{st} = 1 \end{cases} \tag{7.45}$$

Thus, the Lagrangian conditions correspond to an equalization of mutual information on edges. The mutual information is equal to a constant for all edges (s,t) with appearance probability $\widehat{\mu}_{st} < 1$; the mutual information on any edge (s,t) with appearance probability $\widehat{\mu}_{st} = 1$ is boosted by some quantity $\xi[(s,t)] \geq 0$.

To follow up Example 7.3.1(b), the optimal information terms for this problem (with the same ordering of edges as in equation (7.36)) are given by:

$$I_e(\widehat{\lambda}(\widehat{\boldsymbol{\mu}_e})) = [\xi_0 \quad \xi_0 \quad \xi_0 \quad \xi_0 + \xi_{\{34\}}]$$

where $\xi_0 \approx 0.012$ and $\xi_{\{34\}} \approx 0.007$.

The geometric interpretation of Proposition 7.3.1(b) is interesting, as illustrated in Figure 7.6. Each spanning tree incidence vector $\boldsymbol{\nu}(\mathcal{T})$ is an extreme point of the spanning tree polytope $\mathbb{T}(\mathcal{G})$. The inequality (7.44) indicates that the angle between the information vector $I(\widehat{\lambda}(\widehat{\boldsymbol{\mu}_e}))$ and the difference vector $\boldsymbol{\nu}(\mathcal{T}') - \widehat{\boldsymbol{\mu}_e}$ is obtuse for all spanning trees $\mathcal{T}' \in \mathfrak{T}$. Moreover, this angle is orthogonal for all spanning tree incidence vectors $\boldsymbol{\nu}(\mathcal{T})$ in the support of $\widehat{\boldsymbol{\mu}}$ (i.e., trees for which $\widehat{\mu}(\mathcal{T}) > 0$).

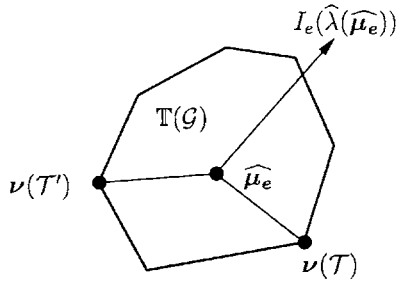


Figure 7.6. Geometry of the optimal edge appearance vector $\widehat{\mu_e}$ in the spanning tree polytope $\mathbb{T}(\mathcal{G})$. Extreme points of this polytope are spanning tree incidence vectors $\nu(\mathcal{T})$. The vector $I_{st}(\widehat{\lambda}(\widehat{\mu_e}))$ forms an obtuse angle with $\nu(\mathcal{T}') - \widehat{\mu_e}$ for all $\mathcal{T}' \in \mathfrak{T}$. It is orthogonal to $\nu(\mathcal{T}) - \widehat{\mu_e}$ whenever $\mathcal{T} \in \text{supp}(\widehat{\mu_e})$.

Proposition 7.3.1(b) also leads to the following result:

Proposition 7.3.2 (Minimax relation).
For all $\|\theta^*\| < \infty$, the following minimax relation holds:

$$\max_{\mu_e \in \mathbb{T}(\mathcal{G})} \min_{\lambda \in \mathbb{L}(\mathcal{G})} \mathcal{F}(\lambda; \mu_e; \theta^*) = \min_{\lambda \in \mathbb{L}(\mathcal{G})} \max_{\mu_e \in \mathbb{T}(\mathcal{G})} \mathcal{F}(\lambda; \mu_e; \theta^*) \tag{7.46}$$

Proof. The function $\mathcal{F}(\lambda; \mu_e; \theta^*)$ is convex in λ and linear (hence concave) in μ_e. Moreover, the constraint sets $\mathbb{L}(\mathcal{G})$ and $\mathbb{T}(\mathcal{G})$ are both convex and compact. Equation (7.46) therefore follows from standard minimax results [61]. □

Proposition 7.3.2 has an interesting game-theoretic interpretation. Imagine a two-person game specified by the payoff function $\mathcal{F}(\lambda; \mu_e; \theta^*)$. The goal of player 1 is to choose $\lambda \in \mathbb{L}(\mathcal{G})$ so as to minimize this function, whereas the goal of player 2 is to choose a spanning tree (or set of spanning trees) so as to maximize this function. Choosing a single spanning tree can be viewed as a *pure strategy* in the game-theoretic sense [170]. Equation (7.46) specifies an equilibrium condition for the optimal solution pair. In contrast to a pure strategy, this equilibrium involves choosing a distribution $\widehat{\vec{\mu}}$ over spanning trees, which gives rise to the optimal edge appearance vector $\widehat{\mu_e}$. Consequently, the optimal strategy for player 2 is not the deterministic choice of a single spanning tree, but rather the *mixed strategy* of randomly choosing a spanning tree from the specified distribution $\widehat{\vec{\mu}}$.

7.3.4 Relation to Bethe free energy

Recall the cost function that (when optimized) gives rise to bounds on the log partition function:

$$\mathcal{F}(\lambda; \boldsymbol{\mu}_e; \theta^*) = -\sum_{s \in \mathcal{V}} H_s(\lambda) + \sum_{(s,t) \in \mathcal{E}} \mu_{st} I_{st}(\lambda) - \sum_{\alpha} \lambda_\alpha \theta^*_\alpha \tag{7.47}$$

The first two terms are a sum of (negative) entropy terms at each node, and a sum of (weighted) mutual information terms for each pair of random variables joined by an edge. Borrowing terminology from statistical physics [135], the final term $\sum_\alpha \lambda_\alpha \theta^*_\alpha$ can be viewed as an "average energy".

On this basis, it can be seen that the function of equation (7.47) is very closely related to the Bethe free energy [180]. (For instance, compare equation (7.47) to equation (6.17) of Chapter 6.) In fact, the Bethe free energy is a special case of equation (7.47), in which all the edge appearance probabilities μ_e are set equal to 1. This particular choice of $\boldsymbol{\mu}_e$ does *not* belong to the spanning tree polytope $\mathbb{T}(\mathcal{G})$, unless of course $\mathcal{G}$ is tree-structured. Therefore, our analysis does not guarantee that the Bethe free energy is convex; indeed, the Bethe free energy fails to be convex for many graphs with cycles, which leads to failures of convergence and multiple local minima.

Nonetheless, this link is interesting. As shown by Yedidia et al. [180], belief propagation (BP) can be viewed as attempting to perform a constrained minimization of the Bethe free energy. The minimizing arguments are taken as approximations to the actual marginals of the original distribution. It is not surprising, then, that the optimality conditions of our variational formulation (see Proposition 7.2.2) are very closely related to the optimality conditions of tree-based reparameterization or BP (see Theorem 5.4.1 of Chapter 5). Overall, the analysis of this chapter is likely to have interesting implications for approximate inference. We shall discuss these possibilities at more depth in Chapter 8.

7.4 Algorithms and simulation results

In this section, we develop algorithms for carrying out the optimizations specified in Theorem 7.3.1. We then present the results of applying these algorithms to compute upper bounds on the log partition function of randomly specified problems.

7.4.1 Inner minimization over λ

We begin by describing an algorithm for carrying out the inner minimization step (i.e., computing $\min_{\lambda \in \mathbb{L}(\mathcal{G})} \{\mathcal{F}(\lambda; \bar{\boldsymbol{\mu}}_e; \theta^*)\}$) required to evaluate the upper bound of Theorem 7.3.1(a). The function $\mathcal{F}$ is strictly convex in λ, and the constraint set $\mathbb{L}(\mathcal{G})$ is formed by a set of $\mathcal{O}(N + |\mathcal{E}|)$ linear constraints (see equation (7.13)). Therefore, the problem is a suitable candidate for *constrained Newton's method* [20], wherein we take Newton steps projected back onto the constraint set.

The steps involved are given in Algorithm 7.4.1. Computing both the gradient $\nabla\mathcal{F}(\lambda^n; \bar{\mu}_e; \theta^*)$ and Hessian $\nabla^2\mathcal{F}(\lambda^n; \bar{\mu}_e; \theta^*)$ are straightforward tasks. Indeed, since $\mathcal{F}$ decouples into a sum of node and edge terms, the Hessian has a structure that reflects the edge structure of $\mathcal{G}$. (Hence, for sparse graphs, the Hessian will also be sparse). The computation of the descent direction $\widetilde{\lambda}^{n+1}$ in step 2 of Algorithm 7.4.1 is a quadratic program (i.e., minimizing a quadratic function subject to linear constraints); and can be solved efficiently. With suitable choice of step sizes α^n (e.g., via the Armijo rule, or limited minimization rule [see 20]), this algorithm is guaranteed to converge to the unique global minimum $\widehat{\lambda}$. The convergence of Algorithm 7.4.1 is guaranteed to be superlinear in a neighborhood of the optimum with unity step size [20].

Algorithm 7.4.1 (Constrained Newton's method).

1. Initialize $\lambda^0 \in \mathbb{L}(\mathcal{G})$.
2. For iterations $n = 0, 1, 2, \ldots$, compute the descent direction:

$$\widetilde{\lambda}^{n+1} = \arg\min_{\lambda \in \mathbb{L}(\mathcal{G})} \left\{ \nabla\mathcal{F}(\lambda^n; \bar{\mu}_e; \theta^*)'(\lambda - \lambda^n) + \frac{1}{2}(\lambda - \lambda^n)\nabla^2\mathcal{F}(\lambda^n; \bar{\mu}_e; \theta^*)(\lambda - \lambda^n) \right\}$$

3. Form the new iterate $\lambda^{n+1} = (1 - \alpha^n)\lambda^n + \alpha^n\widetilde{\lambda}^{n+1}$, where $\alpha^n \in (0, 1]$ is a step size parameter.

■ 7.4.2 Outer maximization over μ_e

We now consider the maximization $\max_{\mu_e \in \mathbb{T}(\mathcal{G})} \mathcal{H}(\mu_e; \theta^*)$ required to compute the upper bound of equation (7.34) in Theorem 7.3.1. Neither the Hessian nor the gradient of $\mathcal{H}$ are difficult to compute. It is therefore tempting to apply a constrained Newton's method once again. However, recall from Lemma 7.2.2 that the spanning tree polytope $\mathbb{T}(\mathcal{G})$ is defined by a very large number ($\mathcal{O}(2^N)$) of linear inequalities. For this reason, solving a constrained quadratic program over $\mathbb{T}(\mathcal{G})$ (as in step 2 of Algorithm 7.4.1) is intractable for large enough graphs.

Fortunately, despite the exponential number of constraints characterizing $\mathbb{T}(\mathcal{G})$, optimizing a linear function subject to the constraint $\mu_e \in \mathbb{T}(\mathcal{G})$ turns out to be straightforward. Two observations are key. First of all, from standard results on linear programming [20], the optimal value of a feasible linear program is always attained at an extreme point of the linear polyhedron formed by the constraints.[3] Secondly, extreme

[3]The optimal value may be attained at more than one extreme point, or at interior points as well.

points of the spanning tree polytope $\mathbb{T}(\mathcal{G})$ are given by spanning tree incidence vectors $\boldsymbol{\nu}(\mathcal{T})$, as defined in equation (7.29) [58].

Algorithm 7.4.2 (Conditional gradient).

1. Initialize $\boldsymbol{\mu_e}^0 \in \mathbb{T}(\mathcal{G})$.
2. For iterations $n = 0, 1, 2, \ldots$, compute the ascent direction as follows:

$$\widetilde{\boldsymbol{\mu_e}}^{n+1} = \arg \max_{\boldsymbol{\mu_e} \in \mathbb{T}(\mathcal{G})} \left\{ \left\langle \nabla \mathcal{H}(\boldsymbol{\mu_e}^n; \theta^*),\ (\boldsymbol{\mu_e} - \boldsymbol{\mu_e}^n) \right\rangle \right\} \tag{7.48}$$

3. Form the new iterate $\boldsymbol{\mu_e}^{n+1} = (1 - \alpha^n)\boldsymbol{\mu_e}^n + \alpha^n \widetilde{\boldsymbol{\mu_e}}^{n+1}$, where $\alpha^n \in (0, 1]$ is a step size parameter.

As a consequence, maximizing a linear function over the spanning tree polytope is equivalent to solving a *maximum weight spanning tree* problem [see 107]. Using these facts, it can be seen that the conditional gradient method [20], as specified in Algorithm 7.4.2, is a computationally feasible proposal.

It is helpful to consider the steps of Algorithm 7.4.2 in more detail. Due to the exponential number of constraints defining $\mathbb{T}(\mathcal{G})$, even the first step — that of assessing whether a given vector belongs to $\mathbb{T}(\mathcal{G})$ — is non-trivial. For instance, the uniform assignment $\mu_e = (N-1)/|\mathcal{E}|$ need not belong to $\mathbb{T}(\mathcal{G})$. (Consider the graph of Figure 7.3). If we are given a distribution $\vec{\boldsymbol{\mu}}$ with a limited support, it is possible to compute the expectations that define the edge appearance probabilities $\boldsymbol{\mu_e}$ (see equation (7.3)) by direct summation. More generally, it turns out to be useful to consider a particular class of distributions over spanning trees, defined by:

$$\vec{\boldsymbol{\mu}}(\mathcal{T}; W) \propto \prod_{e \in \mathcal{T}} W_e \tag{7.49}$$

where $W_e \geq 0$ is a weight assigned to each edge $e \in \mathcal{E}$. That is, the probability of a given spanning tree $\mathcal{T}$ is proportional to the product of the weights on all its edges. For a distribution in this class, Jaakkola et al. [97] showed how a weighted variant of the matrix-tree theorem [22, 168] could be used to compute expectations under $\vec{\boldsymbol{\mu}}(\mathcal{T}; W)$. This method can be used to compute a feasible starting point $\boldsymbol{\mu_e}^0 \in \mathbb{T}(\mathcal{G})$.

In the second step of Algorithm 7.4.2, given a fixed $\boldsymbol{\mu_e}^n$, we first solve a problem of the form in Theorem 7.3.1(a) using Algorithm 7.4.1 to obtain the optimal $\widehat{\lambda}(\boldsymbol{\mu_e}^n)$. Having obtained this optimal point, computing the ascent direction of equation (7.48) is equivalent to solving a maximum weight spanning tree problem. It can be shown (see Appendix D.2) that

$$\frac{\partial \mathcal{H}}{\partial \mu_e}(\boldsymbol{\mu_e}^n; \theta^*) = I_e(\widehat{\lambda}(\boldsymbol{\mu_e}^n)) \tag{7.50}$$

so that the edge weights in the maximum spanning tree problem are mutual information terms. This computation can be performed efficiently using Kruskal's algorithm [116]; see also [107]. In the third step, the step size $\alpha^n \in (0, 1]$ can be chosen via the Armijo rule. (See Bertsekas [20] for details on this and other stepsize selection rules.)

Given equation (7.50), the second and third steps have an interesting interpretation. In particular, let us view the vector of mean parameters $\widehat{\lambda}(\boldsymbol{\mu}_e{}^n)$ as a set of data, which is used to specify mutual information terms $I_e(\widehat{\lambda}(\boldsymbol{\mu}_e{}^n))$ on each edge. In this case, finding the corresponding maximum weight spanning tree is equivalent to finding the tree distribution that best fits the data in the maximum likelihood sense (or KL divergence between the empirical distribution specified by the data, and the tree distribution). See Chow and Liu [36] for more details on this interpretation of the maximum weight spanning tree procedure. Therefore, at each iteration, the algorithm moves towards the spanning tree that best fits the current data.[4]

■ 7.4.3 Empirical simulations

In this section, we present the results of applying the previously described algorithms to compute the upper bounds specified in Theorem 7.3.1. We performed simulations for a binary-valued vector $\mathbf{x}$ (taking values in $\{0, 1\}$) for two different types of graphs (square grids and a fully connected graph) under two different types of interactions (attractive or mixed potentials). For the purposes of comparison, we also calculated lower bounds using the naive mean field approximation. See Section 2.3.1 for details on mean field.

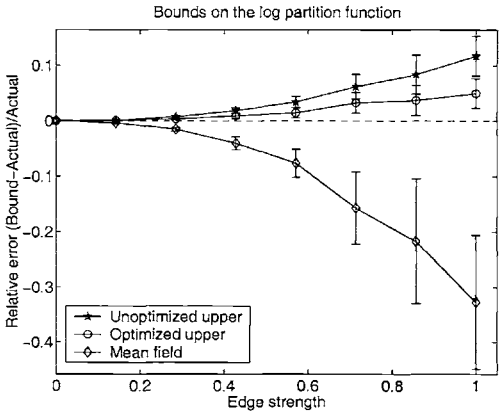
For each trial on a given graph, we defined a distribution $p(\mathbf{x}; \theta^*)$ by randomly choosing an exponential parameter vector θ^* from either the uniform attractive ensemble or the uniform mixed ensemble. See Section 2.2.1 for the definitions of these ensembles of distributions.

Grids of varying sizes

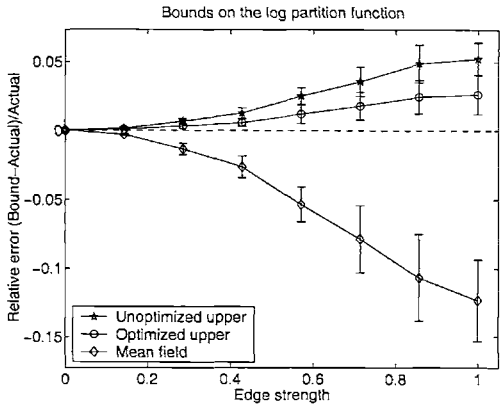
We first performed simulations for square 2-D grids of varying sizes; the number of nodes N was either 9, 36, or 81. For each of these grid sizes and each of the two conditions (attractive or mixed), we ran simulations with edge strengths d ranging[5] from 0 to $\frac{4}{\sqrt{N}}$.For each setting of the edge strength, we performed 30 trials for the $N = 9$ grids, and 10 trials for $N = 36$ or 81. The inner minimization $\min_{\lambda \in \mathbb{L}(\mathcal{G})} \mathcal{F}(\lambda; \boldsymbol{\mu}_e; \theta^*)$ was performed using the constrained Newton's method (Algorithm 7.4.1), whereas the outer maximization was performed with the conditional gradient method (Algorithm 7.4.2). In all cases, step size choices were made by the Armijo rule [20]. The value of the actual partition function $\Phi(\theta^*)$ was computed by forming a junction tree for each grid, and

[4]There is one minor caveat with this interpretation — namely, as noted previously, the vector $\widehat{\lambda}(\boldsymbol{\mu}_e{}^n)$ may not correspond to a valid set of marginals for any distribution.

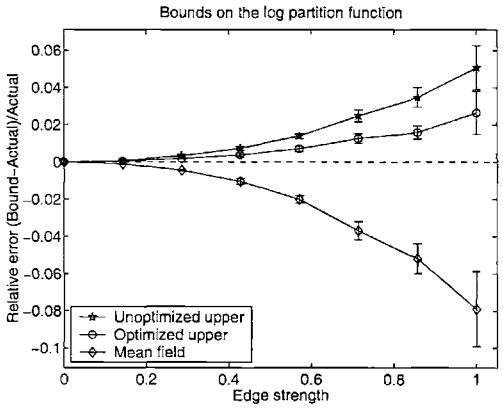
[5]The normalization by $1/\sqrt{N}$ guarantees that the effects impinging on a given node are scale-invariant; they converge to a fixed distribution as the problem size N tends to infinity.



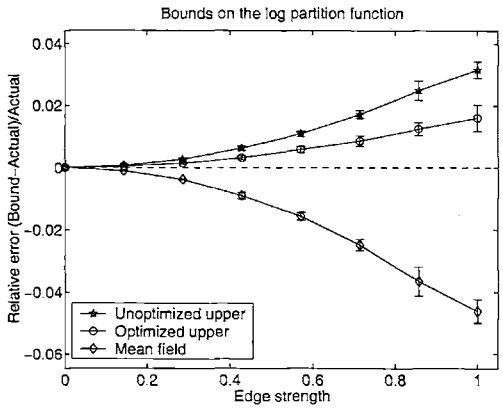
(a) Attractive; $N = 9$



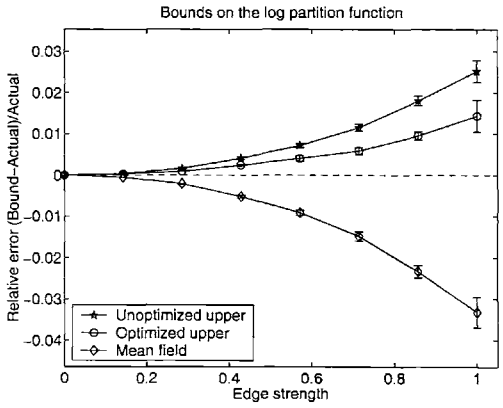
(b) Mixed; $N = 9$



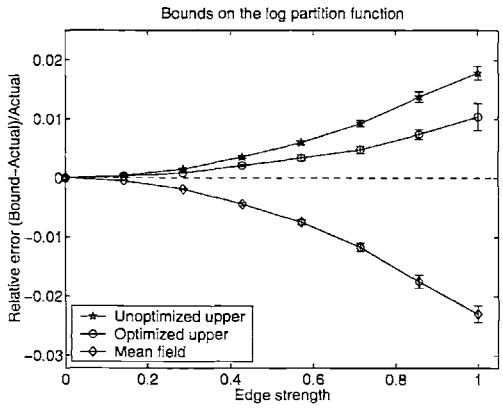
(c) Attractive; $N = 36$



(d) Mixed; $N = 36$



(c) Attractive; $N = 81$



(d) Mixed; $N = 81$

Figure 7.7. Upper and lower bounds on $\Phi(\theta^*)$ for a randomly chosen distribution $p(\mathbf{x}; \theta^*)$ on grids of various sizes: $N = 9$ nodes (first row), $N = 36$ (middle row) or $N = 81$ (bottom row). Panels on the left (respectively right) correspond to the attractive (respectively mixed) condition. Each panel shows the mean relative error $[\text{Bound} - \Phi(\theta^*)]/\Phi(\theta^*)$ versus a normalized measure of edge strength; error bars correspond to plus/minus one standard deviation.

performing exact computations on this junction tree.[6]

Shown in Figure 7.7 are plots of the *relative error* [Bound $-\Phi(\theta^*)]/\Phi(\theta^*)$ versus the edge strength (normalized by $4/\sqrt{N}$ for each N so that it falls in the interval $[0, 1]$). The three rows of this figure corresponds to problem sizes $N = 9$, 36 and 81 respectively. Panels on the left correspond to the attractive condition, whereas those on the right correspond to the mixed condition. Each panel displays the relative error in two types of upper bounds. The "unoptimized" curve shows the bound of Theorem 7.3.1(a) with the fixed choice of uniform edge appearance probabilities $\mu_e = (N-1)/|\mathcal{E}|$. The "optimized" curve corresponds to the jointly optimal (over both λ and $\vec{\mu}$) upper bounds of Theorem 7.3.1(b). The lower curve in each panel corresponds to the relative error in the naive mean field lower bound.

The bounds are tightest for low edge strengths d; their tightness decreases as the edge strength is increased. Optimizing the edge appearance probabilities can lead to significantly better upper bounds. This effect is especially pronounced as the edge strength is increased, in which case the distribution of edge weights θ^*_{st} becomes more inhomogeneous. For these square grids, the tightness of the upper bounds of Theorem 7.3.1 decreases more slowly than the corresponding mean field lower bound. In terms of the relative error plotted here, the upper bounds are superior to the mean field bound by factors of roughly 3 in the attractive case, and roughly 2 in the mixed case. The tightness of the bounds, measured in terms of relative error, decreases slightly as the problem size (number of nodes N) is increased and the edge strengths are rescaled in terms of $1/\sqrt{N}$.

It is worthwhile emphasizing the importance of the dual formulation of our bounds. Indeed, the naive approach of attempting to optimize the primal formulation of the bounds (e.g., see problem (7.10)) would require dealing with a number[7] of spanning trees that ranges from 192 for $N = 9$ nodes, all the way up to the astronomical number $\approx 8.33 \times 10^{33}$ for $N = 81$ nodes.

Fully connected graph

We also performed simulations on the fully connected graph K_9 on $N = 9$ nodes, with edge strengths d ranging from 0 to 4/3. The results are plotted in Figure 7.8. Both types of bounds (upper convex and mean field) are much poorer for this densely connected graph (as compared to the grids). Moreover, in contrast to the grids, there is a striking disparity between the attractive and mixed conditions. In the attractive condition, none of the bounds are good; however, in a relative sense, the optimized upper bounds of Theorem 7.3.1(b) are better than the mean field lower bound. We also note that optimizing the edge appearance probabilities leads to significant improvements; indeed, the unoptimized upper bound is worse than the mean field lower bound. In the mixed condition, the mean field lower bound is of mediocre quality, whereas the upper bounds

[6]Thank you to Yee Whye Teh and Max Welling for generosity in sharing their code for performing exact inference on grids.

[7]These numbers can be calculated by applying the Matrix-Tree theorem [168] to the square grids.

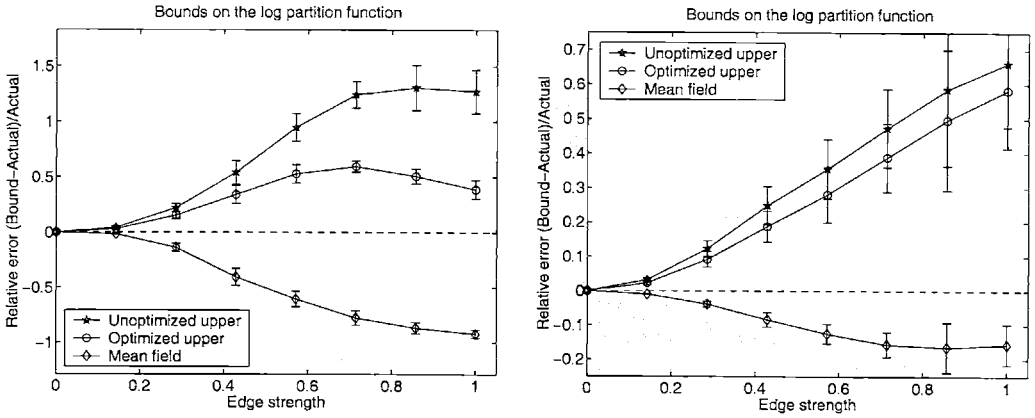


Figure 7.8. Upper and lower bounds on $\Phi(\theta^*)$ for a randomly chosen distribution $p(\mathbf{x};\theta^*)$ on the complete graph K_9. (a) Attractive condition. (b) Mixed condition.

are very poor. Thus, the quality of the bounds in Theorem 7.3.1 appears to degrade for the case of mixed potentials on densely connected graphs. Of course, at least in the limit of large problem sizes, mean field would behave very well for densely connected graphs with mixed potentials, since the aggregation of multiple effects converges to a mean effect [see 92].

■ 7.5 Discussion

In this chapter, we have developed and analyzed a new class of upper bounds for log partition functions in graphical models. These bounds are based on convex combinations of distributions defined on spanning trees of the graph with cycles. We proved that there is a unique distribution over spanning trees and an associated set of exponential parameters that yield the tightest possible upper bound. Despite the prohibitively large number of spanning trees in a general graph, we developed a technique for optimizing the bound efficiently — though implicitly — over all spanning trees.

It would be interesting to compare the quality of the upper bounds of this chapter to the upper bound of Jaakkola and Jordan [94]. In particular, for the Ising model (i.e., a binary process on a graph with pairwise potentials), they developed a recursive procedure, in which the contribution of a given node is bounded above (quadratically), and then eliminated from the graph. As a side effect, all neighbors of the eliminated node are joined together by a modified set of potentials. The procedure then continues by selecting a node from this modified graph. For relatively weak choices of potentials, these bounds are much superior to the (linear) mean field lower bound [94]. However, they are unlikely to behave as well for very strong potentials.

The bounds that we developed in this chapter have a number of potential uses. Of particular relevance to this thesis are their application to computing bounds on the error of the TRP/BP algorithm (Chapter 5), as well as error bounds for more structured approximating algorithms (Chapter 6). More generally, the techniques in this chapter can be used to compute bounds on arbitrary expectations $\mathbb{E}_{\theta^*}[f]$, as described in Chapter 3.

Another potential application of our techniques is in large deviations analysis [e.g., 53,158]. It is well-known that the log partition function plays the role of a rate function: that is, it specifies the exponential rate at which the probability of certain events — namely, large deviations — decays as the number of samples is increased. In cases for which exact computation of these error exponents is infeasible, it would be of interest to obtain bounds. See Chapter 8 for further discussion of this issue.

For clarity in exposition, this chapter focused on the special case of binary-valued nodes, and graphs with pairwise cliques. However, the line of analysis outlined here is broadly applicable, in that it extends naturally to more general state spaces (e.g., $\mathcal{X} = \mathbb{R}$, or the discrete space $\mathcal{X} = \{0, 1, \ldots, m-1\}$) and larger cliques. Analogs of the results given in this chapter hold for these cases. It is also possible to consider more complex approximating structures — e.g., graphs of treewidth $k \geq 2$, as opposed to spanning trees. (See [17, 162] for more background on hypergraphs and the notion of treewidth). One caveat is relevant here: the optimization of the distribution $\vec{\mu}$ would not be as straightforward as with spanning trees. Although solving the maximum weight spanning tree problem required as part of Algorithm 7.4.2 is straightforward, its analog for structures of treewidth $k \geq 2$ is NP-hard [111, 162].

Chapter 8

Contributions and Suggestions

This chapter begins in Section 8.1 with a high-level outline of the contributions of this thesis. It might be argued that certain portions of this thesis raise more questions than they answer; accordingly then, we turn to a discussion of some of these open questions in Section 8.2. Finally, we conclude in Section 8.3 with a rough sketch of the potential implications of our analysis for related research fields, including network information theory, iterative decoding, and large deviations analysis.

■ 8.1 High-level view

The specific goals of this thesis notwithstanding, its high-level contributions include the following:

- illustrating the power of exponential families, as well as the associated information geometry, for studying graphical models
- highlighting the fundamental role of the Legendre transform[1] between exponential and mean parameters
- bringing into sharp focus the crucial differences between tree-structured distributions[2], and distributions structured according to a graph with cycles

With a retrospective viewpoint, all of the contributions particular to this thesis can be understood in terms of these high-level issues. Indeed, the problem of estimation or inference corresponds, in the context of exponential representations, to computing certain elements of the forward Legendre mapping from exponential to mean parameters. The estimation problem makes a clear distinction between tree-structured problems (linear-time algorithms), and graphs with cycles (generally intractable). This distinction is quite broad: problems involving tree-structured distributions are, for the most part, well-understood, whereas their counterparts for graphs with cycles present considerable challenges. Accordingly, our strategies for tackling a range of problems were

[1]The Legendre transform is described in Section 2.2.4.

[2]More generally, we consider distributions structured according to a triangulated graph.

similar: we sought to gain insight about a problem on a graph with cycles by formulating and studying a set of modified problems defined on embedded trees. It was a line of attack that paid dividends for various problems, which we summarize briefly here.

The problem of Gaussian inference corresponds to computing the conditional means as well as the associated error covariances. This problem is equivalent to determining certain elements of the Legendre mapping from exponential parameters (e.g., specified by an inverse covariance matrix in the Gaussian case) to the mean parameters (e.g., the covariance matrix). For a graph with cycles, the embedded trees (ET) algorithm developed in Chapter 4 performs exact inference in an efficient manner by leveraging the existence of fast algorithms for tree-structured problems. Overall, this algorithm has the curious property of performing computations only on embedded trees, yet managing to solve *exactly* a problem on a graph with cycles.

The tree-based reparameterization framework of Chapter 5 gives a different perspective on belief propagation (BP), as well a class of related algorithms — namely, as a sequence of so-called reparameterization updates. Each such update entails altering the factorization of a graph-structured distribution, with the ultimate goal of ensuring that the factorization is consistent in a suitable sense on all embedded trees of the graph. The Legendre mapping figures prominently in defining the reparameterization operators. The use of an *overcomplete* exponential representation clarifies the fundamental property of reparameterization updates: true to their name, they do not change the overall distribution. This invariance property, in conjunction with the characterization of fixed points, has a number of important consequences. Perhaps the most important single consequence is the resultant insight into the error between the BP approximate marginals, and the actual marginals on the graph with cycles. In particular, it leads very naturally to an exact expression for this error, which serves as a starting point for developing bounds.

In Chapter 6, the notion of reparameterization is shown to be more generally applicable. Specifically, this chapter provides a unifying framework for a wide class of approaches to approximate inference, all of which (like belief propagation) are based on minimizing approximations to the Kullback-Leibler divergence. For each approximation in this class, we develop a corresponding set of reparameterization updates for attempting to obtain approximate marginals. Due to the central role of reparameterization, a satisfying fact is that all of the major results of Chapter 5 — including characterizing the fixed points of these algorithms, as well as analyzing the approximation error — carry over in a natural way.

The results of Chapter 7 provide an elegant illustration of the interplay between exponential representations, variational formulations and convex duality. It is natural, in the context of an exponential representation, to consider convex combinations of parameter vectors. As first described in Chapter 3, doing so leads to a new upper bound on the log partition function $\Phi(\theta^*)$ associated with an intractable distribution $p(\mathbf{x}; \theta^*)$. In the formulation of Chapter 7, the choice of exponential parameter vectors and weights in the convex combination ranges over all spanning trees of the graph. Since

reasonably complex graphs tend to have a very large number of spanning trees, the problem of optimizing this upper bound appears intractable. Nonetheless, considering the Lagrangian dual formulation, in which the Legendre mapping plays a central role, leads to considerable simplification. The dual problem not only gives an upper bound on the log partition function, but also can be optimized efficiently (albeit implicitly) over *all* spanning trees of the graph. Moreover, the form of this dual function brings the work of this thesis around in a full circle — in particular, by providing a novel and interesting perspective on the Bethe free energy [180] that is central to belief propagation. Not surprisingly then, the conditions for the optimum of the dual problem are remarkably similar to the consistency conditions associated with belief propagation (or equivalently, tree-based reparameterization).

■ 8.2 Suggestions for future research

From this thesis arise various specific problems for future research. In this section, we outline a number of these problems, as well as possible avenues of attack.

■ 8.2.1 Exact inference for Gaussian processes

From the perspective of numerical linear algebra, the embedded trees (ET) algorithm developed in Chapter 4 is related to the class of so-called Richardson methods [e.g., 54]. The ET algorithm is distinct from standard Richardson iterations, since it is time-varying (i.e., the embedded tree used can change from iteration to iteration). Trees, or more precisely forests, have been used in similar ways in past work (e.g., the alternating direction implicit (ADI) method [23, 136]). Although this use of embedded trees is certainly interesting, a more promising direction of research — at least for the goal of developing fast algorithms — is to consider embedded trees as a means of generating so-called preconditioning matrices. The convergence rate of various linear system solvers (e.g., conjugate gradient [54]) is known to depend on the condition number (ratio of maximum to minimum eigenvalues). The goal of preconditioning is to decrease this condition number so as to speed up convergence.

One interesting direction, then, is further exploration of trees as preconditioners for a linear system defined by a graph with cycles. Sudderth [163] gives some promising examples with regard to the eigenspectrum compression that can be achieved with trees as preconditioners. Of interest for understanding the behavior of preconditioned systems are the eigenvalues of quantities like $B^{-1}A$, or equivalently the generalized eigenvalues[3] of (A, B). In the context of graphical models, A should be viewed as a matrix respecting the structure of the graph with cycles, whereas B is the preconditioner (in our case, a tree-structured matrix). The area of *support graph theory* [e.g., 10, 27, 78, 81, 166] provides techniques for analyzing and bounding the eigenvalues of such systems. A clever idea in support graph theory is that of mapping each path in the original graph onto a path in the graph corresponding to the preconditioner. For example, if we use

[3]The generalized eigenvalues of (A, B) satisfy $A\mathbf{x} = \lambda B\mathbf{x}$ for some $\mathbf{x} \neq \mathbf{0}$.

an embedded tree as the preconditioner (B), then every path in the original graph can be mapped onto a unique path in the tree. A fundamental result is the congestion-dilation lemma [27, 78, 81], which relates the eigenvalues of $B^{-1}A$ to a product of the maximum congestion (roughly, a measure of how many paths use a given edge) times the maximum dilation (roughly, a measure of weighted path length). Elegant results of this nature have also been obtained in the spectral analysis of random walks on graphs [e.g., 56, 159].

It is likely that the use of a single tree as a preconditioner can be thoroughly analyzed by extant techniques in support graph theory. The most promising empirical results, however, are obtained not with a single tree, but rather with multiple trees. Empirical demonstrations of this phenomenon, as well as preliminary theoretical results, are presented in [163]. At a high level, the following research questions merit further study:

- how to precisely capture the effect of using multiple trees?
- how to develop techniques for optimizing the choice and ordering of multiple trees?

Any analysis, rather than being limited to trees, should apply more generally to triangulated graphs (e.g., graphs with treewidth ≥ 2). Overall, the perspective afforded by graphical models could provide valuable insights into the analysis of preconditioned linear systems.

■ 8.2.2 Approximate inference for discrete processes

The analysis of Chapters 5, 6 and 7, though contributing several important results on the subject of approximate inference, also raises a host of challenging questions, which we discuss in this section.

Uses of error bounds

An important result in Chapter 5 is the *exact* expression for the error between the approximate marginals computed by belief propagation (BP) and the actual marginals of $p(\mathbf{x}; \theta^*)$. Since this exact expression cannot be evaluated (in general), we also derived computable bounds on this error. Chapter 6 extended this error analysis to more advanced methods for approximate inference.

For the toy examples presented in both chapters (see, for example, Figures 5.13 and Figure 6.14), the error bounds are *quantitatively* useful: i.e., they provide relatively narrow windows in which the actual marginals must lie. As a consequence, if an approximate marginal (e.g., the BP approximation) happens to fall outside of these windows, then perforce it must be a poor approximation. A desirable feature of the bounds is that they are never vacuous.[4] However, for large problems, the upper bound (respectively lower bound) on the marginals may become arbitrarily close to one (respectively

[4]The union bound, for example, can make vacuous assertions: e.g., $\Pr(A) \leq 10$.

zero). To understand why, note that a major factor controlling the tightness of the log bounds is the absolute error in upper bounding the log partition function $\Phi(\theta^*)$. Unless the model parameters are somehow rescaled, this absolute error will grow as the problem size increases, so that the bounds will tend to extreme values. As a consequence, it is unlikely that (in general) the error bounds will remain quantitatively useful for large-scale problems.

However, the error bounds might still yield important *qualitative* information. For instance, the bounds could be used to assess the rate at which the accuracy of a fixed approximation deteriorates as the model parameters and/or structure are changed. Another possible application, discussed at more length below, is using the error bounds to assess the relative accuracy of a set of approximations for the same problem. Therefore, an open direction is exploring the uses of error bounds in application to large-scale problems. This avenue seems particularly promising given the wide range of important problems to which belief propagation has been applied [e.g., 65, 68, 130, 133].

The error bounds in Chapters 5 and 6 are formulated in terms of a fixed point of the minimization algorithm (e.g., BP/TRP in Chapter 5). However, even though it may not be obvious from the statement of the results, it is possible to compute bounds on the error at any iteration, thereby obviating the need to wait for convergence. To illustrate, consider the tree reparameterization (TRP) updates. After any update on a fixed tree $\mathcal{T}$, the current single node pseudomarginals are guaranteed to be globally consistent with with respect to the tree. The overall distribution on the graph with cycles is invariant under the updates, so that (as with TRP/BP fixed points) the approximations following any TRP iterate are related to the actual marginals by the perturbation of removing edges to reveal the tree $\mathcal{T}$. An argument of this nature applies more generally to the reparameterization algorithms described in Chapter 6. As a consequence, it becomes possible to assess the evolution of the error for each iterate of the algorithm (i.e., in a dynamic fashion). This type of dynamic assessment could be useful, for example, in coding applications where the decoding algorithm (an instantiation of BP) is not necessarily run until convergence. An optimistic view is that understanding the error evolution could help to specify termination times for which the approximation might be *better* than the fixed point obtained when the algorithm ultimately converges.

Choice of substructures

Common to all the approximations considered in Chapter 6 was a decomposition of the graph with cycles into a core structure, and a set of residual elements. The chapter itself provided a unified framework for analyzing approximate inference techniques based on such decompositions; largely left unanswered, however, was the crucial question of how to partition the graph into core and residual components.

Suppose that we are given two possible core and residual sets, and that we run minimization algorithms in order to compute a set of approximate marginals for each. As mentioned in our previous discussion, the error bounds provide one means of comparing

the relative accuracy[5] of these two sets of approximate marginals, albeit in a *post hoc* manner (i.e., only after both approximations have been computed).

Given the problem of computing approximate marginals for some graph-structured distribution, important but difficult open problems include the following:

- how to choose the optimal core structure from a fixed set of possibilities (e.g., the set of all spanning trees)
- assessing the potential accuracy of an approximation in an *a priori* manner (i.e., before running the reparameterization algorithm)
- determining the amount of computation required (i.e., how much structure to incorporate) in order to achieve a pre-specified level of accuracy

At a high level, the analysis and examples of Chapter 6 show that the accuracy of the approximations depends on the strength of higher-order interactions among subsets of nodes. As a result, these research problems all touch upon a fundamental question in graphical models: how to specify precisely the way in which graph structure and the settings of potential functions control interactions among random variables? Answers to questions of this nature, though clearly related to the Legendre transform between exponential and mean parameters, are understood (at best) only partially. We shall return to discussion of this issue in Section 8.2.3.

Uses of convexified Bethe free energies

As noted in Section 7.3.4 of Chapter 7, for each vector $\boldsymbol{\mu}_e$ in the spanning tree polytope $\mathbb{T}(\mathcal{G})$, Theorem 7.3.1(a) guarantees that

$$\mathcal{F}(\lambda; \boldsymbol{\mu}_e; \theta^*) = -\sum_{s \in \mathcal{V}} H_s(\lambda) + \sum_{(s,t) \in \mathcal{E}} \mu_{st} I_{st}(\lambda) - \sum_{\alpha} \lambda_\alpha \theta^*_\alpha \tag{8.1}$$

is convex as a function of λ. In fact, functions in the form of equation (8.1) can be viewed as a convexified versions of the Bethe free energy [180]. Indeed, making the (generally) invalid[6] choice $\boldsymbol{\mu}_e = \mathbf{1}$ in equation (8.1) gives rise to the Bethe free energy.

The results of Yedidia et al. [180] establish that belief propagation attempts to minimize the Bethe free energy. The minimizing arguments are taken as approximations to the actual marginals of the original distribution $p(\mathbf{x}; \theta^*)$. A similar program can be pursued for the family of functions in equation (8.1): namely, given a fixed $\boldsymbol{\mu}_e \in \mathbb{T}(\mathcal{G})$ and fixed θ^*, minimize the function of equation (8.1) subject to λ belonging to the linear polytope $\mathbb{L}(\mathcal{G})$ defined in equation (7.13); and then take the minimizing arguments as approximations to the actual marginals. An advantage of this proposal (compared to

[5]The advantage of considering relative versus absolute accuracy is that the intractable log partition function $\Phi(\theta^*)$ need not be considered because it cancels out.

[6]The vector of all ones does *not* belong to the spanning tree polytope, except when (of course) $\mathcal{G}$ is actually a tree.

minimizing the Bethe free energy) is that the problem is convex with linear constraints, so that a unique global minimum is guaranteed to exist, and a variety of techniques [e.g., 20] can be used to find it.[7] As a result, this formulation obviates failures of convergence and the possibility of multiple (local) minima associated with the Bethe free energy and belief propagation.

Open questions associated with this proposal include:

- For a fixed choice $\boldsymbol{\mu_e} \in \mathbb{T}(\mathcal{G})$, how do the approximate marginals $\widehat{\lambda}$ compare to the BP approximations, or to the actual marginals of $p(\mathbf{x}; \theta^*)$?
- How does the choice of $\boldsymbol{\mu_e}$ affect the approximation? Are certain approximations best-suited to certain graphs?

For the time being, we note that results of Chapters 5 and 7, when suitably modified, can be used to derive an exact expression for the error between the actual marginals, and these new approximations $\widehat{\lambda}$. Again, this exact expression would form the basis for developing bounds.

Another interesting problem to tackle is the evolution of the approximate marginals as a line is traced from a valid choice $\boldsymbol{\mu_e} \in \mathbb{T}(\mathcal{G})$ to the vector $\mathbf{1}$ corresponding to BP. In this context, methods from homotopy theory [4] should be useful (as in [40]). This line of research is likely to have practical consequences, such as better techniques for finding BP fixed points, as well as theoretical implications. For example, tracing the evolution of fixed points would identify bifurcations in the solution space. Such sharp transitions must occur, because for any $\boldsymbol{\mu_e} \in \mathbb{T}(\mathcal{G})$, the associated cost function in equation (8.1) has a single global minimum, whereas the Bethe free energy typically has multiple local minima.

■ 8.2.3 Bounds

The techniques of Chapter 3, in conjunction with the upper bounds of Chapter 7, permit the efficient computation of upper and lower bounds on local marginal probabilities (e.g., the single-node and joint pairwise marginals) associated with any distribution $p(\mathbf{x}; \theta)$. Although these results are clearly useful, the analysis itself was myopic, in that each bound was considered in isolation. Therefore, it is interesting to consider how to strengthen bounds by taking into account more global relations.

Let us focus on a concrete example for the purposes of illustration. Consider the Ising model: i.e., a binary vector $\mathbf{x} \in \{0, 1\}^N$ with distribution of the form:

$$p(\mathbf{x}; \theta) = \exp\Big\{ \sum_{s \in \mathcal{V}} \theta_s x_s + \sum_{(s,t) \in \mathcal{E}} \theta_{st} x_s x_t - \Phi(\theta) \Big\} \tag{8.2}$$

Recall that the associated set of $N + |\mathcal{E}|$ dual variables is given by

$$\begin{aligned} \eta_s &= \mathbb{E}_\theta[x_s] = p(x_s = 1; \theta) \quad \text{for all } s \in \mathcal{V} \\ \eta_s &= \mathbb{E}_\theta[x_s\, x_t] = p(x_s = 1, x_t = 1; \theta) \quad \text{for all } (s,t) \in \mathcal{E} \end{aligned}$$

[7] The constrained Newton's method described in Algorithm 7.4.1 is one possibility.

Now it is clear that the set of all valid dual variables is a subset of $[0,1]^{N+|\mathcal{E}|}$. It is, in fact, a strict subset since the dual variables are coupled by a number of relations. Some of these relations follow trivially from set inclusion properties; e.g., the inequality

$$p(x_s = 1, x_t = 1; \theta^*) \leq p(x_s = 1; \theta^*) \tag{8.3}$$

holds for all pairs of nodes (s,t).

It turns out that for a tree-structured distribution, local inequalities like that of equation (8.3) between variables at adjacent nodes $(s,t) \in \mathcal{E}$ are sufficient to characterize the set of valid dual variables. For a graph with cycles, in contrast, there are highly non-local constraints among the dual variables.

Example 8.2.1 (Invalid set for single cycle). Consider a single cycle on 4 nodes, and set $\eta_s = 0.5$ for all $s \in \{1,2,3,4\}$. Set the pairwise dual variables as $\eta_{12} = \eta_{23} = \eta_{34} = 0.4$ and $\eta_{14} = 0.1$. It can be verified that any subset of these dual variables corresponding to a tree embedded within the graph[8] is valid. However, as shown in [57], the full set of dual variables is not consistent with any distribution.

Therefore, characterizing the set of valid dual variables for a graph with cycles is an interesting problem. Essentially, the question that we have posed is to characterize the *range of the Legendre mapping* $\Lambda : \theta \mapsto \eta$. In certain cases (e.g., the binary case; the Gaussian case), this range set can be characterized in some detail. Answering the same question for more general distributions and graphs remains an important direction to explore. If we return to the original issue motivating this discussion, it should be clear that a deeper understanding of these relations would be helpful in tightening bounds.

■ 8.3 Possible implications for related fields

This section provides a discussion of related research fields where the results of this thesis may have implications.

■ 8.3.1 Network information theory

The subject of network information theory [e.g., 41, 60] is the coding and transmission of information in a distributed network. The basic problem, at a high-level, is analogous to that of classical (single-user) information theory — that is, given a set of sources and receivers and a channel model that describes possible noise and interference, how to transmit the sources reliably over this channel? Despite this conceptual similarity, most problems in multiuser information theory have turned out to be far more difficult than their corresponding analogues in the single-user setting.

For example, one specific problem of interest is computing the capacity region associated with a particular arrangement of sources and receivers. For a given source-receiver pair, an achievable rate is one for which information can be transmitted with

[8] In this case, a subset corresponding to a tree consists of all the single node variables, together with any three of the four pairwise variables.

a probability of error that tends to zero asymptotically; see Cover [41]. The capacity region of a network is the set of achievable rates for all source-receiver pairs. For special cases (e.g., broadcast channel; multiple access channel), these regions are relatively well-understood. For more complicated networks, the most generally applicable tool is a bound derived from the min-cut/max-flow duality that is well-known from network optimization [e.g., 26]. The resulting bounds, however, are quite weak in most cases.

From the point of view of exponential representations and graphical models, the essence of the difficulty is akin to the Legendre mapping problem, as discussed in Section 8.2.3. That is, it can be extremely difficult to characterize the statistical dependencies that arise among a collection of random variables linked in a network with cycles. From this perspective, the classic channel in information theory is relatively easy to analyze precisely because its graphical structure (namely, that of a chain) is relatively simple. The corresponding problems for more complicated graphs, however, are much more difficult because there can be highly non-trivial and non-local interaction among subsets of variables. Overall, the framework of exponential representations and graphical models may be useful in studying problems of network information theory.

■ 8.3.2 Analysis of iterative decoding

One manifestation of belief propagation is as a highly successful iterative decoding technique for various codes defined by graphical models, including turbo codes [e.g., 18, 130] and low-density parity check codes [e.g., 70, 125, 129, 149]. As a consequence, the results of Chapter 5 — especially the error analysis — have implications for coding theory.

In recent work, several groups of researchers [e.g., 125,148,149] have obtained results on the performance of belief propagation decoding of low-density parity check codes. For instance, a remarkable result established by Richardson et al. [149], building from the work in [125], is a capacity-like notion for BP decoding: namely, the existence of noise thresholds (dependent on the channel and code) below which the probability of error tends to zero exponentially in the code length, and above which the error probability is bounded away from zero. In many cases, the density evolution technique [70, 148] can be used to calculate these thresholds.

Two key features of this work are the following:

- the analysis is asymptotic as the code length (or number of nodes in the graph) N tends to infinity
- it entails averaging over all codes in a random ensemble (in addition to averaging over channel noise)

Considering asymptotic behavior permits the application of powerful concentration theorems [e.g., 125]. These results, which are based on martingale sequences [e.g., 80], establish that the limiting behavior of the decoder on a randomly-chosen code becomes concentrated around its expected behavior with high probability. Averaging over all

codes in an ensemble is also important, because it permits probabilistic analysis of the structure of random graphs (associated with randomly-chosen codes). As $N \to \infty$, the graphs in a typical ensemble become tree-like, so that the expected behavior of BP decoding converges to the tree case.[9]

As a consequence, important open questions in iterative decoding include:

- analyzing BP decoding for codes of relatively short lengths (e.g., $\approx 10^3$ bits) for which asymptotic analysis does not apply
- analysis of BP decoding averaged over the channel noise but for a *fixed* code

The results in Chapter 5, in which we gave an exact expression and bounds for the BP error for an arbitrary graph with cycles, appear germane to these problems. It is conceivable, then, to try to calculate averages of the error over the channel noise (which amounts to averaging over random choices of the single node potentials).

To sketch a line of attack in a bit more detail, optimal bit-wise decoding of a binary code is based on the sign of the log likelihood ratio (LLR). Suppose that we represent the problem of decoding a linear binary code defined by graph in an exponential manner—that is, as performing inference for a distribution $p(\mathbf{x}; \theta^*)$ that captures both the parity checks[10] defining the code, as well as the noisy observations of transmitted bits. With this notation, the LLR for optimal decoding is given by $\log[p(x_s = 1; \theta^*)/p(x_s = 0; \theta^*)]$. The results of Chapter 5 show that approximate BP decoding is based, instead, on the sign of the modified LLR $\log[p(x_s = 1; \Pi^i(\theta^*))/p(x_s = 0; \Pi^i(\theta^*))$, where $p(\mathbf{x}; \Pi^i(\theta^*))$ is a distribution structured according to a particular (but arbitrary) tree embedded within the graph with cycles representing the code. This relation suggests a new avenue for analyzing the error between the optimal and BP log likelihood ratios for an arbitrary but fixed code.

■ 8.3.3 Application to large deviations analysis

Large deviations theory [e.g., 53, 158] treats the probability of certain events in the limit of large sample sizes. For example, one might be interested in the probability of obtaining 900 or more heads in 1000 tosses of a fair coin. This is certainly an unlikely event; alternatively phrased, it is a large deviation in the sample mean of heads (here 900/1000 or 0.9) from the true mean (0.5 for a fair coin). Of particular interest are the rates that govern the exponential decay of these probabilities as a function of the sample size. In this context, the log partition function (or equivalently, the cumulant generating function) is well-known to play the role of a *rate function* [see, e.g., 158]. As

[9] To be precise, most random ensembles have the property that for any positive integer $k \geq 3$, the probability that a graph chosen randomly from the ensemble has cycles of length $\leq k$ tends to zero [e.g., 149].

[10] Technically, since parity checks entail deterministic constraints , it would be necessary to either use an extended exponential representation in which elements θ^*_α can assume infinite values, or to consider ϵ-approximations to deterministic parity checks.

a consequence, the bounds of Chapter 7 are potentially useful in application to large deviations.

Exponential formulation of large deviations

To illustrate, we formulate a simple large deviations result using an exponential representation. Consider an exponential family $\{p(\mathbf{x};\theta)\}$ of distributions specified by a set of functions $\boldsymbol{\phi} = \{\phi_\alpha\}$. We assume that there is some underlying graph $\mathcal{G}$ such that any potential function on a clique of $\mathcal{G}$ can be expressed in terms of $\boldsymbol{\phi}$. However, we also allow the possibility that the set $\boldsymbol{\phi}$ may include functions *in addition* to these clique potentials.[11] Let $p(\mathbf{x};\widetilde{\theta})$ be a distribution in this exponential family, and let $\eta_\alpha = \mathbb{E}_{\widetilde{\theta}}[\phi_\alpha(\mathbf{x})]$ be the associated dual variables.

Now suppose that we are given a set of n samples $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$, where each $\mathbf{x}^{(i)}$ is sampled IID from $p(\mathbf{x};\widetilde{\theta})$. In the coin tossing example introduced earlier, each random variable $\mathbf{x}^{(i)}$ would be a indicator function for the event of obtaining a head on the i^{th} trial. Let $p(\mathbf{X};\widetilde{\theta}^n)$ denote the product distribution $\prod_{i=1}^n p(\mathbf{x}^{(i)};\widetilde{\theta})$.

We are interested in the probability that the sample mean of some random variable — say specified by the linear combination $b^T\boldsymbol{\phi}(\mathbf{x})$ — exceeds its true mean $b^T\eta$ by more than $\epsilon > 0$. To be precise, the quantity of interest is the probability:

$$P(n;\widetilde{\theta};\epsilon) \triangleq \Pr_{\widetilde{\theta}^n}\left\{\frac{1}{n}\sum_{i=1}^{n} b^T\left[\boldsymbol{\phi}(\mathbf{x}^{(i)}) - \eta\right] \geq \epsilon\right\} \tag{8.4}$$

where $\Pr_{\widetilde{\theta}^n}$ denotes probability under the product distribution $p(\mathbf{X};\widetilde{\theta})$.

A standard upper bound on this probability is the following *Chernoff bound*:

$$P(n;\widetilde{\theta};\epsilon) \leq \exp\left\{-n \min_{\theta \text{ s.t } \mathbb{E}_\theta[b^T\boldsymbol{\phi}(\mathbf{x})]=b^T\eta+\epsilon} D(\theta \,\|\, \widetilde{\theta})\right\} \tag{8.5}$$

where $D(\theta \,\|\, \widetilde{\theta})$ is the Kullback-Leibler (KL) divergence between $p(\mathbf{x};\theta)$ and $p(\mathbf{x};\widetilde{\theta})$. The key point here is that the optimal error exponent, which governs the rate of decay as n tends to infinity, is given by the closest (as measured by KL divergence) distribution to $p(\mathbf{x};\widetilde{\theta})$ that satisfies the moment constraint (i.e., $\mathbb{E}_\theta[b^T\boldsymbol{\phi}(\mathbf{x})] = b^T\eta + \epsilon$).

Applying bounds from Chapter 7

At an intermediate stage in one proof of equation (8.5), the following relation, valid for all $\delta > 0$, arises:

$$\frac{1}{n}\log P(n;\widetilde{\theta};\epsilon) \leq \Phi(\widetilde{\theta} + \delta b) - \delta[b^T\eta + \epsilon] - \Phi(\widetilde{\theta}) \tag{8.6}$$

[11] This additional flexibility will be important in certain cases, as we will see in Example 8.3.1.

It is by minimizing this RHS over all $\delta > 0$ that we obtain the Kullback-Leibler error exponent of equation (8.5). (See the form of the Kullback-Leibler divergence given in equation (2.44)).

Now suppose that the reference distribution $p(\mathbf{x}; \widetilde{\theta})$ is simple enough so that it is possible to compute $\Phi(\widetilde{\theta})$, but that computing $\Phi(\widetilde{\theta} + \delta b)$ is intractable.

Example 8.3.1. As a simple illustrative example, suppose that $p(\mathbf{x}; \widetilde{\theta})$ corresponds to a (first-order) discrete-time Markov chain on N points. This Markov process can be viewed as living on a linear chain, so that the cliques correspond to singleton nodes and pairs of adjacent nodes. Suppose moreover that we are interested in the probability that the product of the end point random variables (i.e., $\phi_\beta(\mathbf{x}) = x_1 x_N$) exceeds a threshold. This ϕ_β function is *not* a clique potential of the original graph, but can be viewed as a clique potential on an augmented graph (namely, a single cycle).

In this case, the vector b is equal to $\mathbf{e}_\beta$ — that is, the vector of all zeros with a single one in element β. The ϕ_β potential couples the starting and end points, so that for all $\delta > 0$, $p(\mathbf{x}; \widetilde{\theta} + \delta \mathbf{e}_\beta)$ is a distribution with the structure of a single cycle. Thus, the quantity $\Phi(\widetilde{\theta} + \delta \mathbf{e}_\beta)$ is no longer computable by standard tree algorithms.[12]

One can imagine a variety of such scenarios, in which computing $\Phi(\widetilde{\theta} + \delta b)$ is not tractable. For these cases, exact computation of the error exponent in equation (8.5), would be impossible, so that it would be useful to obtain an upper bound. The results of Chapter 7 are relevant in this context. Given any $\boldsymbol{\mu_e} \in \mathbb{T}(\mathcal{G})$, we can use Theorem 7.3.1 to prove that:

$$\log P(n; \widetilde{\theta}; \epsilon) \leq -\mathcal{H}(\boldsymbol{\mu_e}; \widetilde{\theta} + \delta b) - \delta \left[b^T \eta + \epsilon \right] - \Phi(\widetilde{\theta}) \tag{8.7}$$

where $\mathcal{H}$ is defined in equation (7.35). It can also be shown that the RHS is a strictly convex function of δ, so that there is a unique $\widehat{\delta} > 0$ that attains the tightest possible upper bound of this form. Equation (8.7) can be viewed as a poor man's version of the Chernoff bound (8.5), but with the advantage of being computable.

[12] For this simple example, $\Phi(\widetilde{\theta} + \delta \mathbf{e}_\beta)$ could still be computed, for instance, by applying the junction tree algorithm (see Section 2.1.5) to the single cycle, but one can imagine more complex scenarios for which this quantity is truly intractable.

Appendix A

Algorithms for optimal estimation on trees

Here we derive the key equations underlying a variety of algorithms for computing posterior marginal distributions in tree-structured graphs.

■ A.1 Partial ordering in scale

The critical property of a tree-structured graph $\mathcal{T}$ is that its nodes $s \in \mathcal{V}$ can be partially ordered according to their *scale*. (See [168] for a precise definition of a partial ordering). In order to define the notion of scale, we begin by designating an arbitrary node as the *root*; we assume without loss of generality that that the root is labeled with $s = 1$. Once the root is specified, the other nodes ($s \in \mathcal{V}/\{1\}$) can be assigned a scale $i = 0, 1, \ldots, I$ based on their distance from the root. This distance is given by the number of edges in the unique path joining s and the root node. Accordingly, the root is the only node to be assigned scale $i = 0$. At the next finest scale $i = 1$ are $q(0)$ nodes, that correspond to the *children* of the root node. A node at scale $i < I$ gives birth to its children at the next scale $(i + 1)$. The children of node s are indexed by $s\alpha_1, \ldots s\alpha_{q(s)}$, and we let $\mathrm{Ch}(s)$ denote the set of all children of node s. Similarly, each node s at scale $i > 0$ has a unique parent $\bar{\gamma}s$ at scale $(i - 1)$. This hierarchical tree organization is illustrated in Figure A.1(a).

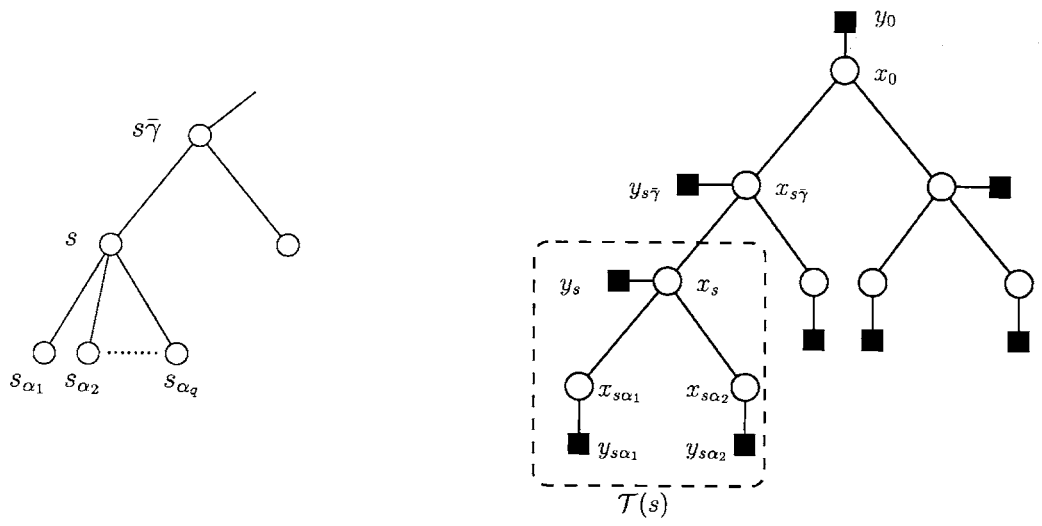
■ A.2 Basic notation

Lying at each node s is a random variable x_s, to which we will refer as the state variable. In Figure A.1(b), these nodes are illustrated with circles. Dangling from each state node is a node containing an observation y_s; in Figure A.1(b), these nodes are drawn with squares. By concatenating these quantities, we define the vectors

$$\mathbf{x} \triangleq \{ x_s \mid s \in \mathcal{V} \} \tag{A.1a}$$

$$\mathbf{y} \triangleq \{ y_s \mid s \in \mathcal{V} \} \tag{A.1b}$$

For any node s, we let $\mathcal{T}(s)$ denote the vertices in the subtree of $\mathcal{T}$ rooted at node



(a) (b)

Figure A.1. (a) Basic types of nodes in a tree: node s and its parent $s\bar{\gamma}$, and children $\{s_{\alpha_1}, \ldots s_{\alpha_q}\}$. (b) Assignment of random variables to nodes of the tree, and definition of subtree $\mathcal{T}(s)$ rooted at s. The data $\mathbf{y}_s$ consists of all the data $\{ y_t \mid t \in \mathcal{T}(s) \}$ at nodes in $\mathcal{T}(s)$.

s. We then define

$$\mathbf{y}_s \triangleq \{ y_t \mid t \in \mathcal{T}(s) \} \tag{A.2}$$

to be the collection of observations in the subtree rooted at s. This set $\mathbf{y}_s$ is illustrated in Figure A.1(b). We let $\mathbf{y}_s^c \equiv \mathbf{y}/\mathbf{y}_s$ denote the complement of $\mathbf{y}_s$ in full data set $\mathbf{y}$.

We shall frequently exploit the following decomposition:

$$\mathcal{T}(s) = \{s\} \cup \left[\cup_{t \in \mathrm{Ch}(s)} \mathcal{T}(t) \right] \tag{A.3}$$

Equation (A.3) expresses the fact that the subtree $\mathcal{T}(s)$ is the disjoint union of node s and the subtrees rooted at children of s.

■ A.3 Markov decomposition

At the heart of the two-pass tree smoothing algorithms is the following decomposition of the single-node marginal probability $p(x_s \mid \mathbf{y})$:

Proposition A.3.1 (Key decomposition). For any node s, the following decomposition holds:

$$p(x_s \mid \mathbf{y}) \;=\; \kappa \; p(y_s \mid x_s) \, p(x_s \mid \mathbf{y}_s^c) \prod_{x_t \in \mathrm{Ch}(t)} p(\mathbf{y}_t \mid x_s) \tag{A.4}$$

where κ is a normalization constant independent of x_s.

Proof. The proof is a basic exercise in applying Bayes' rule, and exploiting the Markov properties of a tree-structured distribution. We write:

$$\begin{aligned} p(x_s \,|\, \mathbf{y}) &= \kappa \; p(\mathbf{y} \,|\, x_s) \, p(x_s) && \text{(A.5a)} \\ &= \kappa \; p(\mathbf{y}_s \,|\, x_s) \, p(\mathbf{y}_s^c \,|\, x_s) \, p(x_s) && \text{(A.5b)} \\ &= \kappa \; p(\mathbf{y}_s \,|\, x_s) \, p(x_s \,|\, \mathbf{y}_s^c) && \text{(A.5c)} \\ &= \kappa \; p(y_s \,|\, x_s) \Big[\prod_{t \in \mathrm{Ch}(s)} p(\mathbf{y}_t \,|\, x_s) \Big] \; p(x_s \,|\, \mathbf{y}_s^c) && \text{(A.5d)} \end{aligned}$$

where the definition of the normalization constant κ has changed from line to line. Here equation (A.5a) follows from Bayes' rule; equation (A.5b) follows from the fact that $\mathbf{y}_s$ and $\mathbf{y}_s^c$ are conditionally independent given x_s; equation (A.5c) follows from Bayes' rule applied to $p(\mathbf{y}_s^c \,|\, x_s) \; p(x_s)$; and equation (A.5d) follows from Markov properties associated with the decomposition in equation (A.3).

□

Equation (A.4) reveals the computation of $p(x_s \,|\, \mathbf{y})$ requires two types of quantities:

(a) the likelihoods $p(\mathbf{y}_t \,|\, x_s)$ of the data $\mathbf{y}_t$ in the subtree $\mathcal{T}(t)$ given x_s, where t is a child of s

(b) the conditional probabilities $p(x_s \,|\, \mathbf{y}_s^c)$.

In the following sections, we describe a recursive upward sweep for computing the likelihoods, and then a downward pass for recursively computing the conditional probabilities.

■ A.4 Upward sweep

Proposition A.4.1 (Likelihood calculation). For any node x_s, the following factorization holds:

$$\begin{aligned} p(\mathbf{y}_s \,|\, x_s) &= \kappa \; p(y_s | x_s) \prod_{t \in \mathrm{Ch}(s)} p(\mathbf{y}_t \,|\, x_s) && \text{(A.6a)} \\ &= \kappa \; p(y_s \,|\, x_s) \prod_{t \in \mathrm{Ch}(s)} \int_{x_t} p(\mathbf{y}_t \,|\, x_t) \, p(x_t \,|\, x_s) dx_t && \text{(A.6b)} \end{aligned}$$

where κ is an arbitrary normalization constant.

Proof. Equation (A.6a) follows from the Markov properties associated with the decomposition of equation (A.3). Equation (A.6b) follows from the relation:

$$p(\mathbf{y}_t \,|\, x_s) = \int p(\mathbf{y}_t \,|\, x_s, x_t) \, p(x_t \,|\, x_s) dx_t$$

and the fact that $\mathbf{y}_t$ is conditionally independent of x_s given x_t whenever $t \in \mathrm{Ch}(s)$. □

■ A.5 Downward sweep

Similarly, the conditional probabilities can be computed by a downward sweep:

Proposition A.5.1 (Conditional calculation). For any node x_s, the conditional probabilities $p(x_s \mid \mathbf{y}_s^c)$ can be computed via the following recursions:

$$p(x_s \mid \mathbf{y}_s^c) = \kappa \int_{x_{s\bar{\gamma}}} p(y_{s\bar{\gamma}} | x_{s\bar{\gamma}}) \left[\prod_{t \in \mathrm{Ch}(s\bar{\gamma})/s} p(\mathbf{y}_t \mid x_{s\bar{\gamma}}) \right] p(x_{s\bar{\gamma}} \mid \mathbf{y}_{s\bar{\gamma}}^c)\, p(x_s \mid x_{s\bar{\gamma}}) dx_{s\bar{\gamma}} \tag{A.7a}$$

$$p(\mathbf{y}_t \mid x_{s\bar{\gamma}}) = \int p(\mathbf{y}_t \mid x_t)\, p(x_t \mid x_{s\bar{\gamma}})\, dx_t \tag{A.7b}$$

Equation (A.7a) is a recursive formula for computing the conditional probabilities $p(x_s \mid \mathbf{y}_s^c)$ in terms of the parent quantity $p(x_{s\bar{\gamma}} \mid \mathbf{y}_{s\bar{\gamma}}^c)$; the local observation $p(y_{s\bar{\gamma}} | x_{s\bar{\gamma}})$; and the likelihoods $p(\mathbf{y}_t \mid x_{s\bar{\gamma}})$. The latter quantities can be computed from the upward sweep via equation (A.7b).

Proof. To establish equation (A.7a), we write:

$$p(x_s \mid \mathbf{y}_s^c) = \kappa\, p(\mathbf{y}_s^c \mid x_s)\, p(x_s) \tag{A.8a}$$

$$= \kappa \int p(\mathbf{y}_s^c \mid x_{s\bar{\gamma}})\, p(x_{s\bar{\gamma}} \mid x_s)\, p(x_s)\, dx_{s\bar{\gamma}} \tag{A.8b}$$

$$= \kappa \int p(\mathbf{y}_s^c \mid x_{s\bar{\gamma}})\, p(x_{s\bar{\gamma}})\, p(x_s \mid x_{s\bar{\gamma}})\, dx_{s\bar{\gamma}} \tag{A.8c}$$

$$= \kappa \int p(y_{s\bar{\gamma}} \mid x_{s\bar{\gamma}})\, p(\mathbf{y}_{s\bar{\gamma}}^c \mid x_{s\bar{\gamma}}) \left[\prod_{t \in \mathrm{Ch}(s\bar{\gamma})/s} p(\mathbf{y}_t \mid x_{s\bar{\gamma}}) \right] p(x_{s\bar{\gamma}})\, p(x_s \mid x_{s\bar{\gamma}})\, dx_{s\bar{\gamma}} \tag{A.8d}$$

$$= \kappa \int p(y_{s\bar{\gamma}} \mid x_{s\bar{\gamma}})\, p(x_{s\bar{\gamma}} \mid \mathbf{y}_{s\bar{\gamma}}^c) \left[\prod_{t \in \mathrm{Ch}(s\bar{\gamma})/s} p(\mathbf{y}_t \mid x_{s\bar{\gamma}}) \right] p(x_s \mid x_{s\bar{\gamma}})\, dx_{s\bar{\gamma}} \tag{A.8e}$$

Here equation (A.8a) follows from Bayes' rule; equation (A.8b) follows from the conditional independence $p(\mathbf{y}_s^c \mid x_s, x_{s\bar{\gamma}}) = p(\mathbf{y}_s^c \mid x_{s\bar{\gamma}})$; and equation (A.8c) is another application of Bayes' rule. Next equation (A.8d) follows from the Markov properties associated with the decomposition:

$$\mathbf{y}_s^c = \{y_{s\bar{\gamma}}\} \cup \mathbf{y}_{s\bar{\gamma}}^c \cup \left[\cup_{t \in \mathrm{Ch}(s\bar{\gamma})/s} \mathbf{y}_t \right]$$

Equation (A.8e) follows from a final application of Bayes' rule.

Equation (A.7b) follows from the fact that $p(\mathbf{y}_t \mid x_t, x_{s\bar{\gamma}}) = p(\mathbf{y}_t \mid x_t)$ whenever t is a child of $s\bar{\gamma}$. □

On the basis of Propositions A.4.1 and A.5.1, it is possible to develop a number of recursive algorithms for computing posterior marginals at each node of the graph. For example, one such algorithm is a generalization of the Maynes-Fraser [64] algorithm for smoothing of time series (i.e., defined on a chain) to more general graphs. It is also straightforward to derive an alternative algorithm in which data is incorporated only on the upward pass. This leads to a downward recursion in terms of the conditional probabilities $p(x_s \mid \mathbf{y})$. This algorithm is a generalization of the Rauch-Tung-Striebel [146] smoother to arbitrary (non-Gaussian) stochastic processes, and general trees.

It is also possible to derive similar algorithms for MAP estimation. These algorithms are formally equivalent to those for computing posterior marginals, in that all integrals are replaced by a maximization operation. In fact, dynamic programming algorithms of this nature can be generalized so as to apply to any commutative semi-ring on which two binary operations are defined [see 169], a general and elegant view emphasized recently in [3].

Appendix B

Proofs for Chapter 3

■ B.1 Proof of Proposition 3.3.1

Using equation (3.11) and the convexity of Φ_f, for any pair θ^* and θ we write:

$$\Phi_f(\theta^*) \geq \Phi_f(\theta) + \sum_\alpha \frac{\partial \Phi_f(\theta)}{\partial \theta_\alpha} [\theta^* - \theta]_\alpha \tag{B.1}$$

Using equations (3.9) and (2.18a), we compute $\frac{\partial \Phi_f}{\partial \theta_\alpha}(\theta) = \mathbb{E}_\theta[\phi_\alpha] + \frac{\mathrm{cov}_\theta\{f, \phi_\alpha\}}{\mathbb{E}_\theta[f]}$. We substitute this relation, as well as the forms of $\Phi_f(\theta)$ and $\Phi_f(\theta^*)$ specified by equation (3.9), into equation (B.1) to obtain:

$$\Phi(\theta^*) + \log \mathbb{E}_{\theta^*}[f] \geq \Phi(\theta) + \log \mathbb{E}_\theta[f] + \sum_\alpha [\theta^* - \theta]_\alpha \Big\{ \mathbb{E}_\theta[\phi_\alpha] + \frac{\mathrm{cov}_\theta\{f, \phi_\alpha\}}{\mathbb{E}_\theta[f]} \Big\}$$

Applying the form of the KL divergence in equation (2.31) and re-arranging yields the bound of equation (3.13a).

In order to derive equation (3.13b), we observe that whenever f satisfies Assumption 3.3.1, then so does the function $\widetilde{f}(\mathbf{x}) \triangleq 1 - f(\mathbf{x})$. We can therefore apply equation (3.13a) to $\widetilde{f}$ to derive an upper bound on $\mathbb{E}_{\theta^*}[f]$.

■ B.2 Proof of Proposition 3.3.2

The log partition function Φ_f corresponding to the tilted distribution is convex, so that we can apply Jensen's inequality in equation (3.12). For any $(\boldsymbol{\theta}; \vec{\boldsymbol{\mu}}) \in \mathcal{A}(\theta^*)$, we have $\Phi_f(\theta^*) = \Phi_f(\mathbb{E}_{\vec{\boldsymbol{\mu}}}[\theta^i]) \leq \mathbb{E}_{\vec{\boldsymbol{\mu}}}[\Phi_f(\theta^i)]$. Using equation (3.9), this is equivalent

$$\Phi(\theta^*) + \log \mathbb{E}_{\theta^*}[f] \leq \mathbb{E}_{\vec{\boldsymbol{\mu}}}\big[\Phi(\theta^i) + \log \mathbb{E}_{\theta^i}[f]\big]$$

which, after re-arranging, is equivalent to the upper bound of equation (3.21a). The weaker upper bound of equation (3.21b) follows by applying the standard mean field lower bound (see Section 2.3.1) on the the log partition function to each exponential parameter θ^i — that is:

$$\Phi(\theta^*) \geq \Phi(\theta^i) + \sum_\alpha \mathbb{E}_{\theta^i}[\phi_\alpha]\, [\theta^* - \theta^i]_\alpha$$

■ B.3 Proof of Proposition 3.4.1

The bound of equation (3.25) follows by applying Proposition 3.3.1 to each individual f^k, and then summing the bounds to obtain a bound on $\mathbb{E}_{\theta^*}[f] = \sum_k \mathbb{E}_{\theta^*}[f^k]$. To prove the superiority of this new bound, define

$$B^k = -D(\theta \,\|\, \theta^*) + \frac{1}{\mathbb{E}_\theta[f^k]} \sum_\alpha (\theta^* - \theta)_\alpha \langle f^k, \phi_\alpha \rangle_\theta$$

and write the log of LHS of equation (3.25) as

$$\begin{aligned}
\log\Big[\sum_k \mathbb{E}_\theta[f^k]\exp(B^k)\Big] &= \log \mathbb{E}_\theta[f] + \log\Big[\sum_k \frac{\mathbb{E}_\theta[f^k]}{\mathbb{E}_\theta[f]}\exp(B^k)\Big] \\
&\geq \log \mathbb{E}_\theta[f] + \sum_k \frac{\mathbb{E}_\theta[f^k]}{\mathbb{E}_\theta[f]} B^k \\
&= \log \mathbb{E}_\theta[f] - D(\theta \,\|\, \theta^*) + \frac{1}{\mathbb{E}_\theta[f]} \sum_\alpha (\theta^* - \theta)_\alpha \langle \sum_k f^k, \phi_\alpha \rangle_\theta \\
&= \log \mathbb{E}_\theta[f] - D(\theta \,\|\, \theta^*) + \frac{1}{\mathbb{E}_\theta[f]} \sum_\alpha (\theta^* - \theta)_\alpha \, \langle f, \phi_\alpha \rangle_\theta
\end{aligned}$$

where the final line is the log of the bound given in Proposition 3.3.1. Here the inequality follows from the concavity of the logarithm. This inequality is strict as long as $\mathbb{E}_\theta[f^k] > 0$ for all k (which holds for all $f^k \neq 0$), and the B^k terms are not all equal to the same constant.

■ B.4 Proof of Proposition 3.4.2

The bounds of equation (3.27) themselves follow by applying Proposition 3.3.2 to each f^k, and then summing the bounds on each f^k to obtain a bound on f.

To prove superiority of these bounds to those of Proposition 3.3.2, we require the following lemma, proved in §6.9 of Hardy et al. [83]:

Lemma B.4.1. Let $\mu^i \geq 0$ be weights such that $\sum_i \mu^i = 1$, and let a_{ik} be positive numbers. Then

$$\sum_k \Big[\prod_i a_{ik}^{\mu^i}\Big] \;\leq\; \prod_i \Big[\sum_k a_{ik}\Big]^{\mu^i} \tag{B.2}$$

The inequality is strict as long as the quantities $[a_{ik}/\sum_k a_{ik}]$ are not all equal.

We now use Lemma B.4.1 to write:

$$\begin{aligned}
\sum_k \Big[\prod_i \Big(\mathbb{E}_{\theta^i}[f^k]\Big)^{\mu^i}\Big] &\leq \prod_i \Big[\sum_k \mathbb{E}_{\theta^i}[f^k]\Big]^{\mu^i} \\
&= \prod_i \Big(\mathbb{E}_{\theta^i}[f]\Big)^{\mu^i}
\end{aligned}$$

where the final line follows from linearity of expectation applied to $\sum_k f^k = f$. This inequality is strict as long as the quantities $\{\mathbb{E}_{\theta^i}[f^k]/\mathbb{E}_{\theta^i}[f]\}$ are not all equal.

Appendix C

Proofs for Chapter 5

■ C.1 Proof of Proposition 5.3.1

We begin by proving equation (5.23a) using induction on the iteration n. The statement is true for $n = 0$, since

$$M_{st}^0 \frac{1}{T_t^0} \sum_{x_s} T_{st}^0 = \kappa \sum_{x_s} \psi_{st}\psi_s \prod_{u \in \mathcal{N}(s)/t} M_{us}^0$$

which is equal to M_{st}^1 using equation (5.3).

Now let us assume that it holds for n and prove it for $n+1$. It is equivalent to prove that $M_{st}^{n+1} = M_{st}^n \frac{1}{T_t^n} \sum_{x_s} T_{st}^n$. Using the definition of T_{st}^n in equation (5.22b), we write

$$M_{st}^n \frac{1}{T_t^n} \sum_{x_s} T_{st}^n = \kappa M_{st}^n \frac{1}{T_t^n} \sum_{x_s} \frac{T_{st}^{n-1}}{(\sum_{x_s} T_{st}^{n-1})(\sum_{x_t} T_{st}^{n-1})} T_s^n T_t^n \quad \text{(C.1a)}$$

$$= \kappa M_{st}^n \sum_{x_s} \frac{T_{st}^{n-1}}{(\sum_{x_s} T_{st}^{n-1})(\sum_{x_t} T_{st}^{n-1})} T_s^n \quad \text{(C.1b)}$$

$$= \kappa M_{st}^n \sum_{x_s} \frac{T_{st}^{n-1}}{(\sum_{x_s} T_{st}^{n-1})(\sum_{x_t} T_{st}^{n-1})} \Big\{ T_s^{n-1} \prod_{u \in \mathcal{N}(s)} \frac{1}{T_s^{n-1}} \sum_{x_u} T_{su}^{n-1} \Big\} \quad \text{(C.1c)}$$

$$= \kappa M_{st}^n \sum_{x_s} \frac{T_{st}^{n-1}}{(\sum_{x_s} T_{st}^{n-1})} \prod_{u \in \mathcal{N}(s)/t} \frac{1}{T_s^{n-1}} \sum_{x_u} T_{su}^{n-1} \quad \text{(C.1d)}$$

$$= \kappa \frac{M_{st}^n T_t^{n-1}}{(\sum_{x_s} T_{st}^{n-1})} \frac{1}{T_t^{n-1}} \sum_{x_s} T_{st}^{n-1} \prod_{u \in \mathcal{N}(s)/t} \frac{M_{us}^n}{M_{us}^{n-1}} \quad \text{(C.1e)}$$

$$= \kappa M_{st}^{n-1} \Big\{ \frac{1}{T_t^{n-1}} \sum_{x_s} T_{st}^{n-1} \prod_{u \in \mathcal{N}(s)/t} \frac{M_{us}^n}{M_{us}^{n-1}} \Big\} \quad \text{(C.1f)}$$

where we have used the definition of T_{st}^n in equation (5.22b) to obtain equation (C.1a); the definition of T_s^n in equation (5.22a) to obtain equation (C.1c); and the induction hypothesis to go to equation (C.1e), and again to equation (C.1f).

Examining the form of equation (C.1f), we see that we can apply the same sequence of steps to the term $\frac{1}{T_t^{n-1}} \sum_{x_s} T_{st}^{n-1} \frac{M_{us}^n}{M_{us}^{n-1}}$ to obtain:

$$\begin{aligned} \kappa M_{st}^{n-1} \Big\{ \frac{1}{T_t^{n-1}} \sum_{x_s} T_{st}^{n-1} \prod_{u \in \mathcal{N}(s)/t} \frac{M_{us}^n}{M_{us}^{n-1}} \Big\} &= \kappa M_{st}^{n-2} \Big\{ \frac{1}{T_t^{n-2}} \sum_{x_s} T_{st}^{n-2} \prod_{u \in \mathcal{N}(s)/t} \frac{M_{us}^{n-1}}{M_{us}^{n-2}} \prod_{u \in \mathcal{N}(s)/t} \frac{M_{us}^n}{M_{us}^{n-1}} \Big\} \\ &= \kappa M_{st}^{n-2} \Big\{ \frac{1}{T_t^{n-2}} \sum_{x_s} T_{st}^{n-2} \prod_{u \in \mathcal{N}(s)/t} \frac{M_{us}^n}{M_{us}^{n-2}} \Big\} \end{aligned}$$

The series telescopes in this multiplicative way until we reach $n = 0$, at which point the right hand side is equal to:

$$\begin{aligned} \kappa M_{st}^0 \Big\{ \frac{1}{T_t^0} \sum_{x_s} T_{st}^0 \prod_{u \in \mathcal{N}(s)/t} \frac{M_{us}^n}{M_{us}^{n-2}} \Big\} &= \kappa M_{st}^0 \sum_{x_s} \psi_{st} \psi_s \prod_{u \in \mathcal{N}(s)/t} M_{us}^n \\ &= M_{st}^{n+1} \end{aligned}$$

where we have used the initialization of T_t^0 and T_{st}^0 given in equations (5.21a) and (5.21b) respectively.

To establish equation (5.23b), we begin by using the definition of B_s^n in equation (5.4) to write

$$\begin{aligned} \frac{B_s^{n+1}}{B_s^n} &= \kappa \prod_{t \in \mathcal{N}(s)} \frac{M_{ts}^{n+1}}{M_{ts}^n} \\ &= \kappa \prod_{t \in \mathcal{N}(s)} \frac{1}{T_s^n} \sum_{x_t} T_{st}^n \\ &= \kappa \frac{T_s^{n+1}}{T_s^n} \end{aligned}$$

This equality, in conjunction with the fact that $B_s^0 = \kappa \psi_s \prod_{u \in \mathcal{N}(s)} M_{us}^0 = T_s^0$, shows that $B_s^n = T_s^n$ for all iterations n.

■ C.2 Proof of Proposition 5.4.1

We begin with some preliminary definitions and lemmas. For a closed and convex set X, we say that x^* is an *algebraic interior point* [30] if for all $x \neq x^*$ in X there exists $x' \in X$, and $\lambda \in (0,1)$ such that $x^* = \lambda x + (1-\lambda)x'$. Otherwise, x is an exterior point. The following lemma characterizes the nature of a constrained local minimum over X.

Lemma C.2.1. Let $f : X \to \mathbb{R}$ be a C^1 function, where X is a closed, convex and nonempty set. Suppose that x^* is a local minimum of f over X. Then

$$\nabla f(x^*)^T (x - x^*) \geq 0$$

for all $x \in X$. Moreover, if x^* is an algebraic interior point, then $\nabla f(x^*)^T (x - x^*) = 0$ for all $x \in X$.

Proof. See [20] for a proof of the first statement. To prove the second statement, assume that x^* is an algebraic interior point, so that for an arbitrary $x \in X$ we can write $x^* = \lambda x + (1-\lambda)x'$ for some x' and $\lambda \in (0,1)$. Then:

$$\begin{aligned} \nabla f(x^*)^T(x^* - x') &= \lambda \nabla f(x^*)^T(x - x') \geq 0 \\ \nabla f(x^*)^T(x^* - x) &= (1-\lambda)\nabla f(x^*)^T(x' - x) \geq 0 \end{aligned}$$

Since $\lambda \in (0,1)$, this establishes that $\nabla f(x^*)^T(x' - x) = 0$, and hence also (from the definition of x^* that $\nabla f(x^*)^T(x - x^*) = 0$. □

Lemma C.2.2. Let $U \in \mathbb{C}^i$ be arbitrary. Then for any θ such that $\widetilde{\theta} = \underline{\mathcal{R}}^i(\theta)$ is bounded:

$$\sum_{\alpha \in \mathcal{A}^i} \left\{ U_\alpha - \Lambda^i(\Pi^i(\theta))_\alpha \right\} [\widetilde{\theta} - \theta]_\alpha = 0 \tag{C.2}$$

Proof. We established in Section 5.4.1 that the point $\Lambda^i(\Pi^i(\theta))$ is the minimizing argument of the function G^i defined in equation (5.25) over the linear and hence convex set $\mathbb{C}^i$. This point will be an exterior point only if some element is equal to zero or one, a possibility that is prevented by the assumption that $\underline{\mathcal{R}}^i(\theta) = \mathcal{I}^i(\Theta^i(\Lambda^i(\Pi^i(\theta))))$ is bounded. Therefore, $\Lambda^i(\Pi^i(\theta))$ is an algebraic interior point, meaning that we can apply Lemma C.2.1 to conclude that for all $U \in \mathbb{C}^i$, we have

$$\sum_{\alpha \in \mathcal{A}^i} \left\{ U_\alpha - \Lambda^i(\Pi^i(\theta))_\alpha \right\} \frac{\partial G^i}{\partial T_\alpha}(\Lambda^i(\Pi^i(\theta)); \theta) = 0 \tag{C.3}$$

It remains to calculate the necessary partial derivatives of G^i. We begin with the decomposition $G^i(T;\theta) = \sum_{(s,t)\in\mathcal{E}^i} G^i_{st}(T_{st};\theta_{st}) + \sum_{s\in\mathcal{V}} G^i_s(T_s;\theta_s)$ where

$$G^i_{st}(T_{st}) = \sum_{j,k} T_{st;jk} \left\{ \log \left[T_{st;jk} / (\sum_j T_{st;jk})(\sum_k T_{st;jk}) \right] - \theta_{st;jk} \right\} \tag{C.4a}$$

$$G^i_s(T_s) = \sum_j T_{s;j} \left[\log T_{s;j} - \theta_{s;j} \right] \tag{C.4b}$$

Using this decomposition, we calculate:

$$\frac{\partial G^i}{\partial T_\alpha}(T;\theta) = \begin{cases} \Theta(T)_{s;j} - \theta_{s;j} + 1 & \text{for } \alpha = (s;j) \\ \Theta(T)_{st;jk} - \theta_{st;jk} - 1 & \text{for } \alpha = (st;jk) \end{cases}$$

Substituting these quantities into equation (C.3), evaluated at $\widetilde{T} = \mathcal{I}^i(\Lambda^i(\Pi^i(\theta)))$, we obtain:

$$\sum_{s\in\mathcal{V}} \sum_j \{U - \widetilde{T}\}_{s;j} [\widetilde{\theta} - \theta + 1]_{s;j} + \sum_{(s,t)\in\mathcal{E}^i} \sum_{j,k} \{U - \widetilde{T}\}_{st;jk} [\widetilde{\theta} - \theta - 1]_{st;jk} = 0 \tag{C.5}$$

where by definition $\widetilde{\theta}_\alpha = \Theta^i(\widetilde{T})_\alpha$ for each $\alpha \in \mathcal{A}^i$. Now since both U and $\widetilde{T}$ belong to $\mathbb{C}^i$, we have $\sum_j [U - \Lambda(\mathcal{Q}^i(\theta))]_{s;j} = 0$ for all $s \in \mathcal{V}$ and $\sum_{j,k} [U - \widetilde{T}]_{st;jk} = 0$ for all $(s,t) \in \mathcal{E}^i$. As a result, the constants 1 or -1 in equation (C.5) vanish in the sums over j or $\{j,k\}$, and we are left with the desired statement in equation (C.2). □

Equipped with these lemmas, we now establish equation (5.27) for $\lambda^n = 1$ by writing:

$$\begin{aligned} G(U;\theta) - G(U;\mathcal{Q}^i(\theta)) - G(\underline{\Lambda}^i(\mathcal{Q}^i(\theta));\theta) &= \sum_{\alpha \in \mathcal{A}} \left[U - \underline{\Lambda}^i(\mathcal{Q}^i(\theta))\right]_\alpha \left[\mathcal{Q}^i(\theta) - \theta\right]_\alpha \\ &= \sum_{\alpha \in \mathcal{A}^i} \left[U - \Lambda^i(\mathcal{Q}^i(\theta))\right]_\alpha \left[\Theta(\Lambda^i(\Pi^i(\theta))) - \theta\right]_\alpha \end{aligned}$$

where we used the fact that $\mathcal{Q}^i(\theta)_\alpha = \begin{cases} \theta_\alpha & \text{for all } \alpha \in \mathcal{A}/\mathcal{A}^i \\ \underline{\mathcal{R}}^i(\theta)_\alpha & \text{for all } \alpha \in \mathcal{A}^i. \end{cases}$

Since $\underline{\mathcal{R}}^i(\theta)$ is bounded by assumption, we can apply Lemma C.2.2 to conclude that

$$G(U;\theta) - G(U;\mathcal{Q}^i(\theta)) - G(\underline{\Lambda}^i(\mathcal{Q}^i(\theta));\theta) = 0 \tag{C.6}$$

thereby establishing equation (5.27) for $\lambda^n = 1$, with the identifications $\theta \equiv \theta^n$ and $i = i(n)$.

To extend the result to $\lambda^n \in [0,1]$, we use the definition of θ^{n+1} given in equation (5.20) to write:

$$\begin{aligned} G(U;\theta^n) - G(U;\theta^{n+1}) &= \sum_\alpha U_\alpha \left[\theta^{n+1} - \theta^n\right]_\alpha \\ &= \lambda^n \sum_\alpha U_\alpha \left[\mathcal{Q}^{i(n)}(\theta^n) - \theta^n\right]_\alpha \\ &= \lambda^n \Big\{ G(U;\theta^n) - G(U;\mathcal{Q}^{i(n)}(\theta^n)) \Big\} \\ &= \lambda^n G(\underline{\Lambda}^{i(n)}(\mathcal{Q}^{i(n)}(\theta^n));\theta^n) \end{aligned}$$

where we have obtained the final line using equation (C.6).

■ C.3 Proof of Theorem 5.4.1

(a): By the cyclic tree ordering, we have $\theta^{Ln+i+1} - \theta^{Ln+i} = \lambda^{Ln+i}[\mathcal{Q}^i(\theta^{Ln+i}) - \theta^{Ln+i}]$ where i is arbitrary in $\{0,\ldots,L-1\}$. Since the sequence $\{\theta^n\}$ converges to θ^*, it is Cauchy so that the left side tends to zero. Since $\lambda^{Ln+i} \geq \epsilon$, this implies that $\mathcal{Q}^i(\theta^{Ln+i}) - \theta^{Ln+i} \to 0$. I.e. $\mathcal{Q}^i(\theta^*) = \theta^*$ for all $i \in \{0,\ldots,L-1\}$.

We now construct the unique $T^* \in \mathbb{C}$ such that $\Pi^i(T^*) = \Lambda^i(\Pi^i(\theta^*))$. For an arbitrary index $\alpha \in \mathcal{A}$, pick a spanning tree $\mathcal{T}^i$ such that $\alpha \in \mathcal{A}^i$. This is always possible since $\cup_i \mathcal{A}^i = \mathcal{A}$ by construction. Define $T^*_\alpha = [\Lambda^i(\Pi^i(\theta^*))]_\alpha$, which is a consistent definition because

$$[\Lambda^i(\Pi^i(\theta^*))]_\alpha = [\Lambda^j(\Pi^j(\theta^*))]_\alpha$$

for any spanning tree indices i, j such that $\alpha \in \mathcal{A}^i \cap \mathcal{A}^j$. By construction, it is clear that $T^* \in \mathbb{C}$, and that $\Pi^i(T^*) = \Lambda^i(\Pi^i(\theta^*))$.

(b): Let $U \in \mathbb{C} = \cap_i \mathbb{C}^i$ be arbitrary. By applying Proposition 5.4.1 repeatedly, we obtain

$$G(U;\theta^0) = G(U;\theta^*) + \sum_{n=1}^{\infty} W_n \tag{C.7}$$

where $W_n \triangleq \lambda^n G(\underline{\Lambda}^{i(n)}(\mathcal{Q}^{i(n)}(\theta^n)); \theta^n)$. By part (a), the parameter θ^* induces a unique pseudomarginal vector $T^* \in \cap_i \mathbb{C}^i$. We then apply equation (C.7) with $U = T^*$ and use the fact that $G(T^*;\theta^*) = 0$ by construction to obtain $G(T^*;\theta^0) = \sum_{n=1}^{\infty} W_n$. Substituting this result back into equation (C.7), we find that

$$G(U;\theta^0) = G(U;\theta^*) + G(T^*;\theta^0)$$

for all $U \in \cap_i \mathbb{C}^i$. To prove that T^* satisfies the necessary conditions to be a local minimum, we note that

$$\begin{aligned} G(U;\theta^0) - G(U;\theta^*) - G(T^*;\theta^0) &= \sum_{\alpha} \frac{\partial G}{\partial T_\alpha}(T^*;\theta^0)\big[U - T^*\big]_\alpha \\ &= 0 \end{aligned}$$

where we have used a sequence of steps similar to the proof of Proposition 5.4.1.

(c) Since the cost function G is bounded below and the constraint set is non-empty, the problem has at least one minimum. Moreover, because the constraint sets are linear, the existence of Lagrange multipliers is guaranteed for any local minimum [20]. By applying Farkas' lemma [20], the condition stated in (b) must be satisfied by any local minimum.

(d) Part (b) establishes that any fixed point T^* satisfies the necessary conditions to be a local minimum of G over the constraint set $\mathbb{C}$. The cost function G agrees with the Bethe free energy on this constraint set. Moreover, Yedidia et al. [180] have shown that BP fixed points correspond the points that satisfy the Lagrangian conditions for an extremum of the Bethe free energy over $\mathbb{C}$. By recourse to Farkas' lemma [20], these Lagrangian conditions are equivalent to the condition stated in (b). Therefore, we conclude that fixed points of the two algorithms coincide.

■ C.4 Proof of Theorem 5.4.2

Throughout this appendix, we will use the notation $\Phi^i(\theta)$ as a shorthand for the quantity $\Phi(\Pi^i(\theta))$. With this notation, we begin with some preliminary lemmas.

Lemma C.4.1. For all indices $i = 0, \ldots, L-1$, we have $G^i(\Lambda^i(\Pi^i(\theta)); \Pi^i(\theta)) = -\Phi^i(\theta)$.

Proof. Note that $\Theta^i(\Lambda^i(\Pi^i(\theta)))$ and $\Pi^i(\theta)$ induce the same distribution on spanning tree $\mathcal{T}^i$ so that $D(\Theta(\Lambda^i(\Pi^i(\theta))) \parallel \Pi^i(\theta)) = 0$. The statement of the lemma then follows from equation (5.26). □

Lemma C.4.2. Let $l \in \{0, \ldots, L-1\}$ be arbitrary, and let $i(n)$ the tree index used at the n^{th} iteration. Then:

$$\Phi^l(\theta^{n+1}) \leq (1-\lambda^n)\Phi^l(\theta^n) + \lambda^n \Phi^l(\mathcal{Q}^{i(n)}(\theta^n)) \tag{C.8}$$

Moreover, in the special case $l = i(n)$, we have

$$\Phi^{i(n)}(\theta^{n+1}) = (1-\lambda^n)\Phi^{i(n)}(\theta^n) \tag{C.9}$$

Proof. Recall that θ^{n+1} is formed as the convex combination

$$\theta^{n+1} = (1-\lambda^n)\theta^n + \lambda^n \mathcal{Q}^{i(n)}(\theta^n)$$

This combination remains convex if we apply the linear projection operator Π^l to both sides, so that equation (C.8) follows from the well-known convexity of the log partition function Φ.

In the special case $l = i(n)$, we have $\Phi^l(\mathcal{Q}^l(\theta^n)) = 0$, so that equation (C.8) reduces to $\Phi^l(\theta^{n+1}) \leq (1-\lambda^n)\Phi^l(\theta^n)$. Moreover, by the convexity of Φ:

$$\begin{aligned}
\Phi^l(\theta^{n+1}) &\geq \Phi^l(\theta^n) + \sum_{\alpha \in \mathcal{A}^l} \mathbb{E}_{\Pi^l(\theta^n)}[\phi_\alpha]\left[\theta^{n+1} - \theta^n\right]_\alpha && \text{(C.10a)} \\
&= \Phi^l(\theta^n) + \lambda^n \sum_{\alpha \in \mathcal{A}^l} \mathbb{E}_{\Pi^l(\theta^n)}[\phi_\alpha]\left[\mathcal{Q}^l(\theta^n) - \theta^n\right]_\alpha && \text{(C.10b)} \\
&= (1-\lambda^n)\Phi^l(\theta^n) && \text{(C.10c)}
\end{aligned}$$

where we have used the fact that $\frac{\partial \Phi}{\partial \theta_\alpha}(\Pi^l(\theta)) = \mathbb{E}_{\Pi^l(\theta)}[\phi_\alpha]$ to obtain equation (C.10a); the definition of θ^{n+1} in equation (C.10b); and Lemma C.4.1 in equation (C.10c). □

With these preliminary lemmas, we can begin the proof of the theorem. Let $U \in \mathbb{C}$ be arbitrary. By applying Proposition 5.4.1 repeatedly, for any iteration $M = 1, 2, \ldots$, we obtain

$$G(U; \theta^0) - G(U; \theta^M) = \sum_{n=0}^{M-1} W_n \tag{C.11}$$

where $W_n = \lambda^n G(\underline{\Lambda}^{i(n)}(\mathcal{Q}^{i(n)}(\theta^n)); \theta^n)$. Lemma C.4.1 and the definition of $\underline{\Lambda}^i$ in equation (5.17b) lead to the equivalent expressions:

$$\begin{aligned}
\Phi^i(\theta^n) &= G^i(\Lambda^i(\Pi^i(\theta^n)); \Pi^i(\theta^n)) && \text{(C.12a)} \\
&= G(\underline{\Lambda}^i(\mathcal{Q}^i(\theta^n)); \theta^n) && \text{(C.12b)}
\end{aligned}$$

meaning that we can write $W_n = -\lambda^n \Phi^{i(n)}(\theta^n)$.

From this point onwards, our goal is to establish that

$$\lim_{n\to\infty} \Phi^i(\theta^n) = 0 \quad \text{for } i = 0, 1 \tag{C.13}$$

Indeed, if equation (C.13) holds, then using equation (C.12), we see that assumption (a) implies that $\mathcal{Q}^i(\theta^n) \to \theta^n$ as well, which is the statement of the theorem. The essence of establishing (C.13) is to choose a sequence $\{\lambda^n\}$ of positive step sizes such that $W_n > 0$ is guaranteed for all $n = 0, 1, 2, \ldots$. This condition ensures that for some fixed $U \in \mathbb{C}$, the LHS of equation (C.11) — namely, the sequence $A_M \triangleq G(U;\theta^0) - G(U;\theta^M)$ — is non-decreasing in M. Moreover, since the sequence $\{\theta^M\}$ remains bounded by assumption, the sequence A_M is also bounded above. That is, the sequence $\{A_M\}$ is both non-decreasing and bounded above, and so must converge. Using equation (C.11), the convergence of $\{A_M\}$ will allow us to conclude that $W_n \to 0$. Finally, we will use this fact to establish equation (C.13).

Without loss of generality, we assume that $\Phi^i(\theta^0) < 0$ for $i = 0, 1$, a condition that can be guaranteed by subtracting a constant from the full vector θ^0 if necessary. We formalize the step size choice in the following lemma:

Lemma C.4.3. At each iteration n, define:

$$\mu^n \triangleq \begin{cases} \left[\frac{-\Phi^i(\theta^n)}{\Phi^i(\mathcal{Q}^{i(n)}(\theta^n)) - \Phi^i(\theta^n)}\right] & \text{if } \Phi^i(\mathcal{Q}^{i(n)}(\theta^n)) > 0 \\ 1/(n+1) & \text{otherwise} \end{cases}$$

where $i(n) \equiv n (\mathrm{mod}\, 2)$ and $i \equiv (n+1) (\mathrm{mod}\, 2)$. Provided that $\Phi^i(\theta^0) < 0$ for $i = 0, 1$, then choosing the step sizes

$$\lambda^n = \frac{1}{2}\mu^n \tag{C.14}$$

will guarantee that $\Phi^i(\theta^n) < 0$ for all n, $i = 0, 1$.

Proof. The proof is by induction; the case $n = 0$ is given, and so we assume it holds for an even iteration n so that $i(n) = 0$. From equation (C.9) in Lemma C.4.2, if $\Phi^0(\theta^n) < 0$, then any step size in $(0, 1)$ will ensure that $\Phi^0(\theta^{n+1}) < 0$. Note that by construction $0 < \lambda^n < 1$, so that it is a valid step size choice.

Now considering $i = 1$: if $\Phi^1(\mathcal{Q}^0(\theta^n)) \leq 0$, then again any choice $\lambda^{n+1} < 1$ will suffice. On the other hand, if $\Phi^1(\mathcal{Q}^0(\theta^n)) > 0$, with the step size

$$0 < \mu^n = \frac{-\Phi^1(\theta^n)}{\Phi^1(\mathcal{Q}^0(\theta^n)) - \Phi^1(\theta^n)} < 1$$

the right hand side (i.e., upper bound) of equation (C.8) is zero. Since the upper bound of equation (C.8) decreases for smaller λ^n, the step size choice of equation (C.14) will ensure that $\Phi^1(\theta^{n+1}) < 0$.

A similar argument can be applied for odd n, where $i(n) = 1$. Therefore, we have established that our step size choice ensures that $\Phi^i(\theta^{n+1}) < 0$ for all n, and $i = 0, 1$. □

We now prove equation (C.13). By the step size choice of Lemma C.4.3 and our earlier reasoning, we are guaranteed that the infinite sum $\sum_{n=0}^{\infty} W_n$ exists, and that $W_n \to 0^+$. So as to exploit assumption (b) of the theorem statement, we now split our analysis into two cases. Note that for $a, b = 0, 1$, we have $\Phi^a(\mathcal{Q}^b(\theta^n)) = G(\underline{\Lambda}^a(\mathcal{Q}^b(\theta^n)); \mathcal{Q}^b(\theta^n))$ by definition of $\underline{\Lambda}^a$ in equation (5.17b). Therefore, assumption (b) means that the quantities $\Phi^1(\mathcal{Q}^0(\theta^n))$ and $\Phi^0(\mathcal{Q}^1(\theta^n))$ are eventually (i.e., for $n \geq K$) of the same sign.

Case 1: Suppose first that for $a, b = 0, 1$, we have $\Phi^a(\mathcal{Q}^b(\theta^n)) \leq 0$ for all $n \geq K$. This implies that $\lambda^n = \frac{1}{2(n+1)}$ for all $n \geq K$, so that the infinite sum $\sum_{n \geq K} W_n = -\sum_{n \geq K} \Phi^{i(n)}(\theta^n)/[2\,(n+1)]$ exists. Since $-\Phi^{i(n)}(\theta^n) > 0$ for all n by construction, this implies that $\Phi^i(\theta^n) \to 0$ for $i = 0, 1$.

Case 2: Otherwise for $a, b = 0, 1$ and $a \neq b$,[1] we have $\Phi^a(\mathcal{Q}^b(\theta^n)) > 0$ for all $n \geq K$. Let $\{n_k\}$ be the even integers for which $i(n_k) = 0$. Then we have:

$$\begin{aligned} \sum_{n_k \geq K}^{\infty} W_{n_k} &= -\sum_{n_k \geq K}^{\infty} \lambda^{n_k} \Phi^0(\theta^{n_k}) \\ &= \frac{1}{2} \sum_{n_k \geq K} \frac{\Phi^0(\theta^{n_k}) \Phi^1(\theta^{n_k})}{\Phi^1(\mathcal{Q}^0(\theta^{n_k})) - \Phi^1(\theta^{n_k})} \end{aligned} \tag{C.15}$$

Since the sequence $\{\theta^n\}$ remains bounded by assumption, the denominator of W_{n_k} is bounded in absolute value. Therefore, the fact that $W_{n_k} \to 0$ implies that the numerator — namely, $\Phi^0(\theta^{n_k})\Phi^1(\theta^{n_k})$ — must converge to zero. This condition does not necessarily imply that one of these two log partition functions converges to zero; for example, we could have $\Phi^0(\theta^{n_k})$ tending to zero for even k, and $\Phi^1(\theta^{n_k})$ tending to zero for odd k.

With the additional constraints of our problem, we shall now prove that, in fact we have $\lim_{n_k \to \infty} \Phi^1(\theta^{n_k}) = 0$. We proceed via proof by contradiction: if this were not the case, then there would exist some infinite subsequence (say $\{n_j\}$) of the even indices $\{n_k\}$ such that $\Phi^1(\theta^{n_j})$ is bounded away from zero. From the condition $\Phi^1(\theta^{n_j})\Phi^0(\theta^{n_j}) \to 0$, this implies that $\Phi^0(\theta^{n_j}) \to 0$. By assumption (a) and the equivalence of Lemma C.4.1, this implies that $[\mathcal{Q}^0(\theta^{n_j}) - \theta^{n_j}] \to 0$. Since Φ^1 is a C^2 function and $\{\theta^n\}$ is bounded, we can apply the mean value theorem to conclude that $\lim_{n_j \to \infty} \inf \left[\Phi^1(\theta^{n_j}) - \Phi^1(\mathcal{Q}^0(\theta^{n_j})\right] = 0$. Moreover, since $\Phi^1(\mathcal{Q}^0(\theta^{n_j})) > 0$ for all $n_j \geq K$ by assumption, we have

$$\begin{aligned} \lim_{n_j \to \infty} \inf \Phi^1(\theta^{n_j}) &\geq \lim_{n_j \to \infty} \inf \left[\Phi^1(\theta^{n_j}) - \Phi^1(\mathcal{Q}^0(\theta^{n_j}))\right] \\ &= 0 \end{aligned}$$

Moreover, by our step size choice, we have $\Phi^1(\theta^{n_j}) < 0$ for all n_j, thereby ensuring the relation $\lim_{n_j \to \infty} \sup \Phi^1(\theta^{n_j}) \leq 0$. In conjunction, these two relations imply that $\lim_{n_j \to \infty} \Phi^1(\theta^{n_j})$ exists, and is equal to zero, so that we have reached a contradic-

[1]Note that by definition, $\Phi^a(\mathcal{Q}^a(\theta^n)) = 0$ for $a = 0, 1$.

tion. Therefore, our initial assumption must have been false, and we can conclude that $\lim_{n_k \to \infty} \Phi^1(\theta^{n_k}) = 0$.

On the other hand, to analyze the behavior of $\Phi^0(\theta^n)$, consider the sequence formed by the odd indices $\{m_k\}$. This leads to equation (C.15), with the roles of 0 and 1 interchanged. Thus, a similar argument allows us to establish that $\Phi^0(\theta^{m_k}) \to 0$.

Therefore, we have proved that $\Phi^1(\theta^{n_k}) \to 0$ and $\Phi^0(\theta^{m_k}) \to 0$. These conditions in conjunction imply that the step sizes λ^n are tending to zero for the infinite subsequences formed by even indices $\{n_k\}$ and odd indices $\{m_k\}$. Therefore, we can conclude that the overall sequence θ^n converges to some θ^* such that $\Phi^i(\theta^*) = 0$, and hence equation (C.13) is proved, which establishes the theorem.

■ C.5 Proof of Proposition 5.4.2

We begin by expressing the delta function $\delta(x_s = j)$ as a linear combination of the monomials in the set $\mathcal{R}(s)$ defined in equation (2.12a) as follows:

$$\delta(x_s = j) \quad = \quad \prod_{k \neq j} \frac{(k - x_s)}{(k - j)} \tag{C.16}$$

This decomposition is extended readily to pairwise delta functions, which are defined by products $\delta(x_s = j)\delta(x_t = k)$; in particular, they can be written as linear combinations of elements in the sets $\mathcal{R}(s)$ and $\mathcal{R}(s,t)$, as defined in equation (2.12a) and equation (2.12b), respectively. Now suppose that $\theta \in \mathcal{M}(\theta^0)$, so that $\log p(\mathbf{x}; \gamma^0) = \log p(\mathbf{x}; \theta)$ for all $x \in \mathcal{X}$. By construction, both the LHS and RHS are linear combination of the elements $\mathcal{R}(s)$ and $\mathcal{R}(s,t)$. Equating the coefficients of these terms yields a set of $d(\gamma) = (m-1)N + (m-1)^2|\mathcal{E}|$ linear equations. We write these equations compactly in matrix form as $A\theta = \gamma^0$.

This establishes the necessity of the linear manifold constraints. To establish their sufficiency, we need only check that the linear constraints ensure the constant terms in $\log p(\mathbf{x}; \gamma^0)$ and $\log p(\mathbf{x}; \theta)$ (i.e., $\Phi(\gamma^0)$ and $\Phi(\theta)$ respectively) are also equal. This is a straightforward verification.

Appendix D

Proofs for Chapter 7

■ D.1 Proof of Proposition 7.2.2

From Lemma 7.2.1, for $\|\theta^*\| < \infty$, the optimal $\widehat{\lambda}$ is attained in the interior of $\mathbb{L}(\mathcal{G})$. I.e., none of the linear inequality constraints defining $\mathbb{L}(\mathcal{G})$ are met with equality. Optimal points $\widehat{\lambda}$ are therefore given as zero points of the gradient $\frac{\partial \mathcal{Q}(\lambda;\bar{\mu};\theta^*)}{\partial \lambda}$. In order to calculate this gradient, we use the fact that $\frac{\partial \Psi(\Pi^{\mathcal{T}}(\lambda))}{\partial \lambda_\alpha} = \theta(\mathcal{T})_\alpha$ by definition of the Legendre duality coupling the log partition function Φ and negative entropy Ψ. Calculating the gradient $\frac{\partial \mathcal{Q}(\lambda;\bar{\mu};\theta^*)}{\partial \lambda}$ and setting it to zero yields the following stationary conditions for the optimum:

$$\mathbb{E}_{\bar{\mu}}[\widehat{\theta}(\mathcal{T})_{st}] = \theta^*_{st} \tag{D.1a}$$

$$\mathbb{E}_{\bar{\mu}}[\widehat{\theta}(\mathcal{T})_{s}] = \theta^*_{s} \tag{D.1b}$$

Now for any spanning tree $\mathcal{T} \in \mathfrak{T}$, we have $p(\mathbf{x};\widehat{\theta}(\mathcal{T})) = p(\mathbf{x};\Pi^{\mathcal{T}}(\widehat{\lambda}))$. By definition, the distribution $p(\mathbf{x};\widehat{\theta}(\mathcal{T}))$ has the following exponential form:

$$\log p(\mathbf{x};\widehat{\theta}(\mathcal{T})) = \sum_{s\in\mathcal{V}} \widehat{\theta}(\mathcal{T})_s\, x_s + \sum_{(s,t)\in\mathcal{E}(\mathcal{T})} \widehat{\theta}(\mathcal{T})_{st}\, x_s x_t \; - \Phi(\widehat{\theta}(\mathcal{T})) \tag{D.2}$$

On the other hand, from equation (7.15), we have:

$$\begin{aligned}\log p(\mathbf{x};\Pi^{\mathcal{T}}(\widehat{\lambda})) = &\sum_{s\in\mathcal{V}}\sum_{j=0}^{1} \log p(x_s = j;\widehat{\lambda})\delta(x_s = j) \\ &+ \sum_{(s,t)\in\mathcal{E}(\mathcal{T})}\sum_{j,k=0}^{1} \log\left[\frac{p(x_s = j, x_t = k;\widehat{\lambda})}{p(x_s = j;\widehat{\lambda})\; p(x_t = k;\widehat{\lambda})}\right]\delta(x_s = j)\delta(x_t = k)\end{aligned} \tag{D.3}$$

where $p(x_s;\widehat{\lambda})$ and $p(x_s, x_t;\widehat{\lambda})$ are defined in equation (7.12).

Using the fact that for binary variables $\delta(x_s = 0) = (1 - x_s)$ (and similarly, $\delta(x_s = 1) = x_s$), we see that equations (D.2) and (D.3) are both binomials in $\{x_s\}$

and $\{x_s x_t\}$. Equating their respective coefficients yields the following relations:

$$\widehat{\theta}(\mathcal{T})_{st} = \delta[(s,t) \in \mathcal{T}] \log\left[\frac{(\widehat{\lambda}_{st})\ (1+\widehat{\lambda}_{st}-\widehat{\lambda}_s-\widehat{\lambda}_t)}{(\widehat{\lambda}_s-\widehat{\lambda}_{st})(\widehat{\lambda}_t-\widehat{\lambda}_{st})}\right] \tag{D.4a}$$

$$\widehat{\theta}(\mathcal{T})_s = \log\left[\frac{\widehat{\lambda}_s}{(1-\widehat{\lambda}_s)}\right] + \sum_{t\in\mathcal{N}(s)} \delta[(s,t) \in \mathcal{T}] \log\left[\frac{(\widehat{\lambda}_s-\widehat{\lambda}_{st})}{(1+\widehat{\lambda}_{st}-\widehat{\lambda}_s-\widehat{\lambda}_t)}\right] \tag{D.4b}$$

Taking expectations with respect to $\bar{\mu}$ and using equations (D.1a) and (D.1b) yields the statement of the proposition.

□

■ D.2 Proof of Proposition 7.3.1

(a) Define the function

$$\mathcal{H}(\boldsymbol{\mu_e};\theta^*) = \min_{\lambda\in\mathbb{L}(\mathcal{G})} \mathcal{F}(\lambda;\boldsymbol{\mu_e};\theta^*) = \mathcal{F}(\widehat{\lambda}(\boldsymbol{\mu_e});\boldsymbol{\mu_e};\theta^*)$$

where $\widehat{\lambda}(\boldsymbol{\mu_e})$ denotes the optimal λ as a function of $\boldsymbol{\mu_e}$. Taking derivatives via the chain rule, we obtain:

$$\frac{\partial\mathcal{H}}{\partial\mu_{st}}(\boldsymbol{\mu_e};\theta^*) = \sum_\alpha \frac{\partial\mathcal{F}}{\partial\lambda_\alpha}\frac{\partial\widehat{\lambda}_\alpha}{\partial\mu_{st}}\bigg|_{\widehat{\lambda}(\boldsymbol{\mu_e})} + \frac{\partial\mathcal{F}}{\partial\mu_{st}}\bigg|_{\widehat{\lambda}(\boldsymbol{\mu_e})} \tag{D.5}$$

Now from Lemma 7.2.1, the optimum $\widehat{\lambda}(\boldsymbol{\mu_e})$ of the problem $\min_{\lambda\in\mathbb{L}(\mathcal{G})}\mathcal{F}(\lambda;\boldsymbol{\mu_e};\theta^*)$ occurs at an interior point of $\mathbb{L}(\mathcal{G})$. Therefore, none of the constraints defining $\mathbb{L}(\mathcal{G})$ are active, so that in fact, by the Karush-Kuhn-Tucker conditions [20], we must have

$$\frac{\partial\mathcal{F}}{\partial\lambda_\alpha}|_{\widehat{\lambda}(\boldsymbol{\mu_e})} = 0 \tag{D.6}$$

at the optimum $\widehat{\lambda}$. Moreover, straightforward calculations yield

$$\frac{\partial\mathcal{F}}{\partial\mu_{st}}|_{\widehat{\lambda}} = I_{st}(\widehat{\lambda}) \tag{D.7}$$

By combining equations (D.6) and (D.7) with equation (D.5), we are led to conclude that $\frac{\partial\mathcal{H}}{\partial\mu_{st}}(\boldsymbol{\mu_e};\theta^*) = I_{st}(\widehat{\lambda}(\boldsymbol{\mu_e}))$.

Now form the Lagrangian associated with the problem $\max_{\boldsymbol{\mu_e}\in\mathbb{T}(\mathcal{G})}\mathcal{H}(\boldsymbol{\mu_e};\theta^*)$:

$$\mathcal{L}(\boldsymbol{\mu_e};\xi;\theta^*) = \mathcal{H}(\boldsymbol{\mu_e};\theta^*) + \xi_0\Big[(N-1) - \sum_{e\in\mathcal{E}}\mu_e\Big] + \sum_{A\subset\mathcal{E}}\xi(A)\Big[r(A) - \sum_{e\in A}\mu_e\Big]$$

where the sum $\sum_{A \subset \mathcal{E}}$ ranges over all critical subsets A. Taking derivatives with respect to μ_e yields the Lagrangian conditions stated in the proposition. The Karush-Kuhn-Tucker conditions guarantee that the Lagrange multipliers $\xi(A)$ associated with the inequality constraints are all non-negative. In particular, $\xi(A) \geq 0$ with equality whenever the constraint associated with A is inactive.

(b) Since $\frac{\partial \mathcal{H}}{\partial \mu_{st}}(\boldsymbol{\mu_e}; \theta^*) = I_{st}(\widehat{\lambda}(\boldsymbol{\mu_e}))$ from part (a) and $\boldsymbol{\nu}(\mathcal{T}) \in \mathbb{T}(\mathcal{G})$, the statement

$$\langle I(\widehat{\lambda}(\widehat{\boldsymbol{\mu}_e})), \boldsymbol{\nu}(\mathcal{T}) - \widehat{\boldsymbol{\mu}_e} \rangle \leq 0 \qquad \forall \quad \mathcal{T} \in \mathfrak{T} \tag{D.8}$$

follows from standard necessary conditions [see 20] for the maximum $\widehat{\boldsymbol{\mu}_e}$ of $\mathcal{H}$ over the linear (hence convex) set $\mathbb{T}(\mathcal{G})$.

We now establish that inequality (D.8) holds with equality for all $\mathcal{T} \in \text{supp}(\vec{\boldsymbol{\mu}})$. Since $\widehat{\boldsymbol{\mu}_e} \in \mathbb{T}(\mathcal{G})$, there exists some distribution $\widehat{\vec{\boldsymbol{\mu}}}$ over spanning trees that

$$\sum_{\mathcal{T} \in \mathfrak{T}} \widehat{\mu}(\mathcal{T}) \delta[e \in \mathcal{T}] = \widehat{\mu}_e \qquad \forall \quad e \in \mathcal{E} \tag{D.9}$$

We now multiply equation (D.9) by $I_e(\widehat{\lambda}(\boldsymbol{\mu_e}))$ and sum over all $e \in \mathcal{E}$ to obtain

$$\begin{aligned} 0 &= \sum_{e \in \mathcal{E}} I_e(\widehat{\lambda}(\boldsymbol{\mu_e})) \Big[\sum_{\mathcal{T} \in \mathfrak{T}} \widehat{\mu}(\mathcal{T}) \delta[e \in \mathcal{T}] - \widehat{\mu}_e \Big] \\ &= \sum_{\mathcal{T} \in \mathfrak{T}} \widehat{\mu}(\mathcal{T}) \sum_{e \in \mathcal{E}} I_e(\widehat{\lambda}(\boldsymbol{\mu_e})) \Big[\delta[e \in \mathcal{T}] - \widehat{\mu}_e \Big] \\ &= \sum_{\mathcal{T} \in \mathfrak{T}} \widehat{\mu}(\mathcal{T}) \ \langle I(\widehat{\lambda}(\widehat{\boldsymbol{\mu}_e})), \boldsymbol{\nu}(\mathcal{T}) - \widehat{\boldsymbol{\mu}_e} \rangle \end{aligned}$$

where we have recognized that for fixed $\mathcal{T}$, the function $\delta[e \in \mathcal{T}] \equiv \boldsymbol{\nu}(\mathcal{T})_e$. Using this relation and inequality (D.8), we must have $\langle I(\widehat{\lambda}(\widehat{\boldsymbol{\mu}_e})), \boldsymbol{\nu}(\mathcal{T}) - \widehat{\boldsymbol{\mu}_e} \rangle = 0$ for all $\mathcal{T}$ such that $\widehat{\mu}(\mathcal{T}) > 0$.

□

Bibliography

[1] S. Aji and R. McEliece. The generalized distributive law and free energy minimization. In *Allerton conference*, Allerton, IL, 2001. To appear.

[2] S. M. Aji, G. Horn, and R. McEliece. On the convergence of iterative decoding on graphs with a single cycle. In *Proceedings IEEE Intl. Symp. on Information Theory*, page 276, Cambridge, MA, 1998.

[3] S.M. Aji and R.J. McEliece. The generalized distributive law. *IEEE Trans. Info. Theory*, 46:325–343, March 2000.

[4] E.L. Allgower and K. Georg. Homotopy methods for approximating several solutions to nonlinear systems of equations. In W. Forster, editor, *Numerical solution of highly nonlinear problems*, pages 253–270. North-Holland, 1980.

[5] S. Amari. Differential geometry of curved exponential families — curvatures and information loss. *Annals of Statistics*, 10(2):357–385, 1982.

[6] S. Amari. Information geometry on a hierarchy of probability distributions. *IEEE Trans. on Information Theory*, 47(5):1701–1711, 2001.

[7] S. Amari, K. Kurata, and H. Nagaoka. Information geometry of Boltzmann machines. *IEEE Trans. on Neural Networks*, 3(2):260–271, 1992.

[8] J. B. Anderson and S. M. Hladnik. Tailbiting map decoders. *IEEE Sel. Areas Comm.*, 16:297–302, February 1998.

[9] S. Arimoto. An algorithm for computing the capacity of an arbitrary discrete memoryless channel. *IEEE Transactions on Information Theory*, 18:14–20, 1972.

[10] O. Axelsson. Bounds of eigenvalues of preconditioned matrices. *SIAM J. Matrix Anal. Appl.*, 13:847–862, July 1992.

[11] D. Barber and Pierre van der Laar. Variational cumulant expansions for intractable distributions. *Journal of Artificial Intelligence Research*, 10:435–455, 1999.

[12] D. Barber and W. Wiegerinck. Tractable variational structures for approximating graphical models. In *NIPS 11*. MIT Press, 1999.

[13] O. E. Barndorff-Nielson. *Information and exponential families*. Wiley, Chichester, 1978.

[14] M. Basseville, A. Benveniste, K. Chou, S. Golden, R. Nikoukhah, and A. Willsky. Modeling and estimation of multiresolution stochastic processes. *IEEE Transactions on Information Theory*, 38(2):766–784, March 1992.

[15] R. J. Baxter. *Exactly solved models in statistical mechanics*. Academic Press, New York, 1982.

[16] C. Berge. *The theory of graphs and its applications*. Wiley, New York, NY, 1964.

[17] C. Berge. *Graphs and hypergraphs*. North-Holland Publishing Company, Amsterdam, 1976.

[18] C. Berroux, A. Glavieux, and P. Thitmajshima. Near Shannon limit error-correcting coding and decoding. In *Proceedings of ICC*, pages 1064–1070, 1993.

[19] D.P. Bertsekas. *Dynamic programming and stochastic control*, volume 1. Athena Scientific, Belmont, MA, 1995.

[20] D.P. Bertsekas. *Nonlinear programming*. Athena Scientific, Belmont, MA, 1995.

[21] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Roy. Stat. Soc. Series B*, 36:192–236, 1974.

[22] N. Biggs. *Algebraic graph theory*. Cambridge University Press, Cambridge, 1993.

[23] G. Birkhoff and R. S. Varga. Implicit alternating direction methods. *Transactions of the AMS*, 92(1):13–24, July 1959.

[24] R. E. Blahut. Computation of channel capacity and rate distortion functions. *IEEE Transactions on Information Theory*, 18:460–473, 1972.

[25] B. Bollobás. *Graph theory: an introductory course*. Springer-Verlag, New York, 1979.

[26] B. Bollobás. *Modern graph theory*. Springer-Verlag, New York, 1998.

[27] E.G. Boman and B. Hendrickson. Support theory for preconditioning. Submitted for publication, 2001.

[28] P. Brémaud. *Markov chains, Gibbs fields, Monte Carlo simulation, and queues*. Springer, 1991.

[29] E. Castillo, J. M. Gutierrez, and A. S. Hadi. Sensitivity analysis in discrete Bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 27:412–423, 1997.

[30] Y. Censor and S. A. Zenios. *Parallel Optimization: Theory, Algorithms, and Applications.* Numerical Mathematics and Scientific Computation. Oxford University Press, 1988.

[31] D. Chandler. *Introduction to modern statistical mechanics.* Oxford University Press, Oxford, 1987.

[32] N. N. Chentsov. A systematic theory of exponential families of probability distributions. *Theor. Probability Appl.*, 11:425–425, 1966.

[33] N. N. Chentsov. *Statistical Decision Rules and Optimal Inference*, volume 53 of *Translations of Mathematical Monographs.* American Mathematical Society, 1982.

[34] S. Chopra. On the spanning tree polyhedron. *Operations Research Letters*, 8:25–29, 1989.

[35] K. Chou, A. Willsky, and R. Nikoukhah. Multiscale systems, Kalman filters, and Riccati equations. *IEEE Trans. AC*, 39(3):479–492, March 1994.

[36] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, IT-14:462–467, 1968.

[37] S.Y. Chung, T. Richardson, and R. Urbanke. Analysis of sum-product decoding of low-density parity check codes using a Gaussian approximation. *IEEE Trans. Info. Theory*, 47:657–670, February 2001.

[38] P. Clifford. Markov random fields in statistics. In G.R. Grimmett and D. J. A. Welsh, editors, *Disorder in physical systems.* Oxford Science Publications, 1990.

[39] G. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

[40] A. Corduneanu and T. Jaakkola. Stable mixing of complete and incomplete information. Technical Report AI-2001-030, MIT Artificial Intelligence Lab, November 2001.

[41] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* John Wiley and Sons, New York, 1991.

[42] I. Csiszár and G. Tusnády. Information geometry and alternating minimization procedures. In E. J. Dudewisc et al., editor, *Recent results in estimation theory and related topics.* 1984.

[43] I. Csiszár. I-divergence geometry of probability distributions and minimization problems. *Annals of Probability,*, 3(1):146–158, Feb. 1975.

[44] I. Csiszár. Sanov property, generalized I-projection and a conditional limit theorem. *Annals of Probability*, 12(3):768–793, Aug. 1984.

[45] I. Csiszár. A geometric interpretation of Darroch and Ratcliff's generalized iterative scaling. *Annals of Statistics*, 17(3):1409–1413, Sep. 1989.

[46] P. Dagum and M. Luby. Approximate probabilistic reasoning in Bayesian belief networks is NP-hard. *Artificial Intelligence*, 60:141–153, 1993.

[47] M. Daniel, A. Willsky, and D. Mclaughlin. Travel time estimation using a multiscale stochastic framework. *Advanced Water Resources*, pages 653–665, 2000.

[48] J. Darroch. Multiplicative and additive interaction in contingency tables. *Biometrika*, 61(2):207–214, 1974.

[49] J. Darroch, S. Lauritzen, and T. Speed. Markov fields and log-linear interaction models for contingency tables. *Annals of Statistics*, 8(3):522–539, 1980.

[50] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, 43:1470–1480, 1972.

[51] A. Darwiche. A differential approach to inference in Bayesian networks. In *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference (UAI-2000)*, pages 123–132, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

[52] A. P. Dawid. Conditional independence in statistical theory (with discussion). *Journal of the Royal Statistical Society, Series B*, 41:1–31, 1979.

[53] A. Dembo and O. Zeitouni. *Large deviation techniques and applications*, volume 38 of *Applications of mathematics*. Springer, New York, 1988.

[54] J.W. Demmel. *Applied numerical linear algebra*. SIAM, Philadelphia, 1997.

[55] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. Royal Stat. Soc. B*, 39:1–38, 1977.

[56] P. Diaconis and D. Stroock. Geometric bounds for eigenvalues of Markov chains. *Ann. Applied Probability*, 1:36–61, 1991.

[57] W. T. Freeman D.J.C. MacKay, J. S. Yedidia and Y. Weiss. A conversation about the Bethe free energy and sum-product algorithm. Technical Report TR2001-18, Mitsubishi Electric Research Labs, May 2001. Available at `http://www.merl.com/papers/TR2001-18/`.

[58] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1:127–136, 1971.

[59] I. Ekeland and R. Temam. *Convex analysis and variational problems.* Elsevier, New York, 1976.

[60] A. El Gamal and T. Cover. Multiple user information theory. *Proceedings of the IEEE*, 68(12):1466–1483, December 1980.

[61] K. Fan. Minimax theorems. *Proc. Nat. Acad. Sci. U.S.A.*, 39:42–47, 1953.

[62] P. Fieguth, W. Karl, A. Willsky, and C. Wunsch. Multiresolution optimal interpolation of satellite altimetry. *IEEE Trans. Geo. Rem.*, 33(2):280–292, March 1995.

[63] A. Frakt. *Internal multiscale autoregressive processes, stochastic realization, and covariance extension.* PhD thesis, Massachusetts Institute of Technology, August 1999.

[64] D. C. Fraser. A new technique for the optimal smoothing of data. Technical report, Massachusetts Institute of Technology, 1967.

[65] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *Intl. J. Computer Vision*, 40(1):25–47, 2000.

[66] W. T. Freeman and Y. Weiss. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Trans. Info. Theory*, 47:736–744, 2001.

[67] B. Frey. *Graphical models for machine learning and digital communication.* MIT Press, Cambridge, MA, 1998.

[68] B. Frey, R. Koetter, and N. Petrovic. Very loopy belief propagation for unwrapping phase images. In *NIPS 14*. MIT Press, 2001. To appear.

[69] R. G. Gallager. Low-density parity check codes. *IRE Trans. Inform. Theory*, IT-8:21–28, 1962.

[70] R. G. Gallager. *Low-density parity check codes.* MIT Press, Cambridge, MA, 1963.

[71] R. G. Gallager. *Information theory and reliable communication.* Wiley, New York, NY, 1968.

[72] I.M. Gel'fand and S.V. Fomin. *Calculus of variations.* Prentice-Hall, Eaglewood Cliffs, New Jersey, 1963.

[73] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Pat. Anal. Mach. Intell.*, 6:721–741, 1984.

[74] Z. Ghahramani and M. Jordan. Factorial hidden Markov models. *Machine Learning*, 29:245–273, 1997.

[75] W. Gibbs. *Elementary principles of statistical mechanics.* Yale University Press, New Haven, 1902.

[76] J. Goodman. The multivariate analysis of qualitative data: Interaction among multiple classifications. *Journal of the American Statistical Association*, 65:226–256, 1970.

[77] A. J. Grant. Information geometry and iterative decoding. In *IEEE Communication Theory Workshop*, Aptos, CA, 1999. Morgan Kaufmann Publishing.

[78] K. D. Gremban. *Combinatorial preconditioners for sparse, symmetric, diagonally dominant systems.* PhD thesis, Carnegie Mellon University, 1996. Available as Tech. Report CMU-CS-96-123.

[79] G. R. Grimmett. A theorem about random fields. *Bulletin of the London Mathematical Society*, 5:81–84, 1973.

[80] G.R. Grimmett and D.R. Stirzaker. *Probability and random processes.* Oxford Science Publications, Clarendon Press, Oxford, 1992.

[81] S. Guattery. Graph embedding techniques for bounding condition numbers of incomplte factor preconditioners. Technical Report 97-47, ICASE, NASA Langley Research Center, 1997.

[82] S. S. Gupta, editor. *Differential geometry in statistical inference*, volume 10 of *Lecture notes - Monograph series.* Institute of Mathematical Statistics, Hayward, CA, 1987.

[83] G. Hardy, E. Littlewood, and G. Polya. *Inequalities.* Cambridge University Press, Cambridge, 1954.

[84] M. Hassner and J. Sklansky. Markov random field models of digitized image texture. In *ICPR78*, pages 538–540, 1978.

[85] M.R. Hestenes and E. L. Stiefel. Methods of conjugate gradients for solving linear systems. *Jour. Res. Nat. Bur. Standards Sec. B*, 49:409–436, 1952.

[86] J. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms.* Springer-Verlag, New York, 1993.

[87] T. Ho. *Multiscale modelling and estimation of large-scale dynamic systems.* PhD thesis, Massachusetts Institute of Technology, September 1998.

[88] W. Irving, P. Fieguth, and A. Willsky. An overlapping tree approach to multiscale stochastic modeling and estimation. *IEEE Transactions on Image Processing*, 6(11), November 1997.

[89] W. Irving and A. Willsky. A canonical correlations approach to multiscale stochastic realization. *IEEE Transactions on Automatic Control*, 46(10):1514–1528, October 2001.

[90] E. Ising. Beitrag zur theorie der ferromagnetismus. *Zeitschrift für Physik*, 31:253–258, 1925.

[91] T. S. Jaakkola. *Variational methods for inference and estimation in graphical models.* PhD thesis, Massachusetts Institute of Technology, 1997.

[92] T. S. Jaakkola. Tutorial on variational approximation methods. In *Advanced mean field methods.* MIT Press, 2001.

[93] T. S. Jaakkola and M. Jordan. Computing upper and lower bounds on likelihoods in intractable networks. In *Uncertainty in Artificial Intelligence: Proceedings of the Twelfth Conference (UAI-1996)*, pages 340–348, San Francisco, CA, 1996. Morgan Kaufmann Publishers.

[94] T. S. Jaakkola and M. Jordan. Recursive algorithms for approximating probabilities in graphical models. In *Advances in Neural Information Processing Systems*, volume 9, 1996.

[95] T. S. Jaakkola and M. Jordan. Improving the mean field approximation via the use of mixture distributions. In M. Jordan, editor, *Learning in graphical models*, pages 105–161. MIT Press, 1999.

[96] T. S. Jaakkola and M. Jordan. Variational probabilistic inference and the QMR-DT network. *Journal of Artificial Intelligence Research*, 10:291–322, 1999.

[97] T. S. Jaakkola, M. Meila, and T. Jebara. Maximum entropy discrimination. Technical Report AITR-1668, Massachusetts Institute of Technology, 1999.

[98] E. T. Jaynes. On the rationale of maximum entropy methods. *Proc. IEEE*, 70:939–952, 1982.

[99] E. T. Jaynes. *Papers on probability, statistics, and statistical physics.* Reidel, Dordrecht, 1982.

[100] A. H. Jazwinski. *Stochastic processes and filtering theory.* Academic Press, New York, 1970.

[101] F. V. Jensen. *An introduction to Bayesian networks.* UCL Press, London, 1996.

[102] M. Jerrum and A. Sinclair. Polynomial-time approximation algorithms for the Ising model. *SIAM Journal Comput.*, 22:1087–1116, 1993.

[103] M. Jordan. *Learning in graphical models.* MIT Press, Cambridge, MA, 1999.

[104] M. Jordan. *Introduction to graphical models.* MIT Press, Cambridge, MA, Forthcoming.

[105] M. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. In *Learning in graphical models*, pages 105–161. MIT Press, 1999.

[106] G. D. Forney Jr., F. R. Kschischang, B. Marcus, and S. Tuncel. Iterative decoding of tail-biting trellises and connections with symbolic dynamics. In *Codes, systems and graphical models*, pages 239–264. Springer, 2001.

[107] D. Jungnickel. *Graphs, networks, and algorithms.* Algorithms and computation in mathematics. Springer, New York, 1999.

[108] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation.* Prentice Hall, New Jersey, 2000.

[109] R. Kalman. A new approach to linear filtering and prediction problems. *The American Society of Mechanical Engineers: Basic Engineering, series D*, 82:35–45, March 1960.

[110] R. Kalman and R. Bucy. New results in linear filtering and prediction theory. *The American Society of Mechanical Engineers: Basic Engineering, series D*, 83:95–108, March 1961.

[111] D. Karger and N. Srebro. Learning Markov networks: maximum bounded tree-width graphs. In *Symposium on Discrete Algorithms*, pages 392–401, 2001.

[112] R.L. Kashyap. Random field models of images. *CGIP*, 12(3):257–270, March 1980.

[113] M. G. Kendall and A. Stuart. *The advanced theory of statistics*, volume 1. Hafner Publishing Company, New York, 1969.

[114] R. Kikuchi. The theory of cooperative phenomena. *Physical Review*, 81:988–1003, 1951.

[115] Uffe Kjaerulff and Linda C. van der Gaag. Making sensitivity analysis computationally efficient. In *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference (UAI-2000)*, pages 317–325, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

[116] J. Kruskal. On the shortest spanning subtree of a graph and the travelling salesman problem. *Proc. AMS*, 7:48–50, 1956.

[117] F. Kschischang and B. Frey. Iterative decoding of compound codes by probability propagation in graphical models. *IEEE Sel. Areas Comm.*, 16(2):219–230, February 1998.

[118] S. Kullback. *Information theory and statistics.* Wiley, New York, 1959.

[119] S. Kullback. Probability densities with given marginals. *Annals of Mathematical Statistics*, 39:1236–1243, 1968.

[120] K. Laskey. Sensitivity analysis for probability assessments in Bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 25:901–909, 1995.

[121] S. L. Lauritzen. *Graphical models.* Oxford University Press, Oxford, 1996.

[122] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society B*, 50:155–224, January 1988.

[123] M.A.R. Leisink and H.J. Kappen. A tighter bound for graphical models. In *NIPS 13*, pages 266–272. MIT Press, 2001.

[124] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. Spielman. Improved low-density parity-check codes using irregular graphs and belief propagation. In *Proceedings 1998 International Symposium on Information Theory*, page 117. IEEE, 1998.

[125] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. Spielman. Improved low-density parity check codes using irregular graphs. *IEEE Trans. Info. Theory*, 47:585–598, February 2001.

[126] M. Luettgen, W. Karl, and A. Willsky. Efficient multiscale regularization with application to optical flow. *IEEE Trans. Image Processing*, 3(1):41–64, Jan. 1994.

[127] M. Luettgen and A. Willsky. Likelihood calculation for a class of multiscale stochastic models, with application to texture discrimination. *IEEE Trans. Image Processing*, 4(2):194–207, February 1995.

[128] D. J. C. MacKay. Introduction to Monte Carlo methods. In *Learning in graphical models*, pages 175–204. MIT Press, 1999.

[129] D.J.C. MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Trans. Info. Theory*, 45(2):399–431, 1999.

[130] R.J. McEliece, D.J.C. McKay, and J.F. Cheng. Turbo decoding as an instance of Pearl's belief propagation algorithm. *IEEE Jour. Sel. Communication*, 16(2):140–152, February 1998.

[131] T. P. Minka. *A family of algorithms for approximate Bayesian inference.* PhD thesis, MIT Media Lab, January 2001.

[132] K. Murphy and Y. Weiss. The factored frontier algorithm for approximate inference in DBNs. In *Uncertainty in Artificial Intelligence*, volume 11, 2001.

[133] K. Murphy, Y. Weiss, and M. Jordan. Loopy-belief propagation for approximate inference: An empirical study. In *Uncertainty in Artificial Intelligence*, volume 9, 1999.

[134] J. A. O'Sullivan. Alternating minimization algorithms: from Blahut-Arimoto to Expectation-Maximization. In A. Vardy, editor, *Codes, curves, and signals: Common threads in communications*, pages 173–192. Kluwer Academic Press, 1998.

[135] G. Parisi. *Statistical field theory.* Addison-Wesley, 1988.

[136] D. W. Peaceman and H. H. Rachford, Jr. The numerical solution of parabolic and elliptic differential equations. *Journal of the SIAM*, 3(1):28–41, March 1955.

[137] J. Pearl. *Probabilistic reasoning in intelligent systems.* Morgan Kaufman, San Mateo, 1988.

[138] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

[139] G. Potamianos and J. Goutsias. Partition function estimation of Gibbs random field images using Monte Carlo simulations. *IEEE Trans. Info. Theory*, 39(4):1322–1332, July 1993.

[140] G. Potamianos and J. Goutsias. Stochastic approximation algorithms for partition function estimation of Gibbs random fields. *IEEE Trans. Info. Theory*, 43(6):1948–1965, November 1997.

[141] J. G. Propp and D. B. Wilson. How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph. *Journal of Algorithms*, 27:170–217, 1998.

[142] W. R. Pulleyblank. Polyhedral combinatorics. In A. Bachem, M. Grötschel, and B. Korte, editors, *Mathematical programming: the state of the art*, pages 312–345. Springer-Verlag, 1983.

[143] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, February 1989.

[144] L. R. Rabiner and B. H. Juang. *Fundamentals of speech recognition.* Prentice Hall, Englewood Cliffs, N.J., 1993.

[145] C. R. Rao. Information and accuracy obtainable in the estimation of statistical parameters. *Bulletin Calcutta Math. Soc.*, 37:81–91, 1945.

[146] H. Rauch, F. Tung, and C. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, August 1965.

[147] T. Richardson. The geometry of turbo-decoding dynamics. *IEEE Trans. Info. Theory*, 46(1):9–23, January 2000.

[148] T. Richardson, A. Shokrollahi, and R. Urbanke. Design of capacity-approaching irregular low-density parity check codes. *IEEE Trans. Info. Theory*, 47:619–637, February 2001.

[149] T. Richardson and R. Urbanke. The capacity of low-density parity check codes under message-passing decoding. *IEEE Trans. Info. Theory*, 47:599–618, February 2001.

[150] B. D. Ripley. *Stochastic simulation.* Wiley, New York, 1987.

[151] G. Rockafellar. *Convex Analysis.* Princeton University Press, Princeton, 1970.

[152] P. Rusmevichientong and B. Van Roy. An analysis of turbo decoding with Gaussian densities. In *NIPS 12*, pages 575–581. MIT Press, 2000.

[153] J. S. Rustagi. *Variational methods in statistics*, volume 121 of *Mathematics in Science and Engineering.* Academic Press, 1976.

[154] L. Saul, T. S. Jaakkola, and M. Jordan. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, 4:61–76, 1996.

[155] L. K. Saul and M.I. Jordan. Exploiting tractable substructures in intractable networks. In *NIPS 8*, pages 486–492. MIT Press, 1996.

[156] M. Schneider. *Krylov Subspace Estimation.* PhD thesis, Massachusetts Institute of Technology, February 2001.

[157] H. R. Schwarz. *Finite element methods.* Academic Press, New York, 1988.

[158] A. Shwartz and A. Weiss. *Large deviations for performance analysis: queues, communications and computing.* Chapman and Hall, New York, 1995.

[159] A. Sinclair. Improved bounds for mixing rates of Markov chains and multicommodity flow. *Combinatorics, Probability and Computing*, 1:351–370, 1992.

[160] T. P. Speed and H. T. Kiiveri. Gaussian Markov distributions over finite graphs. *Annals of Statistics*, 14(1):138–150, March 1986.

[161] M. Spivak. *Calculus*. W. A. Benjamin, New York, 1967.

[162] N. Srebro. Maximum likelihood Markov networks: an algorithmic approach. Master's thesis, Massachusetts Institute of Technology, October 2000.

[163] E. Sudderth. Embedded trees: estimation of Gaussian processes on graphs with cycles. Master's thesis, Massachusetts Institute of Technology, January 2002.

[164] K. Tanaka and T. Morita. Cluster variation method and image restoration problem. *Physics Letters A*, 203:122–128, 1995.

[165] C. J. Thompson. *Classical equilibrium statistical mechanics*. Clarendon Press, Oxford, 1988.

[166] P.M. Vaidya. Solving linear equations with symmetric diagonally dominant matrices by constructing good preconditioners. Unpublished manuscript; presented at IMA Workshop on Graph Theory and Sparse Matrix Computation, Minneapolis, October 1991.

[167] J. H. van Lint. *Introduction to coding theory*. Springer, New York, 1999.

[168] J. H. van Lint and R. M. Wilson. *A course in combinatorics*. Cambridge University Press, Cambridge, 1992.

[169] S. Verdu and H. V. Poor. Abstract dynamic programming models under commutativity conditions. *SIAM J. Control and Optimization*, 25(4):990–1006, July 1987.

[170] J. von Neumann. *Theory of games and economic behavior*. Princeton University Press, Princeton, 1953.

[171] M. J. Wainwright, E. P. Simoncelli, and A. S. Willsky. Random cascades on wavelet trees and their use in modeling and analyzing natural images. *Applied Computational and Harmonic Analysis*, 11:89–123, 2001.

[172] M. J. Wainwright, E. B. Sudderth, and A. S. Willsky. Tree-based modeling and estimation of Gaussian processes on graphs with cycles. In *Advances in Neural Information Processing Systems 13*, pages 661–667. MIT Press, 2001. Paper available at http://ssg.mit.edu/group/mjwain/mjwain.shtml.

[173] Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12:1–41, 2000.

[174] Y. Weiss and W. T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. In *NIPS 12*, pages 673–679. MIT Press, 2000.

[175] M. Welling and Y. Teh. Belief optimization: A stable alternative to loopy belief propagation. In *Uncertainty in Artificial Intelligence*, July 2001.

[176] W. Wiegerinck. Variational approximations between mean field theory and the junction tree algorithm. In *UAI 2000*, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

[177] J.W. Woods. Markov image modeling. *IEEE Transactions on Automatic Control*, 23:846–850, October 1978.

[178] C. H. Wu and Peter C. Doerschuk. Tree approximations to Markov random fields. *IEEE Trans. on PAMI*, 17(4):391–402, April 1995.

[179] N. Wu. *The maximum entropy method.* Springer, New York, 1997.

[180] J. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS 13*, pages 689–695. MIT Press, 2001.

[181] A. Yuille. A double-loop algorithm to minimize the Bethe and Kikuchi free energies. *Neural Computation*, To appear, 2001.

[182] J. Zhang. The application of the Gibbs-Bogoliubov-Feynman inequality in mean-field calculations for Markov random-fields. *IEEE Trans. on Image Processing*, 5(7):1208–1214, July 1996.