Convex Relaxation Methods for Graphical Models: Lagrangian and Maximum Entropy Approaches

by

Jason K. Johnson

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

September, 2008

© 2008 Massachusetts Institute of Technology
All Rights Reserved.

Signature of Author: _____

Department of Electrical Engineering and Computer Science
August 18, 2008

Certified by: _____

Alan S. Willsky, Professor of EECS
Thesis Supervisor

Accepted by: _____

Terry P. Orlando, Professor of Electrical Engineering
Chair, Department Committee on Graduate Students

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
OCT 22 2008
LIBRARIES

ARCHIVES

Convex Relaxation Methods for Graphical Models: Lagrangian and Maximum Entropy Approaches

by Jason K. Johnson

Submitted to the Department of Electrical Engineering
and Computer Science on August 18, 2008
in Partial Fulfillment of the Requirements for the Degree
of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Graphical models provide compact representations of complex probability distributions of many random variables through a collection of potential functions defined on small subsets of these variables. This representation is defined with respect to a graph in which nodes represent random variables and edges represent the interactions among those random variables. Graphical models provide a powerful and flexible approach to many problems in science and engineering, but also present serious challenges owing to the intractability of optimal inference and estimation over general graphs. In this thesis, we consider convex optimization methods to address two central problems that commonly arise for graphical models.

First, we consider the problem of determining the most probable configuration—also known as the maximum a posteriori (MAP) estimate—of all variables in a graphical model, conditioned on (possibly noisy) measurements of some variables. This general problem is intractable, so we consider a Lagrangian relaxation (LR) approach to obtain a tractable dual problem. This involves using the Lagrangian decomposition technique to break up an intractable graph into tractable subgraphs, such as small "blocks" of nodes, embedded trees or thin subgraphs. We develop a distributed, iterative algorithm that minimizes the Lagrangian dual function by block coordinate descent. This results in an iterative marginal-matching procedure that enforces consistency among the subgraphs using an adaptation of the well-known iterative scaling algorithm. This approach is developed both for discrete variable and Gaussian graphical models. In discrete models, we also introduce a deterministic annealing procedure, which introduces a temperature parameter to define a smoothed dual function and then gradually reduces the temperature to recover the (non-differentiable) Lagrangian dual. When strong duality holds, we recover the optimal MAP estimate. We show that this occurs for a broad class of "convex decomposable" Gaussian graphical models, which generalizes the "pairwise normalizable" condition known to be important for iterative estimation in Gaussian models. In certain "frustrated" discrete models a duality gap can occur using simple versions of our approach. We consider methods that adaptively enhance the dual formulation, by including more complex subgraphs, so as to reduce the duality gap. In many cases we are able to eliminate the duality gap and obtain the optimal MAP estimate in a tractable manner. We also propose a heuristic method to obtain approximate solutions in cases where there is a duality gap.

Second, we consider the problem of learning a graphical model (both the graph and its potential functions) from sample data. We propose the maximum entropy relaxation (MER) method, which is the convex optimization problem of selecting the least informative (maximum entropy) model over an exponential family of graphical models subject to constraints that small subsets of variables should have marginal distributions that are close to the distribution of sample data. We use relative entropy to measure the divergence between marginal probability distributions. We find that MER leads naturally to selection of sparse graphical models. To identify this sparse graph efficiently, we use a "bootstrap" method that constructs the MER solution by solving a sequence of tractable subproblems defined over thin graphs, including new edges at each step to correct for large marginal divergences that violate the MER constraint. The MER problem on each of these subgraphs is efficiently solved using the primal-dual interior point method (implemented so as to take advantage of efficient inference methods for thin graphical models). We also consider a dual formulation of MER that minimizes a convex function of the potentials of the graphical model. This MER dual problem can be interpreted as a robust version of maximum-likelihood parameter estimation, where the MER constraints specify the uncertainty in the sufficient statistics of the model. This also corresponds to a regularized maximum-likelihood approach, in which an information-geometric regularization term favors selection of sparse potential representations. We develop a relaxed version of the iterative scaling method to solve this MER dual problem.

Thesis Supervisor: Alan S. Willsky
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I thank my thesis advisor, Alan Willsky, for accepting me into his group and for guiding and supporting my research over the years. As his student, I have enjoyed an uncommon level of intellectual freedom that has allowed me to explore a wide range of ideas. I have great respect for his integrity, dedication and enthusiasm. I am also grateful for his meticulous reading of the thesis and for his rapid return of drafts. I thank my thesis committee members, Sanjoy Mitter and Tommi Jaakkola, for their advice and for quickly reading a draft of the thesis.

I thank Bob Washburn, Bill Irving and Mark Luettgen, whom I worked with at Alphatech, Inc. (now the AIT Division of BAE systems), for having inspired me to pursue a graduate degree in electrical engineering and computer science. I am grateful to all of my past "grouplet" members for the influence they have had on my research: Mike Schneider, Dewey Tucker, Martin Wainwright, Erik Sudderth, Dmitry Malioutov, Venkat Chandrasekaran, Jin Choi, Lei Chen, Pat Kreidel and Ayres Fan. In particular, I thank Dmitry and Venkat for their collaboration on numerous research topics, including work presented in this thesis. I have especially enjoyed our Wednesday night treks, often joined by Pat and Ayres, to the Muddy Charles for drinks over a game of cards. Also, I thank Evan Fortunato and Mark Luettgen for bringing me back to Alphatech one summer to work on multi-target tracking, which sparked some ideas that led to the Lagrangian relaxation work presented in this thesis.

I thank my parents for pretty much everything, particularly for having always encouraged me to go my own way in life. I thank Joel Gwynn for being a good friend and for providing no-nonsense advice when I needed it. I also thank Tricia Joubert for being my best friend and companion these past four yours. She has been an essential support as I have dealt with the stresses of completing the doctoral program.

Contents

Chapter 1

Introduction

1.1 Motivation and Overview

Graphical models [43, 60, 145, 185] are probabilistic models for complex systems of random variables where the joint probability distribution of all variables is compactly specified by a set of *interactions* among variables. In the case of undirected graphical models, which we also refer to as *Markov random fields* (MRFs), each interaction is specified by a *potential function*, defined on a subset of the variables, that provides a relative measure of compatibility between the different joint configurations of these variables. The structure of the model thus defines a graph, each variable is identified with a node of the graph and interactions between variables define edges of the graph. In some cases, the probability model actually represents some naturally occurring random process. In others, we seek to *optimize* some objective function, which may be then be interpreted as finding the ground state of the *Gibbs distribution* [90] based on this objective function. Models of this form arise in many fields of science and engineering:

- statistical physics [129, 195, 229],
- signal processing [16, 19, 83, 130, 207],
- image processing [28, 88, 149, 222, 223],
- medical imaging and tomography [84, 176],
- geophysics and remote sensing [55, 112, 126, 197],
- circuit layout design [12, 148, 151],
- communication and coding theory [85, 153, 179], and
- distributed estimation in sensor networks [44, 51, 66, 113, 162, 189, 224].

However, the utility of these models in practical applications is often limited by the fact that optimal inference and optimization within this model class is generally intractable for large problems with many variables [10, 58]. As a result, there has been an intense, ongoing effort in recent years to develop tractable yet principled approaches to *approximate* inference within this rich class of models.

In this thesis, we focus on two central problems that arise for graphical models. First, we consider the problem of *maximum a posteriori* (MAP) estimation. That is, given a graphical model defined on a set of variables, and possibly partial observations (e.g., noisy measurements) of subsets of these variables, we seek a joint estimate of all unknown variables that maximizes the conditional joint probability of the estimate given the observations. In general, this problem is NP-hard to solve exactly in models with discrete (e.g., binary valued) variables. We develop a *Lagrangian relaxation* (LR) method [22,80,89] that decomposes the problem into tractable subproblems defined on smaller or more tractable subgraphs. This general approach of splitting an intractable problem into tractable subproblems, by introducing copies of some variables and relaxing equality constraints between these copies, is also known as *Lagrangian decomposition* [48,100,159] (we use these terms interchangeably in this thesis). In many cases our *graphical* decomposition approach leads to the *optimal* MAP estimate, in which case one says that *strong duality holds*. However, because the general problem is NP-hard, we must expect to also encounter cases where there is a *a duality gap* and the optimal MAP estimate cannot be obtained. We also propose a simple heuristic method to obtain *approximate* solutions in this case.

The second problem we consider is that of *model selection* [41,106], that is, of selecting *both* the graph structure and a corresponding set of potential functions to obtain a good fit to sample data. Our approach to this problem of learning a graphical model from sample data is also useful if one instead seeks to *thin* a graphical model, that is, to adaptively select a simpler graphical model that still provides a good approximation to a more complex model. While early work on these problems has focused on primarily on greedy combinatorial approaches to select the graph structure [67,177,199], we focus instead on a convex optimization approach to simultaneously learn both the graph and its potentials. The main idea is to relax the well-known maximum entropy modeling approach [59,97,117,177] to obtain a *regularized* maximum entropy method, one that implicitly favors sparser graphical models. This involves introducing constraints on the marginal distributions of the model, that they should be close to the empirical marginals (from sample data or a more complex model that we wish to thin) as measured by relative entropy [59] (also known as *Kullback-Leibler divergence* [142,143]). We also derive a dual version of this problem which leads naturally to a relaxed versions of the iterative scaling algorithm [62, 114, 186, 199], often used for learning graphical models with a fixed graph structure.

A key idea common to both of these approaches is seeking *convex relaxations* of intractable problems [37]. In the case of Lagrangian relaxation for discrete graphical models, a non-convex integer programming problem is relaxed to the convex Lagrangian dual problem. In maximum entropy relaxation, the non-convex problem of selecting a graph is relaxed to convex optimization over a denser graph (e.g., the complete graph) but with a regularization method to enforce sparsity in the potentials on this denser graph, thereby selecting a sparse subgraph.

1.2 Related Work

Before discussing our methods and contributions further, we give a brief account of relevant work on tractable inference and learning methods for graphical models and of approximate methods for intractable models. A more detailed discussion of many of these approaches is given in the background (see Chapter 2).

1.2.1 MAP Estimation

Dynamic Programming and Combinatorial Optimization

There are several classes of graphical models for which inference is tractable, either to compute the marginal distributions of individual variables or the MAP estimate. In graphs with *low tree-width* [6, 31, 32], one can exactly compute either marginals or the so-called *max-marginals* to obtain the MAP estimate. These approaches involve variable elimination steps that either sum or maximize over individual variables to obtain marginals. In the case of maximizing (to solve the MAP problem), this method is a generalization of well-known dynamic programming methods such as the *Viterbi algorithm* [16, 19, 83, 207]. In order to apply this tree-structured inference procedure to general graphs, one converts the graph to a tree using the concept of *junction trees* [146]. Roughly speaking, this involves grouping nodes together to define an equivalent Markov tree representation of the model. The tree-width is determined by how many nodes must be grouped together in this procedure. In the class of bounded tree-widths graphs, the computational complexity of this procedure grows linearly in the number of nodes. However, its complexity is *exponential* in the tree-width and it is therefore only tractable for *thin* graphs, that is, for graphs with low tree-width.

However, for special classes or problems it is still possible to solve the MAP problem exactly even if the graph is not thin. We mention only a few well-known examples. First, there are a number of well-studied combinatorial and network optimization problems that have efficient solutions [42, 171], including: the max-cut/min-flow problem [82], maximum-weight matching in bipartite graphs [73], and minimum-weight perfect matching in planar graphs [57, 165]. Several connections have been found between such network optimization problems and MAP estimation in graphical models. For example, the *ferromagnetic Ising model* can be solved exactly using a max-flow/min-cut reformulation of the problem [11, 98]. This is a binary variable graphical model, with node states $+1$ and -1, in which all interactions are pairwise and where the pairwise potentials prefer configurations in which adjacent variables are assigned the same value. Similarly, *zero-field Ising models* defined on planar graphs can be solved exactly as a minimum-weight perfect matching problem [29, 87, 172, 203]. In this case, pairwise potentials may also be *anti-ferromagnetic* so as to prefer configurations in which adjacent nodes have opposite states. But the model is required to have *zero-field*, which essentially means that every configuration and its negation (with all nodes assigned opposite values) are equally likely. MAP estimation in the general Ising model can be reformulated as a max-cut problem [12, 13]. Although it is not tractable to solve

max-cut in general graphs, Barahona and Mahjoub have proposed a heuristic cutting-plane method based on the *odd cycle inequalities* [13]. In planar graphs, this leads to an optimal solution of the max-cut problem and therefore solves the zero-field planar Ising model. Other connections to network optimization have emerged. For instance, a number of works have recently appeared using linear-programming relaxations of the MAP estimation problem [50, 76, 133, 211, 219]. In earlier work on binary quadratic programming [34, 36, 103], it was found that the value of this linear program (LP) can be computed using max-flow/min-cut techniques. In cases where solution of the LP leads to an integral solution, the correct MAP estimate is obtained. Otherwise, there is an *integrality gap* and the value of the LP provides an upper-bound on the value of the MAP problem. Other approaches use LP methods in conjunction with the branch and bound procedure, and often succeed in identifying the MAP estimate [196]. However, the number of steps required to reach an optimal solution may be exponential in problem size in the worst-case.

Many methods have appeared in the graphical modeling literature aimed at solving (at least approximately) the MAP estimation problem. This problem is closely related to that of computing marginal distributions of the model. The *sum-product algorithm* [85], also known as *belief propagation* (BP) [175], is an iterative message-passing algorithm for computing approximate marginal distributions of each variable in a graphical model. It is based on an exact inference method for trees, which involves passing messages along the edges of the tree. Each node fuses messages, from all but one its neighbors, and then propagates this information to the excluded neighbor based on the edge potential linking the two nodes. In loopy graphs, this procedure does not always converge to a fixed-point and may give inaccurate marginals when it does converge. Nonetheless, it has yielded good results in many practical applications. Another form of belief propagation, the *max-product algorithm*, may be regarded as approximating the "zero-temperature" marginals of a graphical model, which encode the distribution of a variable over the set of MAP estimates, and is closely related to dynamic programming methods such as the Viterbi algorithm. Convergence of max-product tends to be less robust than the sum-product algorithm. Also, if max-product does converge it may still give an incorrect MAP estimate. However, this estimate does at least satisfy a certain local-optimality condition with respect to induced subtrees of the graph [218].

More recent work has focused on convex forms of belief propagation [219, 226], starting with the work of Martin Wainwright [211, 212] on approximation methods based on convex decompositions of a graphical model into a set of more tractable models defined on spanning trees (see also earlier work on fractional belief propagation [221]). Max-product forms of these methods, such as tree-reweighted max-product (TRMP) [211], aim to minimize a convex function that is an upper-bound on the MAP value. This corresponds to a linear-programming dual [25] of previously considered LP relaxations of MAP estimation, either based on an outer-bound approximation of the marginal polytope [50, 133] or the standard linearization method [216] (see also [103]). The advantage

of such dual methods is that they provide efficient solution methods based on BP-like distributed message-passing algorithms (see [226] for an empirical comparison between message-passing approaches and traditional approaches to solve linear programs). However, because belief propagation does not always converge, there is growing interest in other convergent iterative methods to solve these dual formulations using coordinate-descent methods. This includes our own work presented in this thesis and in our earlier paper [125], based on the Lagrangian decomposition formulation, as well as other recent work [93,134] that also used coordinate-descent approaches. Also, Tom Werner recently published a paper [220] reviewing earlier work [140, 192], not previously published in English, on the *max-sum diffusion* algorithm. All of these methods lead to similar style update rules but are not precisely equivalent because they use different parameterizations such that coordinate-descent in these different parameterization does not lead to equivalent algorithms. One difficulty encountered in such coordinate-descent approaches, when applied to a non-differentiable objective (such as the dual functions that arise in these formulations), may get stuck at a *non-minimum* fixed point of the algorithm [134, 191]. One proposal to address this problem has been to use instead a *low-temperature* version of the convex sum-product algorithm [219]. Although this approach is very reasonable insofar as it "smoothes" the objective function, the issue of convergence (and rate of convergence) of this algorithmic approach has not been resolved. For instance, it is known that even convex versions of BP do not necessarily converge. Using sufficient damping of message updates may help, but it seems unlikely to be very efficient at low temperatures. Our approach uses a temperature annealing idea to overcome this difficulty in conjunction with a coordinate-descent method. However, our approach is *deterministic*, and should not be confused with randomized algorithms such as *simulated annealing* [88]. In this regard, our approach is in the same spirit as several methods developed in the neural network literature for solving combinatorial optimization problems [95, 139, 178].

While this thesis was in preparation, several other papers have appeared, in addition to our publication [125], that independently propose Lagrangian decomposition approaches to MAP estimation in graphical models [137, 230]. Also, we recently discovered earlier work of Storvik and Dahl on this topic [200]. All of these methods only consider decompositions that are equivalent to the simplest pairwise relaxation in our method.[1] Also, all of these papers minimize the dual function using subgradient methods, a standard approach from the integer programming literature, which often suffers from slow convergence to the minimum of the dual function (in the worst case, the rate of convergence is sublinear).

Gaussian Inference

We are also interested in the problem of MAP estimation in large-scale Gaussian graphical models, also known as Gauss-Markov random fields (GMRFs) [144, 185, 199]. For

[1] Although [200] considers a decomposition of a 2D grid in vertical and horizontal chains, this is actually equivalent to the pairwise relaxation.

GMRF models, MAP estimation reduces to minimizing a convex, quadratic objective function based on a sparse, symmetric positive-definite matrix, the *information matrix* of the model. Equivalently, the optimal solution of this minimization problem can be obtained by solving a linear system of equations based on the information matrix. This solution can be computed directly by *Gaussian elimination* [96], which has cubic computational complexity in the general case. For GMRFs, the graphical structure of the model is determined by the fill-pattern (sparsity) of the information matrix. This enables solution methods using sparse elimination procedures, such as the *nested dissection* procedure for computing a sparse Cholesky factorization of the information matrix, which results in computational complexity that is cubic in the *tree-width* of the graph (rather than the total number of variables) [182,185]. While this is a tremendous improvement for sufficiently thin models, it is still unsatisfactory for many applications where Gauss-Markov random fields occur with very large tree-width, such as in 2-D models commonly used for image processing and remote sensing (where tree-widths of 1,000 or more are common) or in 3-D models used for tomography or remote sensing (where tree-widths of $100 \times 100 = 10,000$ are common). In such applications, it is impractical to use direct factorization methods and it becomes preferable instead to use iterative methods that obtain approximate solutions with computational complexity (per iteration) that scales linearly in problem size. For instance, one might use classical iterative methods such as the Gauss-Jacobi or Gauss-Seidel iterations [206]. The *embedded trees algorithm* [201] and its variants [45,47,66] were developed to accelerate the convergence of iterative methods. These are iterative methods that use a sequence of *preconditioners*, based on embedded trees or other tractable subgraphs, to update the estimate based on the residual error at each iteration.

Another approach is to use the Gaussian form of belief propagation [124,157,217]. It has been shown [217] that if Gaussian BP converges then it recovers the correct MAP estimate, which, for GMRFs, is equivalent to computing the mean value of each variable. In addition to computing these means, Gaussian belief propagation also computes approximate variances of each variable. Recent work on the walk-sum view of inference in Gaussian graphical models [47, 124, 157] has shown that a wide range of iterative methods may be viewed as computing walk-sums and, for the class of *walk-summable* models, these iterative methods are guaranteed to converge to the correct MAP estimate. In other work, a recursive divide and conquer approach to approximate inference in GMRFs has been proposed using a combination of nested dissection, Gaussian elimination and model thinning operations [118,126]. This approach leads to improved variance estimates and rapid convergence to the correct means when used as a preconditioner. Another method for computing approximate variances was developed in [155,156]. This approach relies upon fast linear solvers and a low-rank approximation to the identity matrix. Then, the covariance matrix, which is equal to the inverse of the information matrix, is approximated by solving a small number of linear systems.

■ 1.2.2 Learning Graphical Models

Parameter Fitting

For a fixed graph structure, learning a graphical model involves selecting the potential functions of the model to obtain the best fit of the overall probability distribution to observed sample data of the model variables. The standard method is to maximize the likelihood of the sample data as a function of the model parameters [81, 174], which specify the potential representation of the graphical model. If there are no hidden variables, that is, if the sample data consists of complete observations of all variables in the graphical model, then this may be posed as a convex optimization problem in an exponential family model (e.g., the Gibbs representation of a discrete variable model) and solved using standard convex optimization methods [24, 37]. However, computing the likelihood, or its gradient, is just as as difficult as inference, that is, computing marginal distributions of the model. For this reason, maximum-likelihood modeling is only tractable for models for which exact inference is tractable. Otherwise, one must resort to approximate learning, based on approximate inference or Monte-Carlo methods to estimate marginal distributions of the model.

The *iterative scaling* algorithm, also known as *iterative proportional fitting*, is one common approach to learning graphical models [62, 114, 186, 199]. This procedure iteratively adjusts the potentials of the graphical model by multiplying each potential (in the product representation of the graphical model) by the ratio of the empirical distribution of the corresponding variables (obtained from sample data) divided by their marginal distribution in the current estimate of the model (computing by some inference method). This has a geometric interpretation, within the information geometric view of the exponential family, as computing the minimum relative-entropy projection onto the set of models that are consistent with the data. The iterative scaling procedure performs a sequence of such projections, where each projection imposes consistency with the data for a subset of nodes. By iterating over all subsets of interacting nodes, this sequence of projections converges to the desired projection onto the intersection of the feasible sets of all constraint.

This approach can be extended to learn hidden-variable models, where not all variable of the model are observed. If there are hidden variables, then the maximum-likelihood problem generally becomes non-convex and may exhibit multiple local minima. The *expectation-maximization algorithm* [68] is an iterative two-step procedure. At each iteration of the algorithm: (1) The *E-step* determines a concave lower-bound of the log-likelihood function that is tight for the current model estimate. This involves inference calculations to compute marginal distributions of hidden variables and their coupling to adjacent variables. (2) The *M-step* maximizes this lower-bound to obtain the next set of model parameters. This procedure is then repeated for this new set of model parameters and continues until a fixed point of the algorithm is reached. This method is guaranteed to monotonically increase the log-likelihood and to converge to a local maximum. The maximization step can be solved using the same inference and

convex optimization methods as are used to solve the maximum-likelihood problem when there are no hidden variables.

Structure Learning

Often, we may not know the correct graph structure to use for modeling some collection of random variables. Then, it is natural to seek a good graph structure based on sample data. Here, one is faced with the problem of *over-fitting*. That is, if we allow a very complex graph (with many edges and associated potential functions), this tends to overfit the sample data, leading to poor generalization performance when we compare the learned model to new sample data. Thus, one must find ways to regularize the model selection to penalize overly complex models. Another concern is that denser graphical models tend to be less tractable for inference calculations, providing further motivation for seeking less complex graphs.

One approach is to add a penalty term to the maximum log-likelihood objective which explicitly favors low-parameter models, as in the Akaike information criteria [2, 174] which uses the ℓ_0-norm of the parameter vector as a measure of model complexity. In the Gibbs representation of a graphical model, where the model parameters correspond to interactions between variables, this is essentially equivalent to favoring sparse graphs. However, the ℓ_0-regularized problem is non-convex and generally intractable to solve for large numbers of variables. Nonetheless, a number of incremental greedy feature selection methods have been developed which aim to approximately solve this model selection problem [64, 67, 69, 199].

Another approach is instead to restrict oneself to some specified set of low-complexity graphs. This approach is also combinatorial in nature and cannot generally be solved exactly by a tractable method. One exception, however, is the case of finding the best tree. This can be formulated as a maximum-weight spanning tree problem that can be solved exactly using a greedy method [56]. Unfortunately, the generalization to finding maximum-likelihood bounded tree-width graphs is NP-complete and one must again resort to approximation methods [131].

Recently, several methods have appeared that use ℓ_1-regularization to favor sparser graphical models [9,147,214]. This may be viewed as a convex proxy for ℓ_0-regularization. A dual interpretation of such methods is provided by the regularized maximum entropy method [72]. In particular, ℓ_1-regularized maximum-likelihood is the dual problem associated with finding the maximum entropy distribution over the set of probability distributions where the expected values of the sufficient statistics of the model are close to the sample average in the ℓ_∞ distance metric. This may also be viewed as *robust maximum-likelihood estimation* [9], which allows for uncertainty of the empirical moments. It is noteworthy that relaxing the parameter estimation in this way automatically leads to selection of sparse graphical models obtained by solving a convex optimization problem. This is also a critical feature in our approach.

1.3 Contributions

1.3.1 Lagrangian Relaxation for MAP Estimation

We develop a general Lagrangian relaxation (LR) approach to MAP estimation based on the idea of decomposing an intractable graphical model into a collection of tractable sub-models (e.g., defined on small subgraphs or on thin subgraphs such as trees), and study the conditions for strong duality to hold in this relaxation. For discrete variable models, we develop an algorithmic approach for solving the resulting dual problem based of a finite-temperature smoothing technique (using a deterministic annealing procedure to gradually reduce the temperature) and the iterative scaling algorithm to minimize a smoothed version of the dual function at each temperature. Additionally, we develop heuristic methods to either (i) enhance the relaxation to include additional structure so as to reduce or eliminate the duality gap, or (ii) provide approximate solutions in cases where it is not tractable to eliminate the duality gap.

While our work clearly has many parallels and connections to prior and ongoing work, there are a number of important innovative aspects in our approach that we now emphasize:

1. Formally relating various decomposition strategies to the classical concept of Lagrangian relaxation serves both to unify and simplify this body of work. For instance, it shows that several recently developed optimality conditions from this literature [93, 211] can all be viewed as instances of the well-known property of Lagrangian relaxation [22] that, when there exists a set of Lagrange multipliers such that all relaxed constraints are satisfied in the optimal solution of the dual problem, then there is no duality gap and the optimal primal solution is obtained.

2. Introducing, in an appropriate way, the finite-temperature relaxation method to "smooth" the non-smooth Lagrangian dual function leads to a very simple class of convergent, distributed algorithms that can successfully solve the dual problem. This involves also gradually reducing the temperature, which may be interpreted as an *interior-point* method for solving the primal version of linear-programming relaxation where entropy is used as a barrier function. The role of entropy as a barrier function function has also been noted in the variational interpretation of convex forms of belief propagation [219]. This is also similar to *entropic regularization* methods for solving min-max and linear programming problems [70, 150]. Although derived from different principles, the entropic regularization method leads to algorithms that are similar to an augmented Lagrange multiplier method due to Bertsekas [22].

3. This finite-temperature approach leads to a surprising connection to the classical *iterative scaling* procedure, typically used to fit a graphical model to data. We show that an appropriate version of the iterative scaling method is equivalent to *block coordinate-descent* [24] on our smoothed version of the Lagrangian dual

function. This leads to a simple message-passing algorithm that solves the dual problem in a distributed, iterative fashion. Each descent step involves passing messages between overlapping subgraphs to force the marginal distribution of shared nodes to be equivalent in each subgraph.

4. An added benefit of the deterministic annealing strategy is that it offers new possibilities to obtain approximate solutions in cases where, in the zero-temperature limit, there is a duality gap and the optimal dual decomposition becomes totally uninformative. We present a simple, heuristic approach for binary models that, at each temperature, assigns each variable to maximize its marginal distribution (output by the marginal matching procedure). This estimate is used to seed a greedy "bit-flipping" algorithm to obtain an improved estimate. Then, over all temperatures, we select the best estimate. The simple method has demonstrated remarkable performance on a wide range of problems.

5. Finally, while other work (with the notable exception of recent work of Sontag et al) has focused mainly on the simplest possible pairwise relaxations (or equivalent tree-based relaxations), we are finding that in many hard problems it is critical to introduce higher-order structure of the model to obtain strong duality. This extension is very straight-forward in our approach, both theoretically and in practice. Moreover, we are finding that a simple heuristic method, based on looking for frustrated cycles in a graphical representation of the resulting optimal dual decomposition and adding these cycles into the decomposition, usually leads to strong duality in the applications that we consider.

6. Similar to recent work [198], that builds on earlier work of Barahona [13], we develop an adaptive method to enhance our Lagrangian relaxation formulation by including additional subgraphs in the relaxation. This method is developed for binary variable models and is based on the idea of examining if the set of MAP estimates on each component of the relaxed graphical model are consistent in that there exists a global configuration which simultaneously is optimal on each subgraph. Although this condition is generally difficult to verify, we suggest an approach that only checks this consistency among the set of pairwise edges. In the case of zero-field Ising models, this reduces to checking for inconsistent cycles in which there are an odd number of edges on which the two-node MAP estimates all have opposite states and where the remaining edges of the cycle have MAP estimates that always have the same state value. This is consistent with the results of Barahona, which also checks for inconsistent cycles, although in a different sense. Our method is based on the fact that testing for strong duality can be viewed as a constraint satisfaction problem and the 2-SAT problem is tractable to solve using linear-time algorithms [7].

7. We also generalize the decomposition method and iterative scaling algorithm to a certain class of *Gaussian graphical models* [67, 145, 185, 199]. Specifically, we

consider Gaussian models that can be decomposed into a set of positive-definite quadratic functions defined on small subsets of variables. This condition generalizes the *pairwise-normalizability* condition, where the objective decomposes into pairwise positive-definite quadratic functions, that is a sufficient condition for convergence of a wide range of iterative estimations methods that submit to a *walk-sum* interpretation [45, 47, 124, 157]. It is straight-forward to implement our LR approach on this broader class of models, and we demonstrate that our iterative scaling method converges and that strong duality holds so that the correct MAP estimate is obtained. We also show that the solution of the LR problems also leads to a set of upper-bounds on the variance of each variable (conditioned on any observations).

8. Finally, we use the Gaussian model to demonstrate a more general form of LR, and of the iterative scaling algorithm, which allows us to formulate and solve a richer class of *multiscale* relaxations of the MAP problem. In the Gaussian case, this multiscale approach helps to accelerate the rate of convergence of our distributed, iterative optimization algorithms that we use to solve the dual problem. We also expect that this perspective will lead to new relaxations of the MAP problem for discrete problems.

■ 1.3.2 Maximum Entropy Relaxation for Learning Graphical Models

Based on the well-known maximum entropy principle [59, 97, 117, 177] and its interpretation as information projection in the exponential family of statistical models [3, 5, 15, 53, 74, 166], we propose a new *relaxed* version of the maximum entropy modeling paradigm. In this relaxed problem, which we refer to as *maximum entropy relaxation* (MER), the usual linear constraints on the moments of a distribution (the expected value of a specified set of features) are replaced by convex non-linear constraints based on relative entropy between the marginal distributions of subsets of variables and their empirical distributions obtained from sample data. Importantly, this provides a *convex optimization approach* to learning both the model parameters *and* the graph structure. The main features and innovative aspects of our approach are now summarized:

1. We develop an efficient *primal-dual interior point method* [37] for solution of the (primal) MER problem that exploits *chordal embedding* and the *sparse Fisher information matrix* in chordal graphs. This uses similar ideas as in several recent approaches to efficiently compute the information projection to a graphical model [64], including our own work in [126]. However, our approach here is developed also for discrete graphical models (e.g., binary variable models) and solves a more general class of relaxed maximum-entropy problems.

2. We derive a dual form of MER and show that this is a regularized version of the maximum-likelihood criterion, where graphical sparsity is enforced through

an information-regularization function. We note that while our relaxation approach has some parallels to recent works on regularized maximum entropy [72] and ℓ_1-regularization methods to obtain sparse graphical models [9,147,214], our approach is distinguished by the fact that it is entirely *information-theoretic* in nature, with both the objective and the constraints being based on natural information-theoretic measures.

3. A consequence of our information-theoretic formulation is that the MER solution is *invariant to reparameterizations* of the exponential family model. That is, while the solution certainly does depend on the choice of exponential family, it does not depend on which of many possible parameterizations of this family we might happen to use. This is *not* the case for any of the regularization methods that have been considered previously. We consider this an essential property, since the best choice of model should not change due to simply re-parameterizing the model.

4. Finally, we develop a *relaxed iterative scaling* approach to solve MER using a simple local update procedure. We show that this procedure performs block coordinate-descent in the MER dual problem. This results in a simple modification of the classical iterative scaling algorithm, one which automatically thins the graphical model.

1.4 Organization

Chapter 2: Background

We begin by presenting an overview of the relevant literature on graphical models, exponential families and variational principles related to inference and learning in these models. We then specialize the recursive inference method to the MAP estimation problem, and review other approaches to MAP estimation from the combinatorial optimization literature. Finally, we summarize the iterative scaling algorithm for learning graphical models and recent work on learning a good graph structure from sample data.

Chapter 3: Discrete Lagrangian Relaxation

In the first chapter on Lagrangian relaxation, we focus on the important case of graphical models with discrete (e.g., binary valued) variables. We develop our general approach for decomposing intractable graphical models, solving the dual problem and adaptively enhancing the formulation in cases in which there is a duality gap. We present examples involving the so-called "frustrated" Ising model from statistical physics.

Chapter 4: Gaussian Lagrangian Relaxation

In this chapter, we present the Gaussian version of Lagrangian relaxation and the appropriate information-form of the iterative scaling procedure. Here, we also present

a multiscale version of Lagrangian relaxation, with the aim of accelerating convergence of the iterative scaling algorithm in large GMRFs with long-range correlations. We demonstrate these methods on some examples using the thin-plate and thin-membrane models commonly used in image processing are remote-sensing applications.

Chapter 5: Maximum Entropy Relaxation

Lastly, we present the maximum entropy relaxation framework for learning graphical models. We handle both discrete and Gaussian models in this chapter, and present both primal and dual forms of MER. A simulation study is presented to analyze the ability of MER to recover the graphical structure of a model from sample data.

Chapter 6: Conclusion

In closing, we summarize our research and propose possible directions for further research and development that are suggested by this work.

Chapter 2

Background

2.1 Preamble

In this chapter, we provide some background on graphical models and relevant methods of inference, optimization and learning. The chapter is organized as follows: *Section 2.2* reviews basic definitions of graph theory, introduces graphical models and Gibbs distributions, and discusses their interpretation as Markov models; *Section 2.3* summarizes some useful facts about exponential families, the maximum entropy principle, Gibbs variational principle and information geometry; *Section 2.4.1* reviews standard approaches to exact inference in tractable graphical models and the approximate method of loopy belief propagation; *Section 2.5* discusses variants of these methods for MAP estimation in graphical models, and other approaches based on classical combinatorial optimization techniques; *Section 2.6* discusses inference and MAP estimation in Gaussian graphical models; and *Section 2.7* summarizes some standard approaches to learning graphical models.

Notational Conventions

We presume the reader is familiar with basic set theory, probability theory and vector calculus. We remind the reader of some standard notations below.

We use standard set-theoretic notation: $A \cup B$ is the union of two sets, $A \cap B$ is the intersection, $A \setminus B$ is the set difference. Let $\emptyset$ denote the empty set. The set of real numbers is denoted by $\mathbb{R}$. We say that the set A *contains* its elements $a \in A$, and *includes* its subsets $B \subset A$. We write $A \subsetneq B$ to indicate that A is a *proper* subset of B ($A \subset B$ and $A \neq B$). Given two sets $\mathbb{X}$ and $\mathbb{Y}$ we write $\mathbb{X} \otimes \mathbb{Y}$ for the Cartesian product set $\{(x, y) | x \in \mathbb{X} \text{ and } y \in \mathbb{Y}\}$. Also, $\mathbb{X}^n$ denotes the set of all n-tuples drawn from $\mathbb{X}$ and we write $\mathbb{X}^A$ for the set of all maps from A to $\mathbb{X}$. We write $2^A \triangleq \{0, 1\}^A$ to denote the set of all subsets of A and write $\binom{A}{k}$ to denote the set of all k-element subsets of A.

Given a probability distribution $P(x) \geq 0$ of a discrete random variable $x \in \mathbb{X}$, which satisfies $\sum_{x \in \mathbb{X}} P(x) = 1$, we define the *expected value* of a function $f(x)$ as $\mathbb{E}_P[f] \triangleq \sum_{x \in \mathbb{X}} P(x) f(x)$. For continuous variables, $P(x) \geq 0$ represents a *probability density*, which satisfies $\int P(x) dx = 1$, and we define $\mathbb{E}_P[f] \triangleq \int P(x) f(x) dx$. Given the joint distribution $P(x, y)$ of two random variables, the *marginal distribution* of x is defined by $P(x) = \sum_y P(x, y)$ for discrete variables and by $P(x) = \int P(x, y) dy$ for continuous

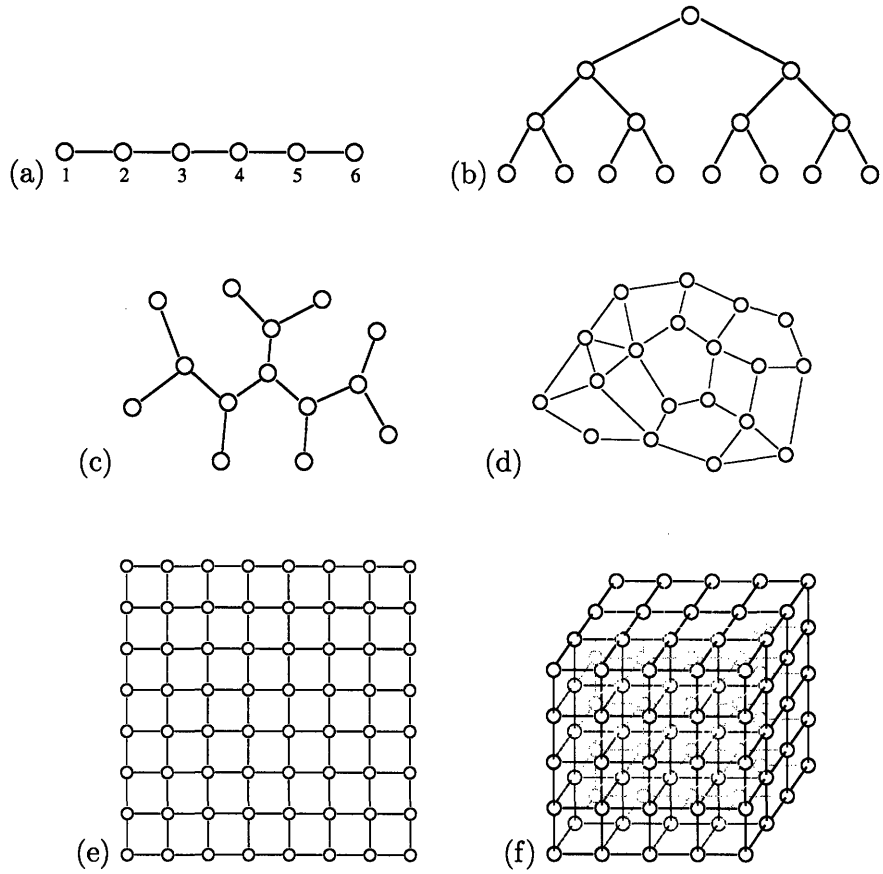

Figure 2.1. Drawings of several graphs. (a) chain, (b) hierarchical tree, (c) irregular tree, (d) planar graph, (e) square lattice (also planar), (f) cubic lattice. In (a), we explicitly label the vertices $V = \{1, 2, 3, 4, 5, 6\}$. The edges of this graph are $\mathcal{G} = \{$ {1,2}, {2,3}, {3,4}, {4,5}, {5,6} $\}$.

variables. The *conditional distribution* of x given y is defined $P(x|y) = P(x, y)/P(y)$ for all y such that $P(y) > 0$.

We may define a matrix A to have matrix elements a_{ij} by writing $A = (a_{ij})$. Given a function $f(\theta) = f(\theta_1, \ldots, \theta_d)$ of parameters θ, we define the *gradient* of f as $\nabla f(\theta) = \left(\frac{\partial f(\theta)}{\partial \theta_i}\right)$. The *Hessian* of f is defined as the matrix of second derivatives: $\nabla^2 f(\theta) = \left(\frac{\partial^2 f(\theta)}{\partial \theta_i \partial \theta_j}\right)$. Given a vector map $\Lambda : \mathbb{R}^d \to \mathbb{R}^d$, we define the *Jacobian* as $\partial\Lambda(\theta) = \left(\frac{\partial \Lambda_i(\theta)}{\partial \theta_j}\right)$. A set $\mathbb{X} \subset \mathbb{R}^d$ is *convex* if $\lambda x + (1 - \lambda)y \in \mathbb{X}$ for all $x, y \in \mathbb{X}$ and $0 \leq \lambda \leq 1$. A function $f : \mathbb{X} \to \mathbb{R}$ is *convex* if $f(\lambda x + (1 - \lambda)) \leq \lambda f(x) + (1 - \lambda) f(y)$. It is *strictly convex* if $f(\lambda x + (1 - \lambda)) < \lambda f(x) + (1 - \lambda) f(y)$ for all $x \neq y$ and $0 < \lambda < 1$.

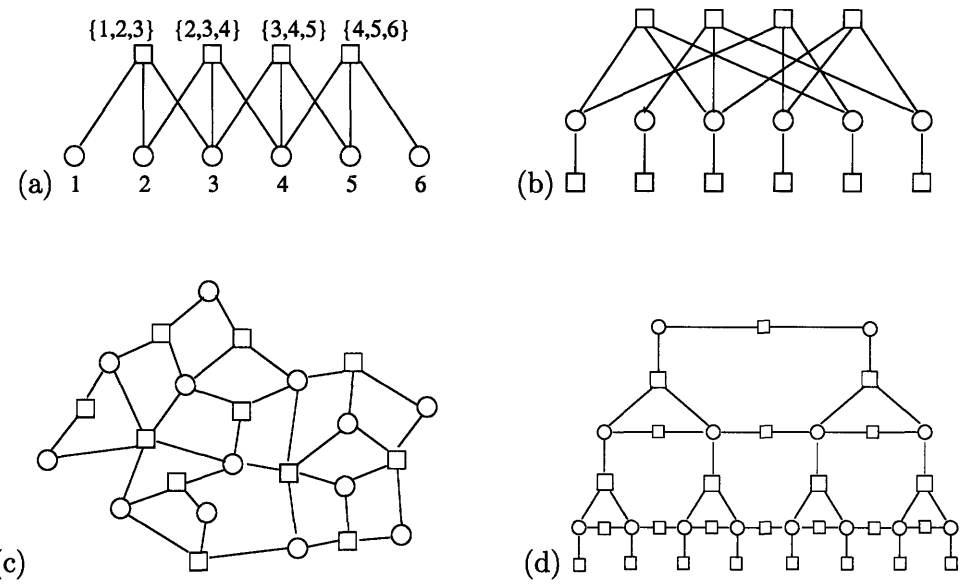

Figure 2.2. Drawings of several hypergraphs (using the factor graph representation). (a) a 3rd-order chain, (b) irregular 3rd-order edges and singleton edges, (c) irregular hypergraph, (d) hierarchical hypergraph having 3rd-order edges between levels, pairwise edges within each level and singleton edges at the bottom level. In (a), we explicitly label the vertices $V = \{1, 2, 3, 4, 5, 6\}$. The edges of this graph are $\mathcal{G}$ = { {1,2,3}, {2,3,4}, {3,4,5}, {4,5,6} }

■ 2.2 Introduction to Graphical Models

■ 2.2.1 Graphs and Hypergraphs

Although we do not require very much graph theory, the *language* of graphs is essential to the thesis. We give a brief, informal review of the necessary definitions here, mainly to establish conventions used throughout the thesis. Readers who are unfamiliar with these concepts may wish to consult the references [20, 33, 105]. A *graph* is defined by a set of *vertices* $v \in V$ (also called the *nodes* of the graph) and by a set of *edges* $E \in \mathcal{G}$ defined as subsets (e.g., pairs) of vertices.[1] Edges are often defined as unordered[2] pairs of vertices $\{u, v\} \in \mathcal{G}$. Such *pairwise* graphs $\mathcal{G} \subset \binom{V}{2}$ are drawn using circle nodes to denote vertices and lines drawn between these nodes to denote edges. Several such drawings of pairwise graphs are shown in Figure 3.3. We also allow more general definitions of graphs $\mathcal{G} \subset 2^V \setminus \emptyset$, also known as *hypergraphs* [20], for which edges (also called *hyperedges*) may be defined as *any* subset of one or more vertices. To display such a generalized graph, it is often convenient to represent it using diagrams such as

[1] We deviate somewhat from standard notation $\mathcal{G} = (V, \mathcal{E})$ where $\mathcal{G}$ denotes the graph and $\mathcal{E}$ denotes the edge set of the graph. We instead use $\mathcal{G}$ to denote both the graph and its edge set, as the vertex set V should be apparent from context.

[2] This definition is for *undirected* graphs. It is also common to define *directed* graphs, with edges defined as *ordered* pairs $(u, v) \in \mathcal{G}$. A directed edge (u, v) is drawn as an arrow pointing from node u to node v. We focus mainly on undirected graphs in this thesis.

seen in Figure 2.2. In these diagrams, each circle again represents a vertex $v \in V$ of the graph but we now use square markers to denote each edge $E \in \mathcal{G}$. The structure of $\mathcal{G}$ is encoded by drawing lines connecting each edge $E \in \mathcal{G}$ to each of its vertices $v \in E$. There is one such connection for each pair $(v, E) \in V \times \mathcal{G}$ such that $v \in E$. Such representations are called *factor graphs* in the graphical modeling and coding literatures [85, 153].

(Generalized) Graph Convention Unless otherwise stated, when we refer to a graph $\mathcal{G}$ or an edge $E \in \mathcal{G}$, then it should be understood that $\mathcal{G}$ may be a generalized graph (a hypergraph) and E may be any subset of one or more edges (a hyperedge). This includes the usual definition of pairwise graphs as a special case, and most of our examples and illustrations do use pairwise graphs to illustrate the basic ideas. Allowing $\mathcal{G}$ to possibly be a hypergraph in general allows us to express the general case without having to always use the more cumbersome terminology of "hyergraph" and "hyperedge" throughout the thesis. If it is essential that a given graph is actually a pairwise graph, then we explicitly say so. We occasionally remind the reader of this convention by referring to $\mathcal{G}$ as a "(generalized) graph".

We now define some basic graphical concepts. Note, although these definition are often presented for pairwise graphs, the definitions given here also apply for generalized graphs (unless otherwise noted). A *subgraph* of $\mathcal{G}$ is defined by a subset of vertices $V_{\text{sub}} \subset V$ and a subset of edges $\mathcal{G}_{\text{sub}} \subset \mathcal{G}$ such that each edge is included in the vertex set (we also say that $\mathcal{G}$ is a *supergraph* of $\mathcal{G}_{\text{sub}}$). Unless otherwise stated, the vertex set is defined by the union of the edges of the subgraph. The *induced subgraph* $\mathcal{G}_A$ based on vertices $A \subset V$ is defined as the set of all edges of $\mathcal{G}$ that contain only vertices in A. A *clique* is a completely connected subset of nodes, that is, a set $C \subset V$ such that each pair of nodes $u, v \in C$ are connected by an edge, that is, $u, v \in E$ for some $E \in \mathcal{G}$. A *path* of length ℓ is a sequence of nodes $(v_0, \ldots, v_\ell)$ and edges $(E_1, ..., E_\ell)$ such that no node or edge is repeated (except possibly $v_0 = v_\ell$) and consecutive nodes (v_k, v_{k+1}) are contained in their corresponding edge E_k. This path *connects* nodes v_0 and v_ℓ. If $v_0 = v_\ell$ we say that the path is *closed*. A graph is *connected* if any two nodes may be connected by a path. The *connected components* of a graph are its maximal connected subgraphs. A *cycle* is a subgraph formed from the nodes and edges of a closed path. A *tree* is a connected, pairwise graph that does not contain any cycles (see Figures 3.3(a), (b) and (c)). A pairwise graph is *planar* if it can be drawn in the plane without any two edges intersecting (see Figures 3.3(d) and (e)).

Some additional definitions are presented as needed in later sections. Graph separators are defined in Section 2.2.3. Chordal graphs and junction trees are discussed in Section 2.4.1. Also, several canonical graphical problems (max-cut, max-flow/min-cut, maximum-weight independent sets and maximum perfect matching), which arise in connection with MAP estimation, are briefly discussed in Section 2.5.

■ 2.2.2 Graphical Factorization and Gibbs Distribution

Let $x = (x_1, \ldots, x_n) \in \mathbb{X}^n$ be a collection of variables where each variable ranges over the set $\mathbb{X}$.[3] For example, a binary variable model is given by $\mathbb{X} = \{0, 1\}$ and a continuous variable model has $\mathbb{X} = \mathbb{R}$. We define a *graphical model* [43, 60, 145] as a probability model defined by a (generalized) graph $\mathcal{G}$ with vertices $V = \{1, \ldots, n\}$, identified with variables $x_1, \ldots, x_n$, and probability distributions of the form

$$P(x) = \frac{1}{Z(\psi)} \prod_{E \in \mathcal{G}} \psi_E(x_E) \tag{2.1}$$

where each $\psi_E : \mathbb{X}^E \to \mathbb{R}$ is a non-negative function of variables $x_E = (x_v, v \in E)$ and $Z(\psi)$ is a normalization constant.[4] We call the individual functions ψ_E the *factors* of the model. In the factor graph representation (Figure 2.2), each circle node $v \in V$ represents a variable x_v and each square node $E \in \mathcal{G}$ represents one of the factors ψ_E.

For strictly positive models ($P(x) > 0$ for all x) the probability distribution may be equivalently described as a *Gibbs distribution* of statistical physics [90, 129, 173, 195], expressed as

$$P(x) = \frac{1}{Z(f, \beta)} \exp\left\{\beta \sum_{E \in \mathcal{G}} f_E(x_E)\right\} \tag{2.2}$$

where $f(x) = \sum_E f_E(x_E)$ is the *energy* function (or *Hamiltonian*) and the individual terms $f_E(x_E)$ are called *potential functions* (or simply *potentials*) of the model.[5] The parameter $\beta \geq 0$ is the *inverse temperature* of the Gibbs distribution and

$$Z(f, \beta) \triangleq \sum_{x \in \mathbb{X}^n} \exp\left\{\beta \sum_{E \in \mathcal{G}} f_E(x_E)\right\} \tag{2.3}$$

is the *partition function*, which serves to normalize the probability distribution. Evidently, the probability distributions defined by (2.1) and (2.2) are equivalent if we take $\psi_E(x_E) = \exp\{f_E(x_E)\}$ (and $\beta = 1$). In statistical physics, the *free energy* is defined as $\mathcal{F}(\theta, \beta) \triangleq \beta^{-1} \log Z(f, \beta)$, which (for $\beta = 1$) is also called the *log-partition function* in the graphical modeling literature. Later, in Section 2.3.2, we discuss the relation of this quantity to Gibbs free energy . The *temperature* $\tau = \beta^{-1}$ may be viewed as a parameter that, for a fixed energy function $f(x)$, controls the level of randomness of

[3] More generally, each variable may have a different range of values $\mathbb{X}_i$ such that $x \in \mathbb{X}_1 \otimes \mathbb{X}_2 \cdots \otimes \mathbb{X}_n$.

[4] We use the notational convention that whenever we define a function of variables $x = (x_v, v \in V)$ in terms of functions defined on subsets $S \subset V$ of these variables, we use $x_S = (x_v, v \in S)$ to denote a subset of the variables x (x and x_A are *not* independent variables). Likewise, $f_A(x_A) + f_B(x_B)$ should be regarded as a function of the variables $x_{A \cup B}$. If $S \triangleq A \cap B$ is non-empty, then the variables x_S are shared by f_A and f_B.

[5] For notational convenience, our definition of energy and potential functions are negated versions of what is normally used in physics (the definition of the Gibbs distribution normally includes a minus sign in the exponent).

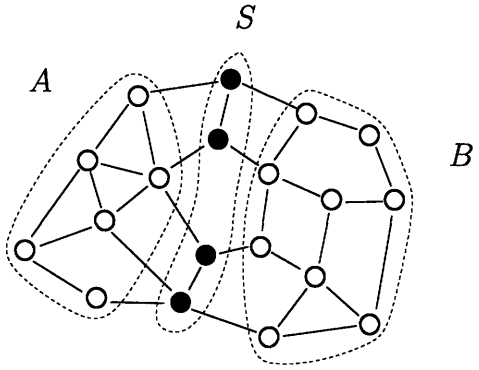

Figure 2.3. Example of a graph separator S (the filled nodes). Also, S separates the node sets A and B (there is no path from A to B that does not pass through S). The *Markov property* implies that x_A and x_B are conditionally independent given x_S, that is, $P(x_A, x_B|x_S) = P(x_A|x_S)P(x_B|x_S)$.

the Gibbs distribution. At high temperatures, the Gibbs distribution becomes approximately uniformly distributed over all configurations $x \in \mathbb{X}^n$. At low temperatures, the probability distribution becomes concentrated on just those configurations $x \in X^n$ for which $f(x)$ is close to the maximum value $f^* \triangleq \max f$.

We emphasize that the precise choice of potential functions that give rise to a specific distribution $P(x)$ is *not* unique. Many different choices of individual potential functions lead to exactly the same probability distribution. There are two reasons for this degeneracy. First, due to the normalization of $P(x)$, we may add any constant to the energy $f(x)$ and it does not change $P(x)$. Moreover, for a fixed energy function $f(x)$ there are many ways it can be split into a set of potentials $f(x) = \sum_{E \in \mathcal{G}} f_E(x)$. For instance if two edges $A, B \in \mathcal{G}$ share nodes $S = A \cap B \neq \emptyset$ then we may *add* an arbitrary function $\lambda(x_S)$ to one potential $f'_A(x_A) = f_A(x_A) + \lambda(x_S)$ and *subtract* this same function from the other potential $f'_B(x_B) = f_B(x_B) - \lambda(x_S)$, and it leaves the overall potential *unchanged* because $f'_A + f'_B = (f_A + \lambda) + (f_B - \lambda) = f_A + f_B$. Such changes of representation do not effect the overall distribution and are sometimes called *reparameterizations* of the model [210].

■ 2.2.3 Markov Property: Separators and Conditional Independence

A graphical model satisfies a certain set of conditional independence relations with respect to its graph. A subset of vertices $S \subset V$ is said to be a *separator* of the graph $\mathcal{G}$ if removing these nodes (and all edges that contain any of these nodes) disconnects some part of the graph (such that the number of connected components is increased). Also, we say that S *separates* two vertex sets $A, B \subset V$ if there is no path connecting A and B that does not pass through S. These definitions are illustrated in Figure 2.3. A probability distribution $P(x)$ is said to be *Markov* with respect to $\mathcal{G}$ if for all (S, A, B), where S separates A from B, it holds that x_A and x_B are *conditionally independent* given

x_S, that is, if $P(x_A, x_B|x_S) = P(x_A|x_S)P(x_B|x_S)$. It is simple to verify this property for a graphical model defined on $\mathcal{G}$. Hence, graphical models are also called *Markov models* or *Markov random fields* (MRFs). For a set of vertices $A \subset V$, let ∂A denote the set of nodes not included in A that are linked to A by some edge. This is called the *Markov blanket* of A, as the Markov property implies $P(x_A|x_{V\setminus A}) = P(x_A|x_{\partial A})$.

The *Hammersley-Clifford theorem* [40,99,104] states that essentially all probability distributions that are Markov with respect to a graph may be represented as a Gibbs distribution defined on this graph. That is, if $P(x) > 0$ for all $x \in \mathbb{X}^n$ and $P(x)$ is Markov on $\mathcal{G}$, then there exists some set of potential functions defined on the cliques of $\mathcal{G}$ such that $P(x)$ can be represented as a Gibbs distribution (2.2). In fact, we may explicitly construct such a potential specification from the conditional distributions of $P(x)$ as follows: Given $P(x)$, we recursively define potential functions on *every* subset of nodes $A \subset V$, based on the conditional probability distribution $P(x_A|x_{V\setminus A})$ and all potentials defined on proper subsets of A:

$$f_A(x_A) = \log P(x_A|x_{V\setminus A} = 0) - \sum_{B \subsetneq A} f_B(x_B) \tag{2.4}$$

Here, we have assumed that $0 \in \mathbb{X}$ (but any other element of $\mathbb{X}$ could have been chosen instead) and write $x_{V\setminus A} = 0$ to indicate $x_v = 0$ for all $v \notin A$. In this construction, we have defined potentials on every subset of nodes (not just the edges or cliques of $\mathcal{G}$). However, it can be shown [40] that if P is Markov on $\mathcal{G}$ and E is not a clique of $\mathcal{G}$ then the conditional independence property implies that $f_A(x_A) = 0$ for all $x_A \in \mathbb{X}^A$. Thus, dropping these zero potentials, we actually obtain a compact representation in terms of potential functions defined *only* on the cliques of $\mathcal{G}$. Then, taking $A = V$ in (2.4) and solving for $P(x)$ gives:

$$P(x) = \exp\left\{ \sum_{C \in \mathcal{C}(\mathcal{G})} f_C(x_C) \right\} \tag{2.5}$$

This defines a graphical model with respect to the generalized graph defined by $\mathcal{C}(\mathcal{G})$, the set of all cliques of $\mathcal{G}$. Also, this particular set of potentials satisfy $Z(f) = 1$. We again remark that this representation is *not* unique, many equivalent potential representations are possible. In particular, we may group potentials together such that only *maximal* cliques (i.e., cliques not contained by a larger clique) are used in this representation. Another interesting point to note is that the potential specification of a Markov model on $\mathcal{G}$ is determined by the *conditional specification* over $\mathcal{G}$ [71], that is, by a consistent set of conditional probability distributions $P(x_E|x_{\partial E})$ for all $E \in \mathcal{G}$. Moreover, it is trivial to recover these conditional distributions from any set of potential functions, so that the two specifications are essentially equivalent. Later, in Section 2.3.1, we find that *marginal specifications*, defined to be any set of consistent marginal distributions, $P(x_E)$ for all $E \in \mathcal{G}$, play an equally important complementary role in the theory of graphical models.

■ 2.3 Exponential Family Models and Variational Principles

We also consider parameterized families of graphical models in the form of an *exponential family* [15, 52, 74]:

$$P(x;\theta) = \exp\{\theta^T \phi(x) - \Phi(\theta)\}. \tag{2.6}$$

Here, $\phi : \mathbb{X}^n \to \mathbb{R}^d$ are the *features* used to define the family (also called *sufficient statistics*) and $\Phi(\theta)$ is the *cumulant generating function* of the family, which serves to normalize the distribution (analogous to the log-partition function in the previous section). In discrete models, we have

$$\Phi(\theta) = \log \sum_{x \in \mathbb{X}^n} \exp\{\theta^T \phi(x)\} \tag{2.7}$$

In continuous models, $\Phi(\theta)$ is defined by an integral rather than a sum. We define $\Theta = \{\theta \in \mathbb{R}^d | \Phi(\theta) < \infty\}$, the set of all θ such that $\Phi(\theta)$ is well-defined so that we may define a normalized probability model. For instance, $\Phi(\theta)$ might diverge in discrete models with infinitely many states $\mathbb{X} = \{0, 1, 2, \dots\}$ or in continuous variable models $\mathbb{X} = \mathbb{R}$. In discrete models with finitely many states, such as binary variable models, we always have $\Theta = \mathbb{R}^d$. In general, Θ is a convex, open set in $\mathbb{R}^d$.

The exponential family $\mathcal{F}$, based on a set of features ϕ, is defined as the set of all normalizable probability distributions of this form $\mathcal{F} = \{P_\theta | \theta \in \Theta\}$. Note that different choices of features may lead to the *same* family of probability distributions. For example, for any invertible $d \times d$ matrix A, the features $\phi' = A\phi$ provide another representation of the same family. We say that a set of features is *minimal* if the the family cannot be represented using fewer features. This is equivalent to requiring that no feature can be expressed as a linear combination (plus a constant) of the other features for all $x \in \mathbb{X}^V$. Then, there is a one-to-one correspondence between parameters $\theta \in \Theta$ and probability distributions $P_\theta \in \mathcal{F}$.

For a given set of features, the *moments* of a probability distribution P are defined to be the expected value of the features $\eta = \mathbb{E}_P[\phi] \in \mathbb{R}^d$. We let $\mathcal{M} = \{\eta = \mathbb{E}_P[\phi], P \in \mathcal{F}\}$ denote the set of moments generated by the family $\mathcal{F}$. The set of *all* realizable moments, generated by arbitrary distributions P (not restricted to $\mathcal{F}$) is equal to the closure $\bar{\mathcal{M}}$, defined as the set of all limit points of $\mathcal{M}$. The boundary $\partial\mathcal{M} \triangleq \bar{\mathcal{M}} \setminus \mathcal{M}$ corresponds to degenerate probability distributions that encode hard constraints (e.g., discrete distributions with $P(x) = 0$ for some x). Such degenerate distributions are not contained in the exponential family representation because there is no (finite) θ that realizes these distributions. We generally assume that we are working with non-degenerate distributions to avoid this technicality.

As in [213], we consider exponential families that define graphical models by using features that only depend on subsets of variables. We use $\alpha, \beta \in \mathcal{I}$ to index features of the model, such that the entire feature vector is $\phi = (\phi_\alpha, \alpha \in \mathcal{I})$. Also, let E_α denote the subset of variables that are used to define ϕ_α, i.e. $\phi_\alpha(x) = \phi_\alpha(x_{E_\alpha}) \triangleq \phi_\alpha(x_\alpha)$. Thus, we obtain a parametric family of graphical models defined on the graph $\mathcal{G} = \cup_\alpha E_\alpha$ with

energy function $f(x) = \sum_\alpha f_\alpha(x_\alpha) = \sum_\alpha \theta_\alpha \phi_\alpha(x_\alpha)$.[6]

Discrete Models

A *Boltzmann machine* [4] is a binary variable model, where each variable may take on values in $\mathbb{X} = \{0, 1\}$. The energy function of the model is

$$f(x; \theta) = \sum_{E \in \mathcal{G}} \theta_E \phi_E(x_E) \tag{2.8}$$

with features defined by products of variables:

$$\phi_E(x_E) = \prod_{v \in E} x_v. \tag{2.9}$$

In this case, there is a one-to-one correspondence between features and edges, so we have $\mathcal{I} = \mathcal{G}$. The moments are given by probabilities $\eta_E = \mathbb{E}_P[\phi_E] = P(\{x | x_v = 1 \text{ for all } v \in E\})$. Thus, each feature ϕ_E acts as an indicator for the event that all nodes $v \in E$ are set to $x_v = 1$ and η_E is the probability of this event. Most commonly, such models are defined using only node potentials $f_i(x_i) = \theta_i x_i$ and pairwise potentials $f_{ij}(x_i, x_j) = \theta_{ij} x_i x_j$. However, we allow arbitrary interactions among the variables so that the model could in principle represent an arbitrary function of $\mathbb{X}^n$ (up to an irrelevant additive constant).

The *Ising model* [10, 129] is defined similarly, except that the allowed states are labeled $\mathbb{X} = \{-1, +1\} \equiv \{-, +\}$. In statistical physics, each node represents a particle with an internal "spin" variable that is in either an "up" $(+)$ or "down" $(-)$ state. The choice of which binary encoding we use (Boltzmann versus Ising) is not too important, as one can easily convert between these two representations. However, it does change the interpretation of θ and η parameters. For example, in the Ising model we have $\eta_i = P_i(+) - P_i(-)$ and $\eta_{ij} = P_{ij}(++) + P_{ij}(--) - P_{ij}(+-) - P_{ij}(-+)$. The generalized Ising model (including interactions on larger subsets) is well-suited for describing parity-check codes, since θ_E expresses the bias favoring $\prod_{v \in E} x_v = +1$ over $\prod_{v \in E} x_v = -1$. We stress that both parameterizations are minimal and general enough to represent an arbitrary potential function.

Next, we consider the general q-state discrete models with $\mathbb{X} = \{0, \ldots, q-1\}$. In this case, it is sometimes convenient to use an *over-parameterized* representation of the model [210, 213], which means that the feature set is non-minimal and the mapping from θ to probability distributions $P \in \mathcal{F}$ is many-to-one. We may encode an arbitrary potential function $f_E(x_E)$ as follows. For each configuration $\tilde{x}_E \in \mathbb{X}^E$ of the variables in E, define one feature to be the indicator function for the event $x_E = \tilde{x}_E$:

$$\phi_{E, \tilde{x}_E}(x_E) = \begin{cases} 1, & x_E = \tilde{x}_E \\ 0, & \text{otherwise} \end{cases} \tag{2.10}$$

[6] Although this representation might define multiple potentials per edge, we may also group these together by edge: $f(x) = \sum_{E \in \mathcal{G}} f_E(x)$ with $f_E(x_E) = \sum_{\alpha : E_\alpha = E} f_\alpha(x_E)$.

In terms of these features we can parameterize the potential function $f_E(x_E)$ by simply enumerating all of its values:

$$f_E(x_E) = \sum_{\tilde{x}_E} \theta_{E,\tilde{x}_E} \phi_{E,\tilde{x}_E}(x_E) \tag{2.11}$$

Thus, indexing features by $\alpha = (E, \tilde{x}_E)$, we obtain an exponential family representation $f(x) = \sum_E f_E(x_E) = \sum_\alpha \theta_\alpha \phi_\alpha(x_\alpha)$. Of course, this is no more than a change of notation. If we instead rewrite θ_{E,x_E} as $\theta_E(x_E)$, then $f(x) = \sum_E \theta_E(x_E)$. Thus the parameters θ correspond directly to the potential specification of the model.

Similarly, the moment parameters specify the edge-wise marginal distributions of the model: $\eta_E(x_E) = P(x_E)$. We refer to the set of all edge-wise marginal distributions $\{P_E(x_E), E \in \mathcal{G}\}$ as the *marginal specification* of a graphical model. A marginal specification $\{P_E(x_E), E \in \mathcal{G}\}$ is *realizable* if there exists a joint distribution $P(x)$ which has these marginals. The set of all such realizable marginal specifications defines the *marginal polytope*. In the exponential family representation, this is precisely the set of realizable moments $\bar{\mathcal{M}}$. In the case of the over-parameterized representation, the marginal on edge E is directly specified by the moments $\eta_E(x_E)$. Using the minimal representations of the binary variable models (Ising or Boltzmann), the marginal $P(x_E)$ is determined by the subset of moment parameters $\eta_{[E]} \triangleq (\eta_{E'}, E' \subset E)$, that is, the set of all moments of variables within edge E.

Lastly, we remark that the "over-parameterized" representation is not strictly necessary. It is simple to obtain a minimal representation by setting $\theta_E(x_E) = 0$ for all configurations x_E where any of the variables x_v, for $v \in E$, are set to a particular value (for instance, let $\theta_E(x) = 0$ if $x_v = 0$ for any $v \in E$). The remaining free parameters then provide a minimal representation of the exponential family. In the case $q = 2$, this recovers the Boltzmann parameterization. However, for pedagogical purposes, it is often simpler to discuss the over-parameterized representation.

Gaussian Model

Finally, we consider *Gaussian* graphical models [67,157,185,199,201,217]. In this case we have $\mathbb{X} = \mathbb{R}$, and consider probability density functions of the form:

$$P(x) = \exp\{-\frac{1}{2}x^T Jx + h^T x - \Phi(h, J)\} \tag{2.12}$$

where $J \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix, called the *information matrix*, and $h \in \mathbb{R}^n$ is the *potential vector*. It is straight-forward to check that the mean of x is given by $\mu \triangleq \mathbb{E}_P[x] = J^{-1}h$ and the covariance of x is $K \triangleq \mathbb{E}_P[(x-\mu)(x-\mu)^T] = J^{-1}$. The constant $\Phi(h, J) = \log \int \exp\{-\frac{1}{2}x^T Jx + h^T x\} dx$ serves to normalize the density and may be calculated as

$$\Phi(h, J) = \frac{1}{2}\{-\log\det J + h^T J^{-1} h + n \log 2\pi\} \tag{2.13}$$

The fill-pattern of the matrix J defines the graphical structure of the model: $\{u, v\} \in \mathcal{G}$ if and only if $J_{ij} \neq 0$. The Gaussian analog of the Hammersley-Clifford theorem states that the class of Gaussian models that are Markov with respect to $\mathcal{G}$ is equivalent to the above family with $J_{ij} = 0$ for all $\{i, j\} \notin \mathcal{G}$ [199].

It is apparent how to translate this into an exponential family representation [67, 199]. We define linear features $\phi_v(x) = x_v$ and quadratic features $\phi_{v,v}(x) = x_v^2$ for all nodes $v \in V$ and interaction terms $\phi_{u,v}(x) = x_u x_v$ for all edges $\{u, v\} \in \mathcal{G}$. The index set $\mathcal{S}$ of these features may be defined as $S = V \cup \{(v, v), v \in V\} \cup \{(u, v) \in \mathcal{G}\}$. Then, components of h and J map to to components of $\theta = (\theta_\alpha, \alpha \in \mathcal{S})$:

$$\begin{aligned} \theta_v &= h_v \\ \theta_{v,v} &= -\tfrac{1}{2} J_{v,v} \\ \theta_{u,v} &= -J_{u,v} \end{aligned}$$

This definition of θ ensures that $\theta^T \phi(x) = -\frac{1}{2} x^T J x + h^T x$. The moment parameters $\eta = (\eta_\alpha, \alpha \in \mathcal{S})$ are similarly related to the mean μ and covariance matrix K:

$$\begin{aligned} \eta_v &= \mu_v \\ \eta_{v,v} &= K_{v,v} \\ \eta_{u,v} &= K_{u,v} \end{aligned}$$

Thus, this gives the exponential family representation of the Gaussian model.

■ 2.3.1 Maximum Entropy Principle

There is a one-to-one correspondence between elements of the set $\eta \in \mathcal{M}$ and probability distributions $P_\eta \in \mathcal{F}$. This is shown by the *maximum entropy principle* [59,97,117,177]. Let

$$H(P) = -\mathbb{E}_P[\log P(x)] = \mathbb{E}_P\left[\log \frac{1}{P(x)}\right] \tag{2.14}$$

denote the *entropy* of probability distribution $P(x)$. Entropy is a measure of uncertainty in the information theory of coding and communication [59, 86, 193]. Consider the following optimization problem:

$$H(\eta) \triangleq \begin{cases} \text{maximize} & H(P) \\ \text{subject to} & \mathbb{E}_P[\phi] = \eta \end{cases} \tag{2.15}$$

That is, among the set of *all* probability distributions (not restricted to $\mathcal{F}$), we seek the maximum entropy distribution (the least informative probability model) that is consistent with a specified set of moment constraints. Typically, when learning a model from sample data, the moments η are given by empirical averages. The maximum entropy principle states that the solution to this problem (when it exists and is strictly positive)[7] is an exponential family distribution based on features ϕ and with parameters θ

[7]The maximum entropy problem might fail to have an exponential family solution in one of two ways: (1) if η is not realizable by any probability distribution then the problem is infeasible, (2) if the

chosen to satisfy the condition $\mathbb{E}_\theta[\phi] = \eta$ (note that we use $\mathbb{E}_\theta$ to denote expectation with respect to the probability distribution P_θ). This can be derived from the perspective of Lagrangian duality [59], where the θ parameters arise as Lagrange multipliers associated with the moment constraints. This analysis also shows that the maximum value of the entropy is given by $H(\eta) = \Phi(\theta) + \eta^T\theta$, indicating a connection between the functions $H(\eta)$ and $\Phi(\theta)$. Also, simple gradient analysis shows that selecting θ to realize the empirical moments is equivalent to *maximum-likelihood estimation*. This shows a fundamental duality between the maximum entropy modeling and maximum-likelihood parameter estimation in exponential families.

Because entropy is a concave function of P, there is a *unique* probability distribution $P_\eta \in \mathcal{F}$ that solves the maximum entropy problem for each $\eta \in \mathcal{M}$. If the features ϕ are minimal, then there is also a unique θ corresponding to each $\eta \in \mathcal{M}$. Then, there is a one-to-one correspondence between the η and θ parameterizations of the exponential family. We denote the forward mapping from θ to η by $\Lambda : \Theta \to \mathcal{M}$, and (for minimal representations) the inverse mapping by Λ^{-1}. The forward mapping $\eta = \Lambda(\theta)$ corresponds to directly computing the moments of a specified distribution (by summation or integration) and also requires calculation of the normalization constants $\Phi(\theta)$. We refer to this as the *inference problem* (discussed further in Section 2.4.1). Also, a simple calculation verifies the following *moment generating property* of $\Phi(\theta)$:

$$\frac{\partial \Phi(\theta)}{\partial \theta_\alpha} = \mathbb{E}_\theta[\phi_\alpha] = \eta_\alpha \tag{2.16}$$

Thus, $\Lambda(\theta) = \nabla\Phi(\theta)$ and we see that inference is closely linked to computation of $\Phi(\theta)$. The inverse calculation $\theta = \Lambda^{-1}(\eta)$ does not generally have a direct solution method, and must usually be solved using iterative methods. We refer to this as the *learning problem* (discussed further in Section 2.7).

■ 2.3.2 Convex Duality and Gibbs Variational Principle

The maximum entropy principle indicates a fundamental duality between the free-energy function $\Phi(\theta)$ and the entropy function $H(\eta)$. This duality is shown using Fenchel's convex-conjugate transform [77,78,180,181] and is known to physicists in the form of *Gibbs variational principle* [90,116,195].

The *convex-conjugate* of a function $f(x)$ is the function f^* defined by

$$f^*(y) \triangleq \max_x \{x^T y - f(x)\}. \tag{2.17}$$

The function f^* is a convex function, as it is the maximum over a set of linear functions of y. From this definition, we have *Fenchel's inequality*: $f(x) + f^*(y) - x^T y \geq 0$ for all x, y, Fenchel duality, $f^{**} = f$, holds if and only if f is convex and closed[8]. For

moments η correspond to a boundary point of $\mathcal{M}$ then the value of the maximum entropy problem is well-defined, but there is no exponential family distribution that achieves the maximum. In the latter case there exists a sequence $P^{(k)} \in \mathcal{F}$ with moments $\eta^{(k)} \to \eta$ and where $H(P^{(k)}) \to H(\eta)$.

[8] A function f is *closed* if the sub-level set $\{x | f(x) \leq h\}$ is closed for all h.

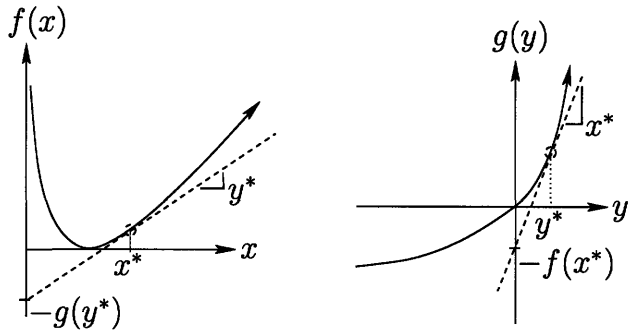

Figure 2.4. Illustration of the Fenchel-Legendre transform for a pair of convex functions: $f(x) = (x-1) - \log x$ $(x > 0)$ and $g(y) = -\log(1-y)$ $(y < 1)$. These are convex-conjugates, that is, $g(y) = f^*(y)$ and $f(x) = g^*(x)$. It holds that $f(x) + g(y) = xy$ for all (x, y) such that $y = \frac{df(x)}{dx}$ (or, equivalently, $x = \frac{dg(y)}{dy}$).

a convex and differentiable function f, the convex-conjugate is given by the *Legendre transform* [17,18]:

$$f^*(y) = \left[x^T y - f(x)\right]_{x=\nabla^{-1}f(y)} \tag{2.18}$$

where $\nabla^{-1}f$ denotes the *inverse gradient map* of f. That is, for a strictly convex, differentiable function $f(x)$ one may define $\nabla^{-1}f(y)$ to denote the unique value of x such that $y = \nabla f(x)$. An example of a pair of single-parameter convex-conjugate functions is shown in Figure 2.4.

Applying this transform to the convex, differentiable function $\Phi(\theta)$, and recalling the moment generating property $\nabla\Phi(\theta) = \mathbb{E}_\theta[\phi] = \Lambda(\theta)$, we see that the convex-conjugate function Φ^* is defined over the set of moments $\eta \in \mathcal{M}$ of the exponential family.[9] For $\eta \in \mathcal{M}$ and $\theta = \nabla^{-1}\Phi(\eta) = \Lambda^{-1}(\eta)$, the entropy is given by $H(\eta) = -P_\theta\{\theta^T\phi(x) - \Phi(\theta)\} = \theta^T\eta - \Phi(\eta) = -\Phi^*(\eta)$. This shows the following duality principle:

Proposition 2.3.1 (Duality between $\Phi(\theta)$ and $H(\eta)$). *The functions $\Phi(\theta)$ and $-H(\eta)$ are convex-conjugate functions. Thus, we have the duality relations:*

$$\Phi(\theta) = \max_{\eta\in\mathcal{M}}\{H(\eta) + \eta^T\theta\} \tag{2.19}$$

and

$$H(\eta) = \min_{\theta\in\Theta}\{\Phi(\theta) - \eta^T\theta\} \tag{2.20}$$

The maximum in (2.19) is uniquely obtained by $\eta = \Lambda(\theta)$. Likewise, the minimum in (2.20) is uniquely obtained by $\theta = \Lambda^{-1}(\eta)$. Also, we have (by Fenchel's inequality)

$$d(\eta, \theta) \triangleq \Phi(\theta) - H(\eta) - \eta^T\theta \geq 0 \tag{2.21}$$

[9] If $\eta \notin \mathcal{M}$, the Fenchel transform is unbounded below and we define $\Phi^*(\eta) = -\infty$.

for all η, θ, where equality holds if and only if $\eta = \Lambda(\theta)$.

In fact, this duality principle is essentially equivalent to *Gibbs variational principle* in statistical mechanics [90,116,195]. Given a potential function $f(x)$ and a probability distribution $P(x)$, we define *Gibbs free energy* (also known as the *variational free energy*) as a function of P (for a fixed energy function f) as the expected value of the energy, with respect to P, plus the entropy of P scaled by temperature:

$$\mathcal{F}_{\text{Gibbs}}(P, \beta) = \mathbb{E}_P[f] + \beta^{-1} H(P). \tag{2.22}$$

This is a concave function of P and is bounded above by the free-energy $\mathcal{F}(f, \beta)$, the log-partition function scaled by temperature:

$$\mathcal{F}_{\text{Gibbs}}(P, \beta) \leq \mathcal{F}(f, \beta) \triangleq \beta^{-1} \log \sum_x \exp\{\beta f(x)\}, \tag{2.23}$$

Moreover, the upper-bound is uniquely achieved by the Gibbs distribution based on $f(x)$. Hence, Gibbs distribution arises as the solution of the variational problem of maximizing Gibbs free energy for a given energy function. Rather than viewing Gibbs free energy as a function of P, we may restrict it to $P \in \mathcal{F}$ and rewrite it as a function of the moments $\eta = \mathbb{E}_P[\phi]$, such that $\mathbb{E}_P[f] = \theta^T \eta$, and the entropy $H(\eta)$:

$$\mathcal{F}_{\text{Gibbs}}(\eta, \beta) = \theta^T \eta + \beta^{-1} H(\eta) \tag{2.24}$$

Note that $\beta \mathcal{F}_{\text{Gibbs}}(\eta, \beta) = (\beta\theta)^T \eta + H(\eta)$ is essentially the same quantity as appears in (2.19) except that θ is scaled by β. Hence, by convex-duality, the maximum of $\mathcal{F}_{\text{Gibbs}}(\eta, \beta)$ (over $\eta \in \mathcal{M}$) is given by the free energy $\mathcal{F}(\theta, \beta) \triangleq \beta^{-1} \Phi(\beta\theta)$. Such variational principles have come to play an important role in recent work on the development of *approximate* inference methods for graphical models [213,227]. This generally involves introducing some tractable approximation to the set $\mathcal{M}$ and the entropy $H(\eta)$. We review one such connection further in Section 2.4.2.

■ 2.3.3 Information Geometry

We give a brief tour of some interesting results of information geometry, which provides a geometric view of the space of probability distributions based on relative entropy and the Fisher information metric [5, 53, 62, 74, 81, 166].

Information Divergence

The *relative entropy* [59] (also known as *Kullback-Leibler divergence* [142,143]) between two probability distributions P and Q is defined as

$$D(P, Q) = \mathbb{E}_P \left[\log \frac{P(x)}{Q(x)} \right]. \tag{2.25}$$

This is commonly used as a measure of contrast between probability distributions, and plays a fundamental role in coding theory, hypothesis testing and large-deviations

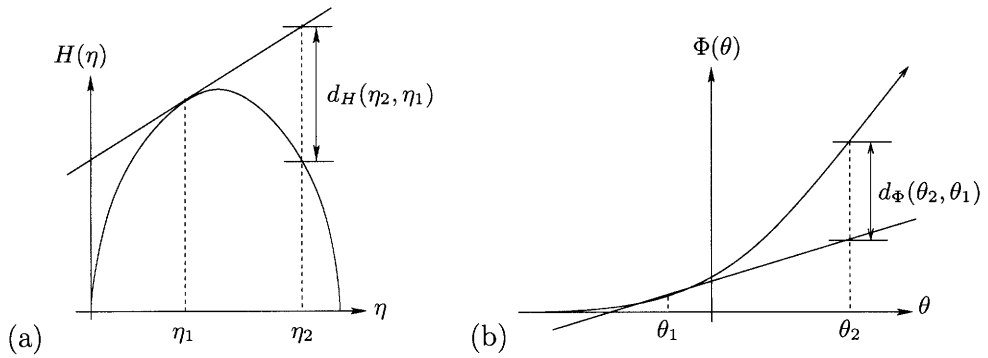

Figure 2.5. Illustration of dual interpretations of Kullback-Leibler divergence as Bregman distance based on either $H(\eta)$ or $\Phi(\theta)$. The example shown corresponds to a single-variable Boltzmann model (a Bernoulli random variable). (a) Bregman distance $d_H(\eta_2; \eta_1)$ in moment coordinates η based on entropy $H(\eta) = -(\eta \log \eta + (1-\eta)\log(1-\eta))$ (defined for $0 \leq \eta \leq 1$. (b) Bregman distance $d_\Phi(\theta_2; \theta_1)$ based on the log-partition function $\Phi(\theta) = \log(1 + e^\theta)$.

theory [59]. Consider the relative entropy $D(P_\eta, P_\theta)$ between two exponential family distributions with (respectively) moments η and parameters θ. It is easily verified that

$$\begin{aligned} D(P_\eta, P_\theta) &= -H(P_\eta) + \mathbb{E}_\eta[-\log P_\theta(x)] \\ &= -H(\eta) + \mathbb{E}_\eta[\Phi(\theta) - \theta^T \phi(x)] \\ &= -H(\eta) + \Phi(\theta) - \theta^T \eta \end{aligned} \tag{2.26}$$

Note, the last expression is equivalent to the $d(\eta, \theta)$ from (2.21). Thus, the information inequality $D(P, Q) \geq 0$ [59] follows from Fenchel's inequality. From this relation, we obtain the following formulas for computing derivatives of relative entropy:

$$\begin{aligned} \frac{\partial d(\eta, \theta)}{\partial \theta_\alpha} &= \Lambda_\alpha(\theta) - \eta_\alpha \\ \frac{\partial d(\eta, \theta)}{\partial \eta_\alpha} &= \Lambda_\alpha^{-1}(\eta) - \theta_\alpha \end{aligned} \tag{2.27}$$

Relative entropy may also be interpreted as the *Bregman distance* [17, 39] based on a convex (or concave) function. To show this, we first express relative entropy as a function of the η parameters in both arguments. Let $d_H(\eta; \eta') \triangleq d(\eta, \Lambda^{-1}(\eta')) = d(\eta, \theta')$ (the reason for the subscript H will be explained). Using the Legendre transform $\Phi(\theta') = H(\eta') + \eta'^T \theta'$ we obtain:

$$\begin{aligned} d_H(\eta, \eta') &= -H(\eta) + \Phi(\theta') - \eta^T \theta' \\ &= -H(\eta) + H(\eta') + \theta'^T(\eta' - \eta) \\ &= -H(\eta) + \{H(\eta') + \nabla H(\eta')^T(\eta - \eta')\} \end{aligned} \tag{2.28}$$

The final expression for $d_H(\eta, \eta')$ shows that it is equal to the amount by which the approximation $\hat{H}(\eta) \triangleq H(\eta') + \nabla H(\eta')^T(\eta - \eta') \approx H(\eta)$, the first-order Taylor-series

expansion of $H(\eta)$ about η', over-estimates $H(\eta)$. This quantity is non-negative because $H(\eta)$ is a concave function (see Figure 2.5(a)). Thus, $d_H(\eta, \eta')$ is a measure of how far η is from η' relative to the the curvature of the function $H(\eta)$. This defines the Bregman distance of a concave function.

In a similar manner, we may express the relative entropy as a function of θ parameters in both distributions, and find that this is equal to the Bregman distance based of the convex function $\Phi(\theta)$ (see Figure 2.5(b)):

$$\begin{aligned} d_\Phi(\theta, \theta') &\triangleq D(P_{\theta'}, P_\theta) \\ &= \Phi(\theta) - \{\Phi(\theta') + \nabla\Phi(\theta')^T(\theta - \theta')\} \end{aligned} \tag{2.29}$$

Thus, Kullback-Leibler divergence can be thought of as the Bregman distance based on either $H(\eta)$ or $\Phi(\theta)$. These two interpretations are consistent due to the duality of $H(\eta)$ and $\Phi(\theta)$. For two distributions P_1 and P_2 it holds:

$$D(P_1, P_2) = d(\eta_1, \theta_2) = d_H(\eta_1, \eta_2) = d_\Phi(\theta_2, \theta_1). \tag{2.30}$$

Note that the order of the arguments is reversed between d_H and d_Φ.

Information Projections

Given an exponential family $\mathcal{F}$, there are two types of *information projections* one may define based on minimizing relative entropy. Each projection defines a different notion of a flat submanifold of the exponential family. Let $S \subset \mathbb{R}^d$ be an affine subspace, that is, $S = \{x \in R^d | Ax = b\}$ for some $A \in \mathbb{R}^{d \times d}$ and $b \in \mathbb{R}^d$. An *e-flat submanifold* of $\mathcal{F}$ is a non-empty subset $\mathcal{F}' \subset F$ with parameters $\theta \in \Theta' = S \cap \Theta$. Analogously, an *m-flat submanifold* is defined by a non-empty, affine subspace in the moment parameterization. The *m-projection* of a probability distribution $P_\eta \in \mathcal{F}$ to an e-flat submanifold S is defined by the solution to the minimum relative entropy problem:

$$\min_{\theta \in \Theta'} d(\eta, \theta) = \min_{\theta \in \Theta'} \{\Phi(\theta) - \eta^T\theta\} - H(\eta) \tag{2.31}$$

We illustrate the m-projection problem in Figure 2.6. Similarly, the *e-projection* of P_θ to an m-flat submanifold S' is defined by:

$$\min_{\eta \in \mathcal{M}'} d(\eta, \theta) = -\max_{\eta \in \mathcal{M}'} \{H(\eta) + \theta^T\eta\} + \Phi(\theta) \tag{2.32}$$

Note that both problems correspond to convex optimization problems minimizing convex functions over affine sets. These two optimization problems are quite similar to those defining the convex-conjugate relation between $\Phi(\theta)$ and $-H(\eta)$. In fact, there is a duality principle relating these two types of information projections. Let us say that an e-flat submanifold Θ' is *orthogonal* to an m-flat submanifold $\mathcal{M}'$ if it holds that $(\eta_2 - \eta_1)^T(\theta_2 - \theta_1) = 0$ for all $\theta_1, \theta_2 \in \Theta'$ and $\eta_1, \eta_2 \in \mathcal{M}'$. This simply means that the coordinate representations, respectively in θ and η coordinates, lie in orthogonal affine subspaces of $\mathbb{R}^d$.

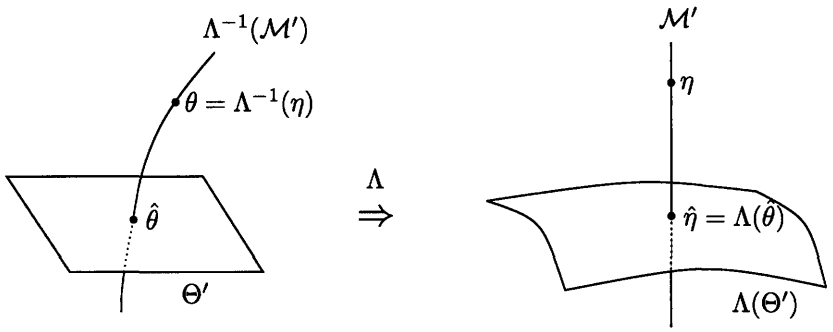

Figure 2.6. Illustration of the m-projection problem. The two figures (on the left and right) are respectively drawn in θ and η coordinates with their respective coordinate-axes $(\theta_1, \ldots, \theta_d)$ and $(\eta_1, \ldots, \eta_d)$ being aligned. Note that Θ' is flat in the θ-space, $\mathcal{M}'$ is flat in η-space and these are orthogonal subspaces. In this example, $\mathcal{M}'$ is also a *one-dimensional* m-flat manifold, called an m-geodesic. The m-projection of η to the e-flat submanifold Θ' is determined by the condition that $\hat{\eta} - \eta$ is orthogonal to Θ', where $\hat{\theta} \in \Theta'$ and $\hat{\eta} = \Lambda(\hat{\theta})$. This can be obtained by following the orthogonal m-geodesic $\mathcal{M}'$ until it intersects the image of Θ' in moment coordinates. Alternatively, it can be found by varying $\hat{\theta}$ over Θ' until the condition $\hat{\eta} \in \mathcal{M}'$ is satisfied.

Proposition 2.3.2 (Duality of e-projection and m-projection). *Let $\Theta' \perp \mathcal{M}'$ be orthogonal submanifolds. Suppose that $\hat{\eta} = \Lambda(\hat{\theta})$ for $\hat{\eta} \in \mathcal{M}$ and $\hat{\theta} \in \Theta$. Then, the following conditions are equivalent:*

- *Intersection: $\hat{\eta} \in \mathcal{M}'$ and $\hat{\theta} \in \Theta'$. In other words, the corresponding probability distribution $\hat{P}$ lies in the intersection of these two manifolds:*

$$\hat{P} \in \{P_\eta, \eta \in \mathcal{M}'\} \cap \{P_\theta, \theta \in \Theta'\}.$$

- *M-Projection: $d(\eta, \theta^*) \leq d(\eta, \theta)$ for all $\eta \in \mathcal{M}'$ and $\theta \in \Theta'$. In other words, $\hat{\theta}$ is the solution of the m-projection problem (2.31).*
- *E-Projection: $d(\hat{\eta}, \theta) \leq d(\eta, \theta)$ for all $\eta \in \mathcal{M}'$ and $\theta \in \Theta'$. In other words, $\hat{\eta}$ is the solution of the e-projection problem (2.32).*
- *Pythagoras Relation: $d(\eta, \theta) = d(\eta, \theta^*) + d(\eta^*, \theta)$ for all $\eta \in \mathcal{M}'$ and $\theta \in \Theta'$.*

Using this duality principle, we may reformulate one type of information projection as a dual information projection. For example, the m-projection of P_η to an e-flat submanifold Θ' is equivalently specified as the e-projection of any element $\theta \in \Theta'$ to an m-flat submanifold $\mathcal{M}'$ containing η which is orthogonal to Θ'. This idea plays an important role in the information geometric interpretation of the iterative scaling algorithm, discussed later in Section 2.7. Note that, following the terminology of Amari [3], the projection to an e-flat manifold is called an m-projection. This convention may seem a bit backwards at first. However, it may be justified by the following observation. Let $\mathcal{M}'$ denote the straight line through η which is orthogonal to Θ' (as shown in Figure 2.6). This is the minimal m-flat submanifold that satisfies the conditions of

the preceding proposition. Such one-dimensional submanifolds are called *m-geodesics*. Then, according to Proposition 2.3.2, the m-projection is determined by the point at which this m-geodesic intersects the e-flat submanifold. Thus, we may obtain the m-projection by following a straight line through η that is orthogonal to Θ' until we reach the point η^* on this line that intersects $\Lambda(\Theta')$.

Fisher Information

For a parametric family of probability distributions $\mathcal{F}$, with parameters $\xi \in \Xi$, the *Fisher information matrix* is defined by:

$$G(\xi) = \mathbb{E}_\xi \left[\nabla_\xi \log P(x;\xi) \nabla_\xi^T \log P(x;\xi) \right] \tag{2.33}$$

This quantity plays an essential role in estimation theory and statistics [59] as well as in exponential families and information geometry [5]. In the exponential family, the Fisher information with respect to the θ parameters is given by the covariance matrix of the feature vector ϕ:

$$G(\theta) = \mathbb{E}_\theta \left[(\phi(x) - \Lambda(\theta))(\phi(x) - \Lambda(\theta))^T \right] \tag{2.34}$$

This same covariance appears as the Hessian matrix $\nabla^2\Phi(\theta)$ of the cumulant-generating function $\Phi(\theta)$ or, equivalently, the Jacobian matrix $\partial\Lambda(\theta)$ of the change of variables from θ to $\eta = \Lambda(\theta) = \nabla\Phi(\theta)$:

$$G(\theta) = \left(\frac{\partial^2 \Phi(\theta)}{\partial\theta_\alpha \partial\theta_\beta} \right) = \left(\frac{\partial \Lambda_\alpha(\theta)}{\partial\theta_\beta} \right) \tag{2.35}$$

Let $G^*(\eta)$ denote the Fisher information with respect to the η parameters. (Here, we add a star to avoid confusion with $G(\theta)$. Elsewhere we may drop the star if it is unambiguous.) Using the chain rule, we can relate this to the corresponding Fisher information $G(\theta)$ computed for $\theta = \Lambda^{-1}(\eta)$:

$$\begin{aligned} G^*(\eta) &= \frac{\partial\theta}{\partial\eta} G(\theta) \frac{\partial\theta}{\partial\eta} \\ &= \left(\frac{\partial\eta}{\partial\theta} \right)^{-1} G(\theta) \left(\frac{\partial\eta}{\partial\theta} \right)^{-1} \\ &= G(\theta)^{-1} G(\theta) G(\theta)^{-1} \\ &= G(\theta)^{-1} \end{aligned} \tag{2.36}$$

Thus, the Fisher information of the two parameterizations are inversely related to one another. Note that $G^*(\eta) = G(\theta)^{-1}$ also equals the Jacobian $\partial\Lambda^{-1}(\eta)$ of the inverse mapping $\Lambda^{-1}(\eta)$. Furthermore, because $\Phi(\theta)$ and $-H(\eta)$ are convex-conjugate functions, the Hessian of $-H(\eta)$ is equal to the inverse of its conjugate. Hence,

$$G^*(\eta) = -\left(\frac{\partial^2 H(\eta)}{\partial\eta_\alpha \partial\eta_\beta} \right)$$

Recalling the expression $d(\eta, \theta)$ for relative entropy, we see Fisher information also appears as the Hessian of relative entropy in either argument:

$$\begin{aligned} G(\theta) &= \left(\frac{\partial^2 d(\eta, \theta)}{\partial \theta_\alpha \partial \theta_\beta} \right) \\ G^*(\eta) &= \left(\frac{\partial^2 d(\eta, \theta)}{\partial \eta_\alpha \partial \eta_\beta} \right) \end{aligned} \tag{2.37}$$

Thus, we can already see that Fisher information inevitably plays an important role in variational approaches to inference and learning.

In fact, Fisher information may be used to define a Riemannian metric for statistical manifold of probability distributions [5]. Although a complete explanation of this perspective is beyond the scope of this background review, we can partially explain the significance of this statement. Given the Fisher information matrix $G(\xi)$ at a point ξ of the parametric model, one can define an inner-product operator at ξ with respect to small perturbations (differentials) $\Delta\xi$ away from ξ by

$$\langle \Delta\xi_1, \Delta\xi_2 \rangle_\xi \triangleq \Delta\xi_1^T G(\xi) \Delta\xi_2 \tag{2.38}$$

An important feature of this metric is that it is *invariant* to smooth reparameterizations of the family. That is, given a second parameterization ξ', related to the first by a smooth bijective map $\rho : \xi \to \xi'$, it holds that

$$G'(\rho(\xi)) = \left(\frac{\partial \rho(\xi)}{\partial \xi} \right)^{-T} G(\xi) \left(\frac{\partial \rho(\xi)}{\partial \xi} \right)^{-1} \tag{2.39}$$

where $G'(\xi')$ denotes the Fisher information in the ξ' parameterization. This implies that $\langle \Delta\xi_1, \Delta\xi_2 \rangle_\xi = \langle \Delta\xi_1', \Delta\xi_2' \rangle_{\xi'}$ for appropriately transformed differentials $\Delta\xi' = \frac{\partial \rho}{\partial \xi} \Delta\xi$. Thus, the condition $\langle \Delta_1, \Delta_2 \rangle = 0$ defines a notion of orthogonality in the space of probability distributions that is independent of the specific choice of parameterization. In the case of exponential families, with dual parameterizations θ and η, this notion of orthogonality is consistent with the one given earlier between e-flat and m-flat submanifolds. This follows from the fact that $G(\eta) = G(\theta)^{-1}$ and $G(\theta)$ is also the Jacobian matrix that describes the change of variables from θ to η, that is, $G(\theta) = \partial\Lambda(\theta)$ and $G(\eta) = \partial\Lambda^{-1}(\eta)$. Then, it is easily verified that

$$\langle \Delta\theta_1, \Delta\theta_2 \rangle_\theta = \langle \Delta\eta_1, \Delta\eta_2 \rangle_\eta = \Delta\eta_1^T \Delta\theta_2 \tag{2.40}$$

where $\Delta\eta_k = G(\theta)\Delta\theta_k$ $(k = 1, 2)$. The third representation, as the inner-product between dual perturbations $\Delta\eta$ and $\Delta\theta$, shows that an e-flat submanifold is orthogonal to an m-flat submanifold (with respect to the Fisher information metric) if an only if their respective coordinate representations in Θ' and $\mathcal{M}'$ lie in orthogonal subspaces.

■ 2.4 Inference Algorithms for Graphical Models

In this section we review a number of *inference algorithms*, the aim of which is to calculate (at least approximately) the marginal distributions of a graphical model. That is, given the potential representation of the graphical model, we wish to calculate the marginal distributions

$$P(x_i) = \frac{1}{Z} \sum_{x_{V \setminus i}} \prod_{E \in \mathcal{G}} \psi_E(x_E) = \frac{1}{Z} \sum_{x_{V \setminus i}} \exp \left\{ \sum_{E \in \mathcal{G}} f_E(x_E) \right\} \tag{2.41}$$

for each $i \in V$ where the sum is over all other variables except for x_i and the partition function Z is a normalization constant. We may also wish to calculate marginal distributions for each edge $E \in \mathcal{G}$. The problem of computing marginals is essentially equivalent to that of computing the partition function itself:

$$Z = \sum_{x} \prod_{E \in \mathcal{G}} \psi_E(x_E) = \sum_{x} \exp \left\{ \sum_{E} f_E(x_E) \right\} \tag{2.42}$$

Direct calculation of this sum becomes intractable in larger models as the number of terms grows as $|\mathbb{X}|^n$ where n is the number of variables. For instance, in binary variable models (e.g., $\mathbb{X} = \{0, 1\}$) direct computations are feasible with present technology only up to about $n = 40$ nodes. For larger graphical models, commonly involving hundreds or thousands of nodes, such brute-force computations are simply not possible.

For models defined on thin (low tree-width) graphs, it becomes possible to compute these sums *recursively* in a nested manner such that the total complexity scales linearly with n (but exponentially with the tree-width of the graph). We review these methods in Section 2.4.1. However, for non-thin graphs, even these recursive methods become intractable. Approximate inference methods, such as the iterative *belief propagation* algorithm, have been developed to provide a tractable approach to inference in graphical models where exact methods are intractable. We discuss belief propagation, its variants and the variational interpretation of such methods in Section 2.4.2.

■ 2.4.1 Recursive Inference Algorithms

We begin by discussing recursive inference in the simplest case of Markov chains and Markov trees and then discuss a general method using junction trees. Although we derive these methods from the perspective of variable elimination (also known as decimation), we also present the resulting message-passing form of these algorithms. In fact, this message-passing approach is the precursor of the iterative belief propagation algorithm, to be discussed later in Section 2.4.2.

Markov Chains

To begin with, let's consider the simple case of a *Markov chain*. This is a graphical model in which the nodes are linearly ordered $V = \{1, \ldots, n\}$ and potential functions

are defined between consecutive nodes. That is, the edges of $\mathcal{G}$ are $\{1,2\},\{2,3\},\ldots,\{n-1,n\}$, as seen in Figure 2.7, and we consider the representation

$$P(x) \propto \prod_{t=1}^{n-1} \psi_{t,t+1}(x_t, x_{t+1}). \tag{2.43}$$

For notational simplicity, we assume that any node factors are absorbed into these pairwise factors. Then, the Markov property implies that, conditioned on x_t, the "past" variables $\{x_v, v < t\}$ are independent of "future" variables $\{x_v, v > t\}$.[10] Now we may evaluate the partition function Z by summing over one variable at at time, in order, a procedure that is known as *variable elimination*. By nesting the sum in this manner, we obtain a tractable calculation:

$$\begin{aligned}
Z &= \sum_{x_n} \cdots \sum_{x_1} \prod_{t=1}^{n-1} \psi_{t,t+1}(x_t, x_{t+1}) \\
&= \sum_{x_n} \sum_{x_{n-1}} \psi(x_{n-1}, x_n) \cdots \sum_{x_2} \psi_{2,3}(x_2, x_3) \sum_{x_1} \psi_{1,2}(x_1, x_2) \\
&= \sum_{x_n} \sum_{x_{n-1}} \psi(x_{n-1}, x_n) \cdots \sum_{x_2} \psi_{2,3}(x_2, x_3) \mu_{1\to 2}(x_2) \\
&= \sum_{x_n} \sum_{x_{n-1}} \psi(x_{n-1}, x_n) \cdots \mu_{2\to 3}(x_3) \\
&\vdots \\
&= \sum_{x_n} \mu_{n-1\to n}(x_n)
\end{aligned} \tag{2.44}$$

Here, each step of variable elimination has the effect of deleting node t and the edge $\{t, t+1\}$ but also creates an addition factor at the next node that we denote by $\mu_{t\to t+1}(x_{t+1})$. This results in a graphical model with one less node that has the same partition function Z and marginal distributions (on the remaining nodes) as the original model. These induced factors $\mu_{t\to t+1}$ may also be regarded as node-to-node "messages" being passed along the chain in a "forward sweep", as illustrated in Figure 2.7(a). Each message is calculated from the preceding message and the corresponding edge potential according to the rule:

$$\mu_{t\to t+1}(x_{t+1}) = \sum_{x_t} \psi_{t,t+1}(x_t, x_{t+1}) \mu_{t-1\to t}(x_t) \tag{2.45}$$

[10] Markov chains are often described using a causal representation defined by pairwise factors $\psi(x_t, x_{t+1}) = P(x_{t+1}|x_t)$ and an initial node factor $\psi_1(x_1) = P(x_1)$. In our discussion, we allow arbitrary factorizations that respect the graph structure (this causal representation is included as a special case).

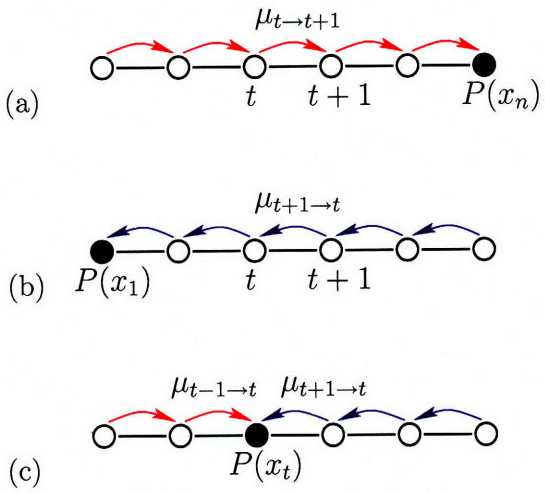

Figure 2.7. Illustration of forward-backward algorithm on a Markov chain. (a) The forward sweep passes messages from left to right. Each node receives a message from its predecessor before sending a message to is successor. (b) The backward pass is identical except that messages are passed from right to left. Note that the two passes do not interact. (c) Combining messages from the forward and backward sweeps, we obtain the marginal distribution at each node. Thus, the two-pass message passing algorithm produces the same results as performing variable elimination separately for each node.

Once the last message is computed, we may also compute the partition function $Z = \sum_{x_n} \mu_{n-1\to n}(x_n)$ and the marginal distribution of the last node, given by

$$P(x_n) = \frac{1}{Z}\mu_{n-1\,\to\, n}(x_n). \tag{2.46}$$

In a similar fashion, we can perform a "backward sweep" on the chain to compute the marginal distribution at the first node, simply by reversing the elimination order and passing messages in the reverse direction down the chain as seen in Figure 2.7(b). For that matter, we can calculate the marginal at *any* intermediate node by passing messages from both ends of the chain towards the desired node as seen in Figure 2.7(c). Importantly, the messages computed in this variable elimination procedure are *identical* to those computed in the forward and backward sweeps. Hence, by a simple two-pass algorithm we obtain all messages (two per edge, one in either direction) necessary to compute the marginal distributions of *all* variables in the chain according to:

$$P(x_t) = \frac{1}{Z}\mu_{t-1\to t}(x_t)\mu_{t+1\to t}(x_t) \tag{2.47}$$

Thus, by exploiting the simple structure of the Markov chain, we reduce the complexity of inference from exponential to linear in n.

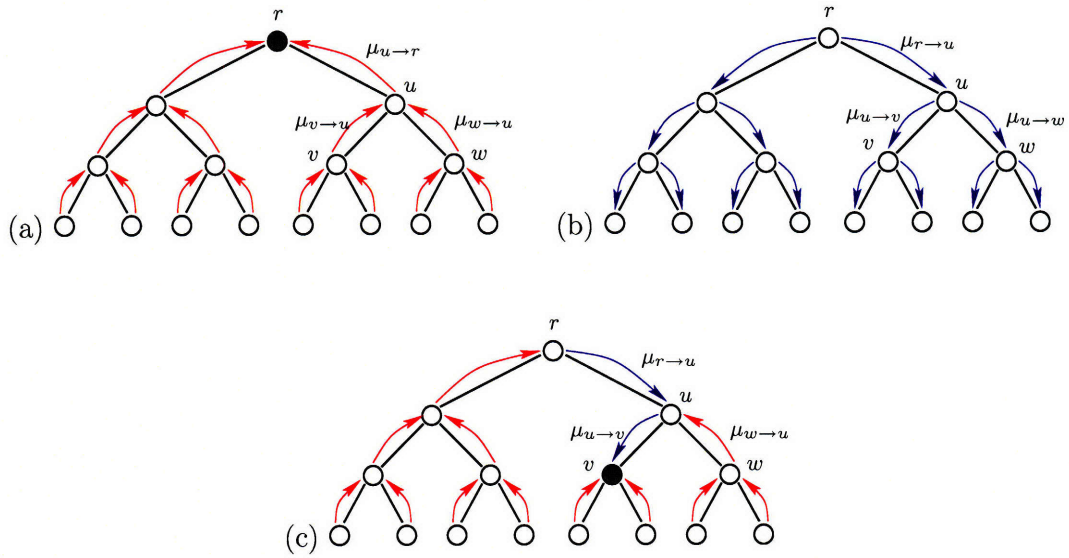

Figure 2.8. Illustration of upward-downward algorithm on a Markov tree. (a) In the upward pass, each node receives messages from all of its children before sending a message to its parent. For instance, $\mu_{u\to r}$ is computed from $\mu_{v\to u}$ and $\mu_{w\to u}$. (b) In the downward pass, each node receives messages from its parent before sending messages to its children. The downward sweep is performed *after* the upward sweep. (c) The set of messages involved in the computation of the marginal distribution at node v. Note, these are the same set of messages that would have been produced if we had selected v to be the root node and performed variable elimination to compute $P(x_v)$. This figure also illustrates that the downward messages depend upon upward messages. For instance, $\mu_{u\to v}$ is computed from $\mu_{w\to u}$ and $\mu_{r\to u}$.

Markov Trees

This idea easily generalizes to the case of *Markov trees*, that is, graphical models with pairwise interaction defined over the edges of a tree. Viewing this as a variable elimination procedure, the algorithm proceeds by eliminating one node at a time. At each step, the next node eliminated must be a leaf in the remaining subtree, that is, a node with only one remaining neighbor. The general rule for eliminating variable x_u, after all but one of its neighbors $v \in \partial u$ have been eliminated, is:

$$\mu_{u\to v}(x_v) = \sum_{x_u} \psi_{uv}(x_u, x_v) \prod_{w\in\partial u\setminus v} \mu_{w\to u}(x_u) \tag{2.48}$$

This may also be envisioned as a two-pass message-passing algorithm on the tree. First, we pick some node $r \in V$ and consider this to be the root node of the tree as seen in Figure 2.8. Then, we perform an "upward" pass on this tree (relative to the root), starting from the leaves of the tree and passing messages in the upwards direction (towards the root node). This upward sweep is depicted in Figure 2.8(a). In this upward sweep, each node waits until it receives messages from all of its children before passing a message up to its parent node. The partition function and marginal distribution at

the root node are computed once the root node is reached. The remaining marginals are determined by performing a reverse downward sweep, as depicted in Figure 2.8(b). In this downward sweep, each node waits until it receives a message from its parent before passing messages to it children. Note, as seen in Figure 2.8(c), the downward message to a given child also takes into account upwards messages from the siblings of that child. Once all messages have been computed, the partition function of the model Z can be computed from any node by fusing all messages to this node (from both the upward and downward sweep) and computing the normalization constant:

$$Z = \sum_{x_u} \prod_{u \in \partial v} \mu_{u \to v}(x_v). \tag{2.49}$$

Note that this must give the same value at each node as it represents the result of variable elimination to compute Z using different elimination orders. The marginal distribution of each node is then given by:

$$P(x_v) = \frac{1}{Z} \prod_{u \in \partial v} \mu_{u \to v}(x_v), \tag{2.50}$$

As depicted in Figure 2.8(c), this is again equivalent to variable elimination. Hence, by computing exactly two messages per edge, we are able to calculate all the messages throughout the tree and obtain both the partition function and marginal distributions with linear complexity in the number of nodes.

Junction Trees and Chordal Graphs

The recursive inference method can be generalized beyond simple Markov chains and trees. The basic idea is to map a Markov model defined on a loopy graph to an equivalent tree-structured model obtained by aggregating together sets of nodes of the loopy graph to define node variables in the tree model. However, complexity of inference in this new tree model then depends on how many nodes we must group together in order to obtain such an equivalent Markov tree representation. To make these ideas precise, we must introduce some additional definitions from graph theory. This discussion is based on inference methods developed in the graphical modeling literature [43, 60, 145, 146], although many results concerning variable elimination and junctions trees were developed earlier in the linear algebra literature [182]. We also note that this general approach is broadly similar to methods developed in the literature on multiscale modeling of 2-D Markov random fields using quad-trees (see the survey paper [222]).

Definitions A graph $\mathcal{G}$ is *chordal* if every cycle of four or more nodes is cut by a *chord* (that is, an edge between two non-consecutive nodes of the cycle). For example, the graph seen in Figure 2.9(a) is *not* chordal because it contains the chordless cycle $(1, 2, 5, 4, 1)$. The graph seen in Figure 2.9(b), however, is chordal. For instance, the cycle $(1, 2, 5, 4, 1)$ is now cut by the chord $(2, 4)$. A *junction tree* of a graph $\mathcal{G}$ is a tree with vertices defined as the set of maximal cliques $\mathcal{C}(\mathcal{G})$ of the graph $\mathcal{G}$ that satisfies the

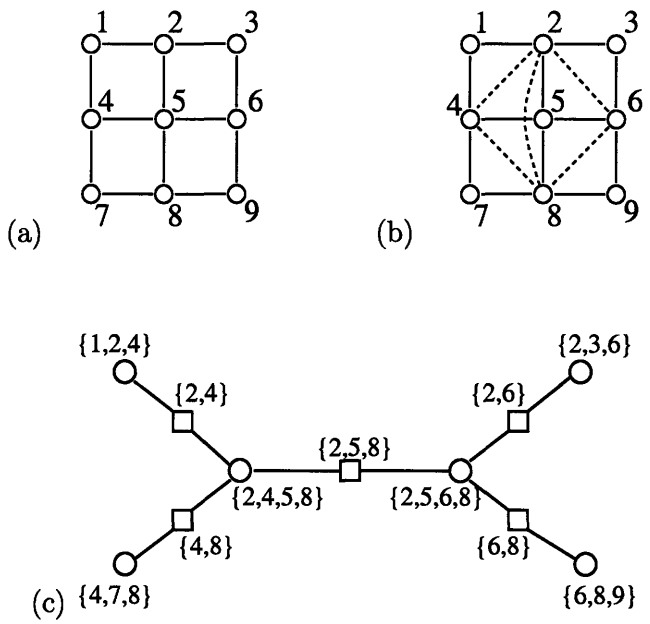

Figure 2.9. Illustration of a junction tree of a graph. (a) A 3×3 grid with vertices $V = \{1, \ldots, 9\}$. This graph is *not* chordal, it contains a chord-free four-cycle $(1, 2, 5, 4, 1)$. (b) A chordal super-graph of this grid. The additional edges are the dashed lines. It can be verified that this graph is chordal by checking that $(1, 3, 7, 9, 4, 6, 2, 8, 5)$ is a perfect elimination order. (c) A junction tree of the chordal graph. The nodes (circles) of this tree represent maximal cliques of the chordal graph. The edges (squares) of the tree represent separators of the graph obtained by taking the intersection of the adjacent cliques. It is verified that this is a junction tree by checking that each pair of nodes satisfies the running intersection property.

following *running-intersection* property: for every pair of cliques $A, B \in \mathcal{C}(\mathcal{G})$ it holds that their intersection $A \cap B$ is included in every other clique C along the path from A to B in the junction tree. We also define the *separators* $\mathcal{S}(\mathcal{G})$ as the intersections of adjacent cliques in the junction tree, one for each edge of the junction tree. For example, a junction tree of the graph seen in Figure 2.9(b) is displayed in Figure 2.9(c). The round markers denote nodes of the junction tree (cliques of $\mathcal{G}$) and the square markers denote edges of the junction tree (separators of $\mathcal{G}$). One can check that the running-intersection property holds. For instance, the intersection of cliques $\{1, 2, 4\}$ and $\{4, 7, 8\}$ is $\{4\}$ and this is included in clique $\{2, 4, 5, 8\}$, which is the only clique along the path from $\{1, 2, 4\}$ to $\{4, 7, 8\}$. A *perfect elimination order* of a graph $\mathcal{G}$ with vertices $V = \{1, \ldots, n\}$ is a permutation π of the vertices such that we may eliminate vertices in the order $(\pi(1), \pi(2), \ldots, \pi(n))$ and at each step t, when we eliminate vertex $\pi(t)$, it holds that the remaining neighbors $\partial\pi(t) \cap V$ of $\pi(t)$ form a clique in the induced subgraph $\mathcal{G}_{\{\pi(t+1),\ldots,\pi(n)\}}$. In other words, every pair of neighbors of $\pi(t)$ at the time that $\pi(t)$ is eliminated are linked by an edge in $\mathcal{G}$. For example, the graph seen in Figure 2.9(b) has the perfect elimination order $\pi = (1, 3, 7, 9, 4, 6, 2, 8, 5)$. These various concepts are closely inter-related, as shown by the following fundamental graph-

theoretic result:

Proposition 2.4.1. *All of the following conditions on $\mathcal{G}$ are equivalent:*

1. *The graph $\mathcal{G}$ is chordal.*
2. *There exists a junction tree based on the maximal cliques of $\mathcal{G}$.*
3. *There exists a perfect elimination order for $\mathcal{G}$.*

The *tree-width* of a chordal graph $\mathcal{G}$ is defined as the size of its largest cliques minus one (so that trees have tree-width one). This definition is extended to non-chordal graphs by defining tree-width as the *minimum* tree-width of any chordal supergraph. Hence, both graphs (a) and (b) in Figure 2.9 have tree-width three. An implication of Proposition 2.4.1 is that a chordal supergraph of a graph $\mathcal{G}$ can be obtained from *any* elimination order. One simply eliminates vertices in the specified order, adding additional *fill* edges at each step, between any two neighbors of the node being eliminated that are not already connected by an edge. This then ensures that in the augmented graph, the specified elimination order is a perfect elimination order and the augmented graph is therefore chordal. Of course, some elimination orders are better than others. Ideally, one might seek to minimize the number of fill edges or the tree-width of the resulting graph. In general, finding such optimal elimination orders in NP-hard [225]. Nonetheless, a number of useful heuristic methods have been developed that work well in practice [6, 21, 31, 183]. In particular, in planar graphs it is tractable to find elimination orders that result in $\mathcal{O}(n^{1/2})$ tree-width graphs [152]. Once an elimination order is determined, it is simple to build a corresponding junction tree (see [43]). An example of this procedure is shown in Figure 2.10, where we illustrate several different chordal super-graphs and corresponding junction trees of an eight-node cycle graph produced by different elimination orders.

The relevance of these graphical concepts to inference can be seen from two perspectives. First, given a junction tree of a chordal super-graph of $\mathcal{G}$ we obtain a Markov tree representation of $P(x)$ by defining variables x_C at each node C of the junction tree. This node variable of the junction tree is identified with the correspond subset of variables $(x_v, v \in C)$ of the graphical model defined on $\mathcal{G}$. Each edge factor $\psi_E(x_E)$ on $\mathcal{G}$ is absorbed into a clique factor $\psi_C(x_C)$ of the junction tree model. Note, however, that each variable x_v of the original model may now be duplicated in multiple nodes on the junction tree. To ensure consistency among these duplicates, we introduce pairwise consistency constraints on each edge of the junction tree, defining pairwise factors $\psi_{A,B}(x_A, x_B)$ for each edge (A, B) of the junction tree to encode the constraint that x_A and x_B should be equal on the subset of variables $x_S = x_{A \cap B}$. Thus, $\psi_{A,B}(x_A, x_B) = 1$ if x_A and x_B are consistent and $\psi_{A,B}(x_A, x_B) = 0$ otherwise. Note that the running-intersection property is essential in order to ensure that these local consistency constraints imply global consistency among all duplicates of each variable in the junction tree. Given this Markov tree representation, we can perform a two-pass message-passing algorithm to compute the partition function of the model and

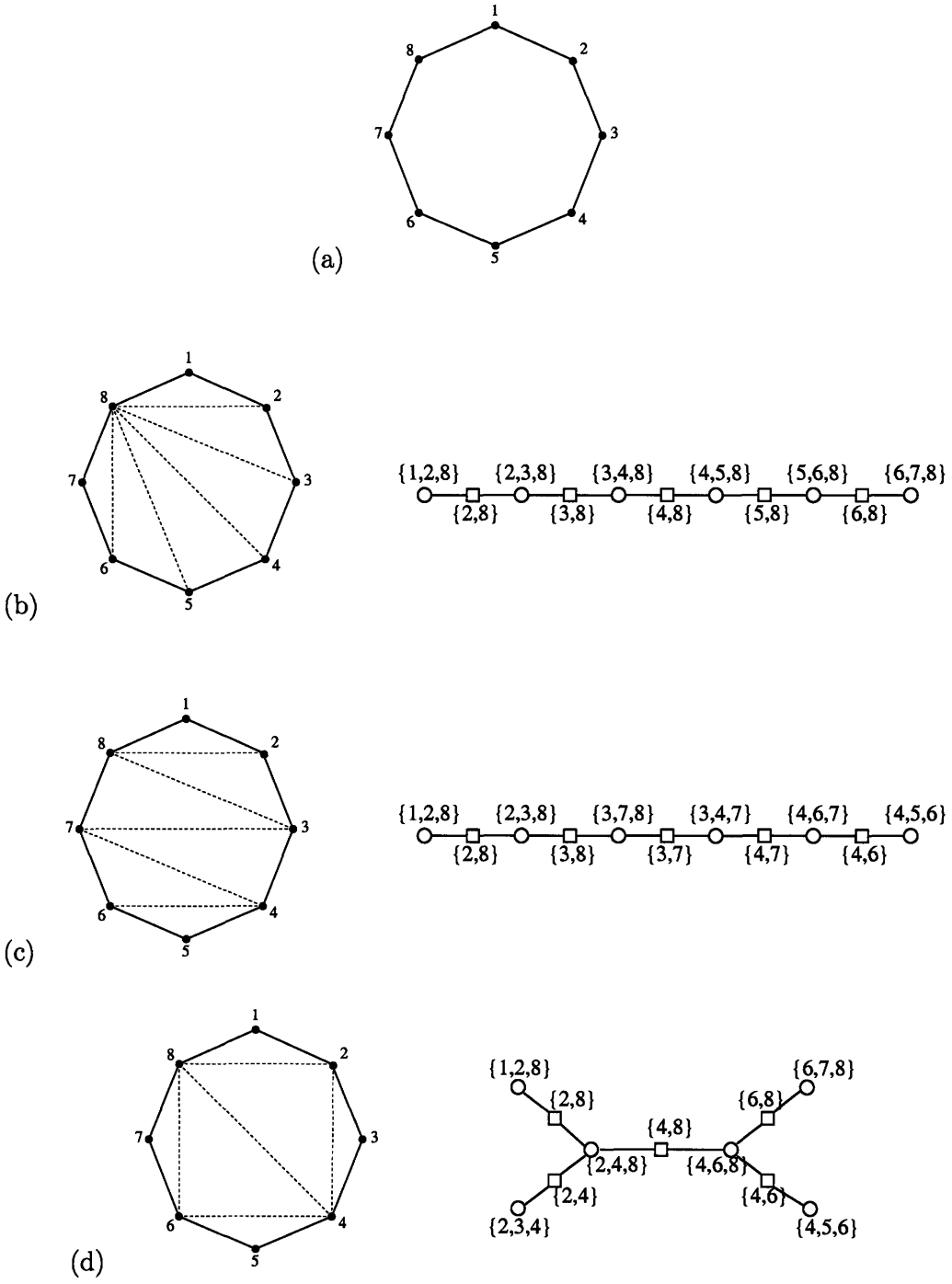

Figure 2.10. Illustration of construction of a junction tree for single-cycle graph with eight nodes. (a) The octagon graph, which is not chordal. In (b), (c) and (d) we show the chordal supergraphs and junction trees resulting from several elimination orders: (b) $(1, 2, 3, 4, 5, 6, 7, 8, 9)$, (c) $(1, 2, 8, 3, 7, 4, 6, 5)$ and (d) $(1, 3, 5, 7, 2, 6, 4, 8)$. The chordal supergraph (with additional fill edges displayed as dashed lines) is shown on the left. The resulting junction tree is shown on the right. In all three cases, the tree-width is two.

the marginal distributions on all cliques of the graph (corresponding to nodes of the junction tree). Messages are passed along the edges of the junction tree in a manner consistent with variable elimination:

$$\begin{aligned}\mu_{A\to B}(x_B) &= \sum_{x_A} \psi_{A,B}(x_A, x_B)\psi_A(x_A) \prod_{C\in\partial A\setminus B} \mu_{C\to A}(x_A) \\ &= \sum_{x_{A\setminus B}} \psi_A(x_A) \prod_{C\in\partial A\setminus B} \mu_{C\to A}(x_A) \end{aligned} \tag{2.51}$$

Here, ∂A denotes the set of *cliques* in the junction tree that are adjacent to clique A. Note that the pairwise consistency constraint, encoded by $\psi_{A,B}$, results in the the sum over x_A being reduced to a sum over $x_{A\setminus B}$. This message passing procedure starts at the leaves of the junction tree and continues until all messages (two for each edge of the junction tree, one in either direction) have been calculated. Once this is done, the partition function can be computed from any node as

$$Z = \sum_{x_C} \psi_C(x_C) \prod_{A\in\partial C} \mu_{A\to C}(x_C) \tag{2.52}$$

and the clique marginals are obtained as:

$$P_C(x_C) = \frac{1}{Z}\psi_C(x_C) \prod_{A\in\partial C} \mu_{A\to C}(x_C). \tag{2.53}$$

The computational complexity of this procedure is bounded by $O(n|\mathbb{X}|^w)$ where w is the tree-width of the graph and n is the number of nodes (the linear dependence on n arises as the number of maximal cliques of a chordal graph is bounded by n). Thus, inference is tractable for the class of *thin* graphs, that is, for graphs where the tree-width is not too large.

Another perspective on inference and tree-width is seen by considering the variable elimination method in a loopy graph. As we have already seen, variable elimination is tractable in trees, because there exists elimination orders that do not result in any fill edges. The effect of variable elimination can then be captured entirely by node-to-node messages. We can also perform variable elimination in loopy graphical models, using an arbitrary elimination order. But variable elimination is then complicated by the fact that, when we eliminate a node that has multiple neighbors (at the time that it is eliminated), this induces a new factor (or message) that couples all of these remaining neighbors so as to faithfully capture the influence of the eliminated node between these neighbors. In this manner, adding fill edges to represent this induced coupling of nodes, the sparsity of the graph is gradually lost such that further variable elimination steps may become intractable. Hence, the apparent sparsity of a graph can conceal the true computational complexity of inference via variable elimination. For this reason, it is natural to consider what class of graphs admit low-fill elimination orders. According to Proposition 2.4.1, this is precisely the class of low tree-width graphs. Thus, the

computational complexity of inference via recursive methods is fundamentally linked to the tree-width of the graph.

As a corollary of Proposition 2.4.1, we also have that any graphical model defined on a chordal graph $\mathcal{G}$ can be factored directly in terms of its marginal distributions defined on the cliques and separators of this chordal graph:

$$P(x) = \frac{\prod_{C \in \mathcal{C}(\mathcal{G})} P_C(x_C)}{\prod_{S \in \mathcal{S}(\mathcal{G})} P_S(x_S)} \tag{2.54}$$

This is called the *junction tree factorization*, as it represents a generalization of the usual factorization of a Markov tree model. We will see that this representation has important consequences both in the context of inference algorithms and for learning. Note that a graphical model defined on a non-chordal graph $\mathcal{G}$ can also be put into this form provided we first add edges to $\mathcal{G}$ so as to create a chordal supergraph. Then, we can express $P(x)$ in terms of marginals on cliques and separators of that chordal super-graph. However, this representation is usually only useful for thin graphs.

While this greatly extends the class of models that can be handled using exact inference methods, there are still many graphical models that arise in practice that fall well outside of this class. For instance, in image processing applications it is common to consider models defined on $w \times w$ lattice with nearest-neighbor connections. The tree-width of this graph is $\mathcal{O}(w)$, resulting in recursive inference methods requiring $\mathcal{O}(|\mathbb{X}|^w)$ computational complexity. Hence, we cannot use these exact methods for even a moderately sized 100×100 images. This motivates the development of approximate inference methods such as discussed in the next section.

■ 2.4.2 Belief Propagation and Variational Methods

In the preceding development we described exact inference on trees from the point of view of variable elimination and message-passing. This suggests a simple heuristic approach to *approximate* inference in graphs with loops, which was first introduced by Pearl [175] and is commonly known as *belief propagation*. Here, we focus on the special case of models defined on *pairwise* graphs with the pairwise factorization

$$P(x) \propto \prod_{\{u,v\} \in \mathcal{G}} \psi_{uv}(x_u, x_v). \tag{2.55}$$

One may also incorporate node potentials $\psi_v(x_v)$ into this discussion, but for notational brevity we assume these have already been absorbed into the pairwise potentials.

The Algorithm The basic idea is to myopically view each node of the graph and its local neighborhood as though it were part of a tree and to apply the tree-based message-passing rules within each such local neighborhood. In graphs with cycles, however, there is no longer a natural order in which messages should be computed. Instead, we *initialize* a complete set of messages (two messages per edge, one in either direction),

e.g., setting these to the uninformative values $\mu^{(0)}_{u\to v}(x_v) = 1$ for all x_v, and then we *iteratively* update these messages according to the equation:

$$\mu^{(t+1)}_{u\to v}(x_v) = \sum_{x_u} \psi_{uv}(x_u, x_v) \prod_{w\in\partial u\setminus v} \mu^{(t)}_{w\to u}(x_u) \tag{2.56}$$

This is also known as the *sum-product* algorithm. We write $\mu^{(t)}_{u\to v}$ to denote the value of the message passed from node u to v, along edge $\{u, v\} \in \mathcal{G}$, at step t of the algorithm. Note, however, that the procedure is memoryless in that only the *last* message from u to v is stored at any given time. We present the version of the algorithm where all messages at step $t+1$ are computed "in parallel" based on the preceding set of messages at step t. Other "serial" methods are also possible, in which messages are updated sequentially, one at a time. The presence of cycles in the graph creates feedback effects due to messages propagating around cycles. Thus, in loopy graphs, the method does not generally reach a fixed point in a finite number of steps. Hence, we now perform belief propagation iteratively in the hope that it is converging to a stable fixed-point of the belief propagation equations, and then terminate the procedure once the differences between consecutive sets of message becomes sufficiently small. If this procedure does converge, then we may estimate the marginal distributions based on these fixed point messages. At step t, this yields the marginal estimates:

$$P^{(t)}(x_v) = \frac{1}{Z^{(t)}_v} \prod_{u\in\partial v} \mu^{(t)}_{u\to v}(x_v) \tag{2.57}$$

where $Z^{(t)}_v$ is a normalization constant. In trees, these belief propagation equations are equivalent to steps of variable elimination and the procedure converges in a finite number of steps, yielding the correct marginal distributions. But in loopy graphs (that is, a graph containing cycles), it may or may not converge, and may yield inaccurate marginals when it does converge. Nonetheless, many examples have been found in non-trivial applications where belief propagation often provides good approximations to the correct marginals distributions.

The Computation Tree To understand how belief propagation works, it is helpful to consider its interpretation as inference on the *computation tree*. The basic idea here is to describe an equivalent tree-structured model such that the marginal distributions computed by belief propagation correspond to marginals of the tree model. To be precise, the marginal distribution $P^{(t)}(x_v)$, of node v at step t of belief propagation, is equivalent to the marginal distribution at the root of the t-step computation tree centered at node v. As seen if Figure 2.11, this tree is formed by exploring all t-step paths of the graph $\mathcal{G}$, starting from node v, and "unrolling" any loops of the graph to build the tree. That is, whenever the next step of a path creates a closed circuit in $\mathcal{G}$, we must add a duplicate node in the computation tree. The structure of this tree then exactly mirrors the structure of the belief propagation algorithm. In particular, provided

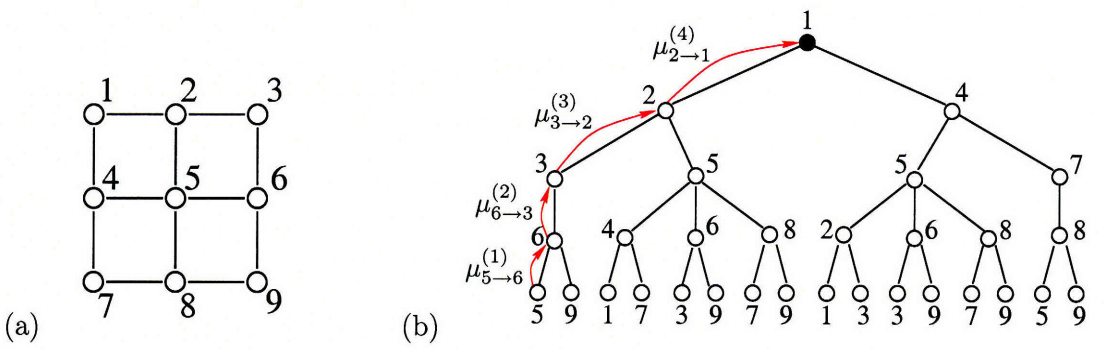

Figure 2.11. Illustration of the computation tree interpretation of belief propagation. (a) The original graph $\mathcal{G}$ on which we perform belief propagation. (b) The four-step computation tree rooted at node 1. The marginal at the root node of the computation tree is equal to the marginal estimate produced by iterative belief propagation after four iterations. Intuitively, iterative belief propagation is equivalent to performing an upwards pass on this tree.

we copy node and edge factors from $\mathcal{G}$ to the corresponding nodes and edges of the computation tree, the marginal distribution at the root of the computation tree is then identical to the estimate computed by belief propagation. Indeed, belief propagation may be thought of as performing the "upward sweep" algorithm on this computation tree. This discussion focuses on the case of the parallel version of belief propagation. However, the idea can be generalized to arbitrary message schedules (see [157]).

It has been shown [202] that convergence of belief propagation is equivalent to the computation tree being well-defined in a certain sense. If the marginal distribution at the root node of the computation tree becomes insensitive to arbitrary boundary conditions as the tree grows, a condition known as *Dobrushin's condition* in the statistical mechanics literature, then the infinite computation tree is well-posed (in the sense that their exists a unique Gibbs measure on the infinite computation tree). While this result is of fundamental theoretical importance, it is difficult to check. One simple sufficient condition, known as *Simon's condition*, is given in [202]. More recent work [163], building upon [113], has developed tighter sufficient conditions based on the spectral radius of a certain matrix, representing the message-passing dynamics of iterative belief propagation on the loopy graph, being less than one.

Variational Interpretation Further insight into belief propagation is provided by its variational interpretation as minimizing the *Bethe free energy* [227, 228]. Let us define a set of *pseudo-marginals* $\eta \in \hat{\mathcal{M}}(\mathcal{G})$ over a pairwise graph $\mathcal{G}$ to be any collection of node and edge marginal distributions, $\eta_i(x_i)$ for $i \in V$ and $\eta_{ij}(x_i, x_j)$ for $\{i, j\} \in \mathcal{G}$, such that each marginal is a valid probability distribution and edge marginals are consistent with

node marginals. That is, $\hat{\mathcal{M}}(\mathcal{G})$ is defined by the set of constraints:

$$\eta_i(x_i) \geq 0 \text{ and } \sum_{x_i} \eta_i(x_i) = 1 \tag{2.58}$$

$$\eta_{ij}(x_i, x_j) \geq 0 \text{ and } \sum_{x_i, x_j} \eta_{ij}(x_i, x_j) = 1 \tag{2.59}$$

$$\sum_{x_i} \eta_{ij}(x_i, x_j) = \eta_j(x_j) \tag{2.60}$$

The set $\hat{\mathcal{M}}(\mathcal{G})$ is also called the *local marginal polytope*. It should be noted that these conditions are *not* in general sufficient to ensure that there exists a distribution $P(x)$ having this collection of marginal distributions. The set of all such *realizable* marginal specifications is called the *marginal polytope* and denoted $\mathcal{M}(\mathcal{G})$. We note that this is precisely the set of realizable moments in the over-parameterized representation of the exponential family of discrete Markov models defined on $\mathcal{G}$. In general, it is intractable to exactly characterize the marginal polytope, as the number of faces of this polytope generally grows exponentially with n. Because any realizable set of marginal distributions must satisfy those consistency constraints defining the pseudo-marginal polytope, we have that $\mathcal{M}(\mathcal{G}) \subset \hat{\mathcal{M}}(\mathcal{G})$. In other words, the pseudo-marginal polytope provides an *outer bound* on the (intractable) marginal polytope.

Now, given any pseudo-marginal specification $\eta \in \hat{\mathcal{M}}(\mathcal{G})$ and potential specification

$$f(x) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j), \tag{2.61}$$

we define the *Bethe free energy* by

$$\mathcal{F}_{\text{Bethe}}(\eta) = \eta^T \theta - H_{\text{Bethe}}(\eta) \tag{2.62}$$

where

$$\eta^T \theta = \sum_i \sum_{x_i} \eta_i(x_i)\theta_i(x_i) + \sum_{ij} \sum_{x_i, x_j} \eta_{ij}(x_i, x_j)\theta_{ij}(x_i, x_j) \tag{2.63}$$

and

$$\begin{aligned} H_{\text{Bethe}}(\eta) &= \sum_{i \in V} (1 - \deg(i)) H(\eta_i) + \sum_{\{i,j\} \in \mathcal{G}} H(\eta_{ij}) \\ &= \sum_{i \in V} H(\eta_i) - \sum_{\{i,j\} \in \mathcal{G}} I(\eta_{ij}) \end{aligned} \tag{2.64}$$

where $H(\eta_i)$ and $H(\eta_{ij})$ denote the marginal entropy of a node or edge and $I(\eta_{ij}) = H(\eta_{ij}) - H(\eta_i) - H(\eta_j)$ is the mutual information between x_i and x_j with distribution η_{ij}. The motivation for the definition of free energy is that, in graphical models defined on *trees*, $H_{\text{Bethe}}(\eta)$ is then equal to the entropy $H(\eta)$ (and the local marginal

polytope $\hat{\mathcal{M}}(\mathcal{G})$ is also equal to the marginal polytope $\mathcal{M}(\mathcal{G})$). Also, for a realizable set of marginals, we note that $\eta^T\theta$ as defined above is then equal to the expected value of the energy $f(x)$ with respect to any $P(x)$ having the marginals specified by η. Thus, in trees, the Bethe free energy is equivalent to Gibbs free energy and, by Gibbs variational principle, exact inference in trees is therefore equivalent to maximizing Bethe free energy. This then motivates using Bethe free energy more generally, effectively disregarding the fact that $\mathcal{G}$ is a loopy graph and taking $H_{\text{Bethe}}(\eta)$ as an approximation to the intractable entropy function $H(\eta)$. This leads to the following variational formulation for approximate inference in loopy graphs:

$$\begin{array}{ll} \text{maximize} & \mathcal{F}_{\text{Bethe}}(\eta) = \eta^T\theta - H_{\text{Bethe}}(\eta) \\ \text{subject to} & \eta \in \hat{\mathcal{M}}(\mathcal{G}) \end{array} \tag{2.65}$$

It should be emphasized, however, that while Gibbs variational problem is a convex problem (maximizing a concave function over a convex set), the approximate entropy H_{Bethe} may no longer be a concave functions of η. Hence, Bethe free energy $\mathcal{F}_{\text{Bethe}}$ is not necessarily concave and so there may exist multiple local maxima.

In the break-through work of Yedidia et al [227, 228], it was shown that this variational formulation of approximate inference is in fact closely related to belief propagation. Essentially, the messages of belief propagation are simply related to a set of *Lagrange multipliers* arising in a dual version of (2.65) obtained by relaxing the constraints that pairwise marginals are consistent with node marginals. Then, the belief propagation fixed-point equations are derived from the Karush-Kuhn-Tucker conditions. In later work [107], this connection was further strengthened to show that *any* stable fixed-point of belief propagation is a local minima of the Bethe free energy. It is not necessarily the case that there exist any such stable fixed points, that these are unique if they do exist nor that belief propagation converges to such a fixed point if there is one. But there are now sufficient conditions for existence and uniqueness of fixed-points, related to convexity of the Bethe free energy and to the corresponding saddle-point problem being convex-concave [108].

Variants of Belief Propagation There are a number a ways in which this basic belief propagation algorithm may be extended. One approach is to allow for higher-order potentials between nodes which leads a version of belief propagation for factor graphs [85]. In the factor-graph form of belief propagation, there are two types of messages: messages for variable nodes to factor nodes and vice versa. Also, a number of methods have been proposed to account for short loops of the graph by considering block versions of belief propagation. This general approach is known as *generalized belief propagation* [228]. It also has a variational interpretation of minimizing the *Kikuchi free energy*, based on an entropy approximation using larger "blocks" of more than two nodes. This entropy approximation is constructed by first summing marginal entropies of a specified set of blocks. Then, to correct for the fact that intersections of these blocks are overcounted, the entropy of subsets of nodes contained in the intersections of larger blocks are added or subtracted as needed in order to obtain an approximation that does not

"over-count" any nodes. This leads to a set of fixed-point equations which aim to minimize the Kikuchi free energy and involves passing messages on the *region graph*. The nodes of the region graph represent blocks in the Kikuchi approximation, and edges are added that link blocks to sub-blocks.

Other methods attempt to address the problem of convergence of belief propagation. Several method have introduced double-loop algorithms that are guaranteed to converge to a local minimum of the Bethe or Kikuchi free energy [109,231]. However, this does not address the problem of multiple local minima or improving the accuracy of marginal estimates. More recently, there have been a variety of approaches which introduce *fractional* [221] or *convex* [212,219] versions of belief propagation. This approach began with the ground-breaking work of Martin Wainwright based upon convex combinations of trees [212]. The main advantage of convex approaches is that they provide a convex free energy and thus ensure a unique solution. This typically (but not always) leads to convergence in the resulting message-passing algorithms, and can sometimes also provide better marginal estimates. However, for models where belief propagation is well-conditioned, it can often provide more accurate marginal estimates than in the "convexified" versions of the algorithm.

■ 2.5 MAP Estimation and Combinatorial Optimization

In this section we consider the problem of finding $x \in \mathbb{X}^n$ to maximize the probability $P(x) \propto \exp f(x) = \exp \sum_{E \in \mathcal{G}} f_E(x_E)$. In applications, problems of this form often arise where we instead seek to maximize the *conditional probability* $P(x|y) \propto P(y|x)P(x)$ given some set of (possibly stochastic) observations $y_k = \zeta_k(x_{E_k})$ on subsets of variables $E_k \subset V$. The problem of maximizing $P(x|y)$ with respect to x is known as *maximum a posteriori* (MAP) estimation. The effect of conditioning on measurements that are conditionally independent given x is to multiply the prior distribution $P(x)$ by additional factors $\prod_k P(y_k|x_{E_k})$ corresponding to likelihood functions of individual measurements. Thus, conditioning on these measurement has the effect of creating additional potentials on edges $E_k \in \mathcal{G}$. However, to simplify notation, we omit explicit reference to these measurements y in the sequel (assuming that the likelihood functions of any measurements have already been incorporated into the energy function), but we still refer to the generic problem of maximizing $f(x)$ as MAP estimation. For discrete models, this is an integer programming problem, which is an *NP-hard* optimization problem.[11] However, there are certain subclasses of this problem which *are* tractable to solve. Moreover, there are a variety of heuristic methods which may solve many instances of the problem, but can also fail to provide any solution in other cases. Finally, it may also be of interest to seek near-optimal solutions in cases for which a provably optimal solution cannot be easily obtained. In this section we review some tractable

[11] This means that, assuming a fundamental hypothesis of complexity theory ($P \neq NP$), there is no algorithm that can solve every instance of the problem for arbitrary n with computational complexity that grows polynomially in n.

model classes and heuristic methods which aim to solve the general problem.

■ 2.5.1 The Viterbi and Max-Product Algorithms

The *Viterbi algorithm* [83, 207], a form of dynamic programming [19], is similar to the recursive inference methods we described earlier for computation of the partition function in Markov chains. Rather than computing Z by summing over all $x \in \mathbb{X}^n$, we instead compute $f^* \triangleq \max f(x)$ by maximizing over all $x \in \mathbb{X}^n$. Again, the ideas of variable elimination (now maximizing over a single variable at a time) and message-passing are essential to efficiently compute the maximum via the recursive calculation:

$$\begin{aligned} f^* &= \max_{x_n} \cdots \max_{x_1} \{f_{1,2}(x_1, x_2) + \cdots + f_{n-1,n}(x_{n-1}, x_n)\} \\ &= \max_{x_n} \left\{ f_{n-1,n}(x_{n-1}, x_n) + \cdots + \max_{x_2} \left\{ f_{2,3}(x_2, x_3) + \max_{x_1} f_{1,2}(x_1, x_2) \right\} \cdots \right\} \\ &= \max_{x_n} \left\{ f_{n-1,n}(x_{n-1}, x_n) + \cdots + \max_{x_2} \{f_{2,3}(x_2, x_3) + \gamma_{1\to 2}(x_2)\} \cdots \right\} \\ &= \max_{x_n} \{f_{n-1,n}(x_{n-1}, x_n) + \cdots + \gamma_{2\to 3}(x_3)\} \\ &\vdots \\ &= \max_{x_n} \gamma_{n-1\to n}(x_n) \end{aligned} \tag{2.66}$$

This defines a message-passing algorithm, with messages $\{\gamma_{u\to v}\}$ computed according to the following rule:

$$\gamma_{t\to t+1}(x_{t+1}) = \max_{x_t} \{f_{t,t+1}(x_t, x_{t+1}) + \gamma_{t-1\to t}(x_t)\} \tag{2.67}$$

At the last elimination step, we obtain $f^* = \max_{x_n} \gamma_{n-1\to n}(x_n)$. We can also define a set of backward messages and use this to compute *max-sum marginals* at each node:

$$\hat{f}_t(x_t) \triangleq \max_{x_{V\setminus t}} f(x) = \gamma_{(t-1)\to t}(x_t) + \gamma_{(t+1)\to t}(x_t) \tag{2.68}$$

If there is a unique MAP estimate $x^* \in \arg\max f$, then it is simple to obtain x^* from the max-sum marginals, it is given by $x_t^* = \arg\max \hat{f}_t$ for all t. Otherwise, we may pick an $x^* \in \arg\max f$ at random as follows. In this case, we also need to compute the following *edge-wise* max-sum marginals:

$$\begin{aligned} \hat{f}_{t,t+1}(x_t, x_{t+1}) &\triangleq \max_{x_{V\setminus\{t,t+1\}}} f(x) \\ &= f_{t,t+1}(x_t, x_{t+1}) + \gamma_{(t-1)\to t}(x_t) + \gamma_{(t+2)\to(t+1)}(x_{t+1}) \end{aligned}$$

We first select $x_1^* \in \arg\max \hat{f}_1$ (at random) and then select the remaining variables sequentially to maximize the edge marginals. That is, given x_t^* we randomly select

x^*_{t+1} subject to the constraint $(x^*_t, x^*_{t+1}) \in \arg\max \hat{f}_{t,t+1}$. Due to the definition of max-marginals, there is always an x^*_{t+1} that satisfies this constraint. The resulting estimate must then maximize $f(x)$.

Analogous to our discussion in Section 2.4.1, these recursive message-passing algorithms extend to Markov trees and to junction trees of loopy graphs. This leads to calculation of max-marginals of each node and edge of the Markov tree or of each clique in the junction tree representation, from which one may solve the MAP estimation problem using a similar method as described above for chains. This also implies that, analogous to the junction tree factorization, the energy function of a graphical model can be expressed in terms of the max-marginals over the cliques and separators arising in the junction tree representation:

$$f(x) = \sum_{C \in \mathcal{C}(\mathcal{G})} \hat{f}_C(x_C) - \sum_{S \in \mathcal{S}(\mathcal{G})} \hat{f}_S(x_S) \tag{2.69}$$

However, junction tree methods are only tractable for the class of *thin* graphs, with computational complexity growing exponentially in the tree-width of the graph. This motivates the development of approximate methods to deal with non-thin graphs.

The Max-Product Algorithm

Analogous to the sum-product form of iterative belief propagation (Section 2.4.2), iterative versions of the Viterbi algorithm for non-thin, loopy graphs have also been developed. This approach is most commonly presented in the form of the *max-product algorithm* [218], but we also describe the *max-sum* form of the algorithm that is similar to the Viterbi algorithm.

In the max-product algorithm, we define messages $\hat{\mu}_{u \to v}$ over the edges of the graph and then iteratively update these messages according to the rule:

$$\hat{\mu}_{u \to v}(x_v) = \frac{1}{Z_{u \to v}} \max_{x_u} \left\{ \psi_{uv}(x_u, x_v) \prod_{w \in \partial u \setminus v} \hat{\mu}_{w \to u}(x_u) \right\} \tag{2.70}$$

where $Z_{u \to v}$ is a normalization constant.[12] Note that this is all but identical to the sum-product form of belief propagation (2.56), except the sum over x_u is now replaced by the maximum over x_u. This procedure is equivalent to one defined in the log-domain, with messages defined by $\gamma_{u \to v}(x_v) \triangleq \log \hat{\mu}_{u \to v}(x_v)$ and with the factors $\psi_{uv}(x_u, x_v)$ replaced by potential functions $f_{uv}(x_u, x_v) = \log \psi_{uv}(x_u, x_v)$. This leads to the *max-sum* message-passing algorithm:

$$\gamma_{u \to v}(x_v) = \max_{x_u} \left\{ f_{u,v}(x_u, x_v) + \sum_{w \in \partial u \setminus v} \gamma_{u \to v}(x_u) \right\} + \log Z_{u \to v} \tag{2.71}$$

[12] In the context of the max-product algorithm, one might also define the normalization constant so that the maximum value of the messages are scaled to one.

In this form of the algorithm, the constant $\log Z_{u \to v}$ may instead by chosen so that $\max \gamma_{u \to v}(x_v) = 0$. Again, these iterative algorithms may or may not converge. If they do converge, then we can obtain approximate max-marginals by combining messages at each node and use these to derive an approximate MAP estimate. In the max-sum form of the algorithm, we sum messages at each node to obtain the max-marginals:[13]

$$\hat{f}_v(x_v) = \sum_{u \in \partial v} \gamma_{u \to v}(x_v) \tag{2.72}$$

An estimate $\hat{x}$ is then obtained by selecting each component to maximize the corresponding max-marginal: $\hat{x}_v = \arg\max \hat{f}_v$ (assuming their is a unique maximum at each node). However, even when this procedure does converge (and yields an unambiguous estimate) this may still not be an optimal MAP estimate. But it has been shown [218] that this estimate is a *local maximum* in the following sense: $f(\hat{x}) \geq f(x)$ for all x that differ from $\hat{x}$ only on a subset of nodes $A \subset V$ such that there are no cycles in the induced subgraph $\mathcal{G}_A$. In other words, the estimate $\hat{x}$ cannot be improved by changing variables on any subset of nodes whose induced subgraph is a tree. In particular, this does imply that if $\mathcal{G}$ is either a tree or contains at most one cycle, then the estimate $\hat{x}$ is optimal (it is actually a MAP estimate).

The Zero-Temperature Limit of Inference

We digress for a moment to consider a connection between the sum-product and max-product forms of belief propagation. Earlier, in Section 2.2.2, we commented that the Gibbs distribution becomes concentrated on the set of MAP estimates as the temperature approaches zero. This suggests that, in the limit of zero-temperature, inference becomes equivalent to MAP estimation. To strengthen this analogy, we show that the sum-product algorithm (reformulated in the log-domain), applied to a Gibbs distribution with variable temperature β^{-1}, reduces to the max-sum algorithm in the limit as the temperature approaches zero.

We begin by showing that calculation of the free energy (the log-partition function scaled by the temperature) reduces to the value of the MAP problem. More precisely, we consider the free energy function:

$$\mathcal{F}_\beta(f) = \beta^{-1} \log \sum_x \exp\{\beta f(x)\} \tag{2.73}$$

This is a smooth, convex function of the potential specification of the model and may be considered as a smooth approximation to the max-function $\mathcal{F}_{\max}(f) \triangleq \max_x f(x)$ [37]. This perspective is justified by the following result:

[13]Note, however, that it is no longer meaningful to interpret these as literally being estimates of the corresponding max-marginals of the energy function $f(x)$ defined on the graph $\mathcal{G}$. Instead, these correspond (up to an additive constant) to max-marginals over the *computation tree* of $\mathcal{G}$. Moreover, because the computation tree is growing, these max-sum marginals are typically divergent. Hence, we instead define convergence in terms of the exponentiated max-marginals $\hat{p}(x_v) = \exp \hat{f}_v(x_v)$ corresponding to max-product marginals.

Proposition 2.5.1. *The function $\mathcal{F}_\beta$ is strictly greater than $\mathcal{F}_{\max}$ for all f and $\beta > 0$. Also, $\mathcal{F}_\beta$ is monotonically decreasing with β and converges uniformly to $\mathcal{F}_{\max}$ as $\beta \to \infty$.*

Proof. We rewrite the free energy as

$$\mathcal{F}_\beta(f) = \mathcal{F}_{\max}(f) + \beta^{-1} \log \sum_x \exp\{\beta(f(x) - \mathcal{F}_{\max}(f))\}. \tag{2.74}$$

The sum is greater than one, because at least one x satisfies $f(x) = \mathcal{F}_{\max}(f)$ and the other terms are positive. Therefore, the log of the sum is greater than zero and $\mathcal{F} > \mathcal{F}_{\max}$. Also, the sum is bounded above by $|\mathbb{X}|^n$ because $f(x) - f^* \leq 0$ for all x. Hence, $\mathcal{F} \leq \mathcal{F}_{\max} + \beta^{-1} n \log |\mathbb{X}|$. As $\beta^{-1} \to 0$ the upper-bound converges to the lower-bound and $\mathcal{F}$ therefore converges to $\mathcal{F}_{\max}$. Moreover, the difference between the upper and lower bound is $\beta^{-1} n \log |\mathbb{X}|$, which is independent of f. Thus, $\mathcal{F}_\beta$ converges *uniformly* to $\mathcal{F}_{\max}$. Monotonicity is shown by noting that $\frac{\partial \mathcal{F}}{\partial \beta}$ is equal to the entropy of the Gibbs distribution (which is positive) scaled by $-\beta^2$. □

Now, we use this result to relate the sum-product and max-product forms of inference. Let us reparameterize the sum-product messages in terms of the log-messages:

$$\gamma^{(\beta)}_{u \to v}(x_v) \triangleq \beta^{-1} \log \mu^{(\beta)}_{u \to v}(x_v) \tag{2.75}$$

where $\mu^{(\beta)}_{u \to v}$ is the sum-product messages computed with respect to the Gibbs distribution:

$$P(x) \propto \exp\left\{ \beta \sum_{\{u,v\} \in \mathcal{G}} f_{uv}(x_u, x_v) \right\} \tag{2.76}$$

Note that the effect of varying the temperature is to scale the potential functions by β, which essentially means that we replace each potential $f_{u,v}$ by $\beta f_{u,v}$ (or, equivalently, replace $\psi_{u,v}$ by $\psi^\beta_{u,v}$) in the sum-product algorithm. In terms of these log-messages, belief propagation has the form:

$$\gamma^{(\beta)}_{u \to v}(x_v) = \beta^{-1} \log \sum_{x_u} \exp\left\{ \beta \left(f_{uv}(x_u, x_v) + \sum_{w \in \partial u \setminus v} \gamma^{(\beta)}_{w \to u}(x_u) \right) \right\} \tag{2.77}$$

Note that this has the same "log-sum-exp" form as in the free energy. Applying Proposition 2.5.1, we have:

$$\lim_{\beta \to \infty} \gamma^{(\beta)}_{u \to v}(x_v) \triangleq \gamma^{(\infty)}_{u \to v}(x_v) = \max_{x_u} \left\{ f_{uv}(x_u, x_v) + \sum_{w \in \partial u \setminus v} \gamma^{(\infty)}_{w \to u}(x_u) \right\} \tag{2.78}$$

Thus, the zero-temperature limit $\beta \to \infty$ of these log-messages are essentially equivalent to the messages computed by the max-sum algorithm (the log-form of max-product). We also define the "soft-max" marginals:

$$\hat{f}^{(\beta)}_v(x_v) = \sum_{u \in \partial v} \gamma^{(\beta)}_{u \to v}(x_v, \beta). \tag{2.79}$$

These soft-max marginals are upper-bounds on max-sum marginals and converge to max-sum marginals at zero temperature. Hence, the max-sum algorithm is essentially equivalent to a zero-temperature version of the sum-product algorithm and there is in fact a smooth family of inference algorithms between sum-product and max-product parameterized by the temperature parameter.

■ 2.5.2 LP Relaxation of MAP Estimation

Next, we consider linear-programming (LP) approaches to MAP estimation. Recall that, by Gibbs variational principle, we may express exact inference at finite temperatures as maximizing a concave function over the marginal polytope.

$$\mathcal{F}_\beta(f) = \begin{cases} \text{maximize} & \theta^T\eta + \beta^{-1}H(\eta) \\ \text{subject to} & \eta \in \mathcal{M}(\mathcal{G}) \end{cases} \tag{2.80}$$

In the previous section, we noted that $\mathcal{F}_\beta(f)$ converges to $f^* = \max f$ in the zero-temperature limit. The weight placed on the entropy term is going to zero in this limit and we are left with the following *exact* formulation of MAP estimation:

$$f^* = \begin{cases} \text{maximize} & \theta^T\eta \\ \text{subject to} & \eta \in \mathcal{M}(\mathcal{G}) \end{cases} \tag{2.81}$$

This is a linear program (LP), optimizing a linear objective function over a polytope. It is equivalent to the integer programming problem, $f^* = \max\{\theta^T\phi(x)|x \in \mathbb{X}^V\}$, because the marginal polytope $\mathcal{M}(\mathcal{G})$ is, by definition, equal to the convex hull of the set $\{\phi(x)|x \in \mathbb{X}^V\}$ and every $x \in \mathbb{X}^V$ corresponds to a vertex of $\mathcal{M}(\mathcal{G})$. Typically there is a unique solution η^*, which is then a vertex of the marginal polytope. Such η^* place all of the probability on a single configuration $x^* \in \mathbb{X}^n$, this being the MAP estimate. Thus, x^* can be derived from the marginals by $x_v^* = \arg\max \eta_v(x_v)$.

Pseudo-Marginal LP Relaxation

However, as mentioned in Section 2.4.2, it is generally intractable to represent the marginal polytope exactly because the number of faces of this polytope generally grows exponentially with the number of variables n. Hence, it is not tractable to solve this exact LP formulation of MAP estimation directly. Instead, we may solve a relaxed version of this LP, where the intractable marginal polytope $\mathcal{M}(\mathcal{G})$ is replaced by the tractable local marginal polytope $\hat{\mathcal{M}}(\mathcal{G})$. This then gives a tractable LP which provides an upper-bound on the value of the MAP problem. However, as illustrated in Figure 2.12, this may or may not lead to an optimal MAP estimate x^*. If the solution η^* of the relaxed LP still corresponds to a vertex of the marginal polytope $\mathcal{M}(\mathcal{G})$, as depicted in Figure 2.12(a), then this is also an optimal solution of the exact LP and we recover the MAP solution. But if the solution η^* corresponds to a vertex of the local polytope $\hat{\mathcal{M}}(\mathcal{G})$ that is *outside* of the marginal polytope $\mathcal{M}(\mathcal{G})$, as depicted in Figure 2.12(b), then η^* must contain some fractional values (between zero and one) and we cannot

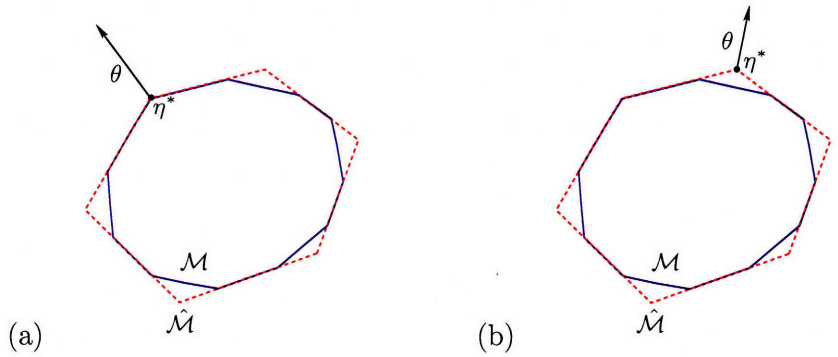

Figure 2.12. Notional illustration of LP relaxation of MAP estimation. The marginal polytope $\mathcal{M}(\mathcal{G})$ (the solid blue polygon) is contained within the pseudo-marginal polytope $\hat{\mathcal{M}}(\mathcal{G})$ (the dashed red polygon). (a) For some choices of θ, the optimal vertex over $\hat{\mathcal{M}}(\mathcal{G})$ is also a vertex of $\mathcal{M}(\mathcal{G})$, corresponding to a probability distribution that assigns all of its probability to a single configuration x^*, this being the optimal MAP estimate. In this case, η^* is *integral* and the optimal x^* is obtained. (b) Other choices of θ lead to an *integrality gap*. Then, the optimal vertex η^* of $\hat{\mathcal{M}}(\mathcal{G})$ is outside of the true marginal polytope $\mathcal{M}(\mathcal{G})$ and η^* must contain some fractional values.

recover the MAP estimate. This is called an *integrality gap*. Optimality can be verified by checking if η^* is integral, or, equivalently, if each node's marginal η_i has a unique maximum. In the case of an integrality gap, at least one node's marginal exhibits "ties", that is, multiple maxima.

Message-Passing Approaches to LP Relaxation

We briefly review a number of message-passing algorithms that may be viewed as dual approaches to solving LP relaxations of MAP estimation.

To begin with, we comment that the max-product algorithm itself may be viewed as attempting to solve this LP, based on the variational interpretation of belief propagation and that max-product is the zero-temperature form of belief propagation. Recent work [187, 188] provides some support for this view, at least for special classes of problems. However, there are some difficulties with this point of view in general. First, max-product is *not* guaranteed to converge. Second, even if max-product does converge, and yields an unambiguous estimate, it is not guaranteed to be optimal. This is seemingly inconsistent with the property of the relaxed LP formulation that integral solutions are optimal. One hypothesis to explain this discrepancy is that failure of max-product to find optimal solutions upon convergence may be linked to non-convexity of the Bethe free energy. This explanation is consistent with the fact that, in trees and graphs with a single cycle, Bethe free energy is convex and max-product does then give optimal solutions in these graphs if it converges.

This motivates considering zero-temperature versions of convex belief propagation such as the tree-reweighted max-product algorithm [211] or low-temperature versions of convex belief propagation [219]. If such a method converges and yields an integral

solution, then it is optimal. However, convergence of these algorithms is not guaranteed and they may yield non-integral solutions if they do converge. There has been closely related work on convergent message-passing algorithms, such as the convergent form of TRMP introduced by Kolmogorov [134, 135], a coordinate-descent version of max-product [93] and earlier work on the max-sum diffusion method [140, 191, 192] (recently reviewed in [220]). All of these approaches may be viewed as coordinate descent methods associated with certain dual functions that provide upper-bounds on the value of the LP relaxation (although, using different formulations and parameterizations, the resulting methods are similar but not quite equivalent). However, because these dual functions are piecewise-linear functions, which are non-differentiable, coordinate-descent can actually fail to minimize these dual function, that is, it may converge to a *non-minimal* fixed point. This is a fundamental problem with the zero-temperature approach, and may also be related to why max-product is not necessarily optimal when it does converge.

We also mention that there have been a number of earlier works relating to linear dual approaches to MAP estimation that are not being reviewed here. In particular, there has been much work on binary quadratic optimization in the optimization research literature [34–36, 103]. Although these methods are expressed in very different forms and are solved using different methods, they often lead to the same fundamental optimization problem. For instance, the methods just cited are equivalent to the local marginal polytope relaxation in the case of binary variable models with pairwise interactions.

■ 2.5.3 Combinatorial Optimization Methods

In this section we briefly discuss some special cases where the MAP estimation problem reduces to a tractable problem in the network optimization literature. These methods are also linear programming approaches, although the emphasis here is on *graphical* formulations that have special-purpose solution techniques.

Max-Cut

We discuss the max-cut formulation of MAP estimation in binary variable models, and review the work on cutting-plane methods for solving this problem [12, 13, 198]. Throughout this section we consider pairwise graphs $\mathcal{G}$.

A *cut* of the graph $\mathcal{G}$ is a subset of its edges defined by $K = \{\{u, v\} \in \mathcal{G} | u \in V_1, v \in V_2\}$ for some bipartition of the vertices $V = V_1 \cup V_2$ (where V_1 and V_2 are disjoint). For example, a cut of the graph seen in Figure 2.13(a) is shown in (b). Given edge weights w_E for all $E \in \mathcal{G}$, which may be positive or negative, the *max-cut* problem is to find the maximum-weight cut of the graph where the weight of a cut is defined $w(K) = \sum_{E \in K} w_E$. In the case of planar graphs, cuts may be equivalently described as follows. Recall that a planar graph is one that may be drawn in the plane without any intersecting edges. This drawing then divides the plane into a set of disjoint regions separated by the edges of the graph. These regions are called the

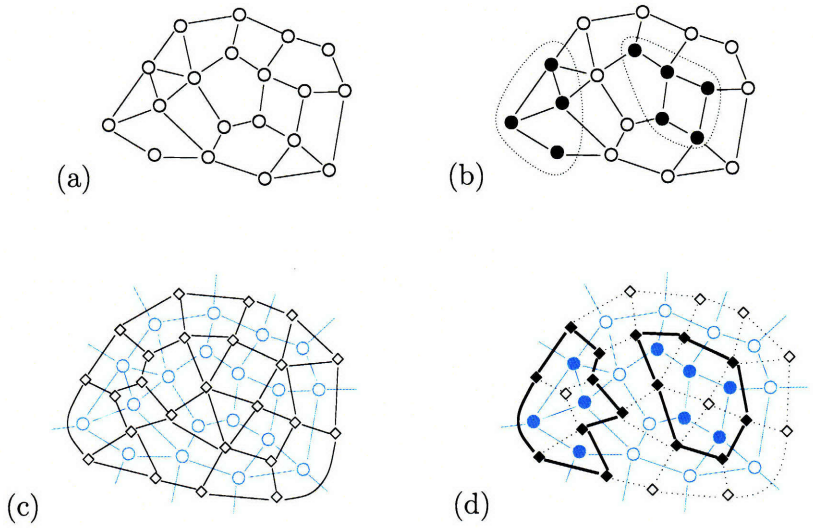

Figure 2.13. Illustration of a cut in a planar graph $\mathcal{G}$ and its dual graph $\mathcal{G}^*$. (a) The planar graph $\mathcal{G}$. (b) A cut shown both as a bi-partition of the node set (white and black nodes) and as a set of cut edges (the ones intersecting the closed dotted contours). (c) The dual graph $\mathcal{G}^*$ in which each node (displayed as a diamond) corresponds to a face of the planar graph and each edge connects adjacent faces. To simplify the diagram of $\mathcal{G}^*$, we have split the outer face into smaller faces by adding fictitious edges in $\mathcal{G}$ connecting each node around the perimeter of the graph to an imaginary point at infinity. (d) The set of cycles in $\mathcal{G}^*$ (shown in bold) corresponding to the cut of $\mathcal{G}$ seen in (b).

faces of the planar graph. The *dual graph* $\mathcal{G}^*$ is the graph whose nodes are identified with the faces of $\mathcal{G}$ and with pairwise edges connecting adjacent faces. An example of a planar graph and its dual graph are seen in Figure 2.13(c). Note that there is a one-to-one correspondence between edges $E \in \mathcal{G}$ and corresponding edges of the dual graph $E^* \in \mathcal{G}^*$. The dual edge E^* is the one that crosses E when we superimpose $\mathcal{G}^*$ on $\mathcal{G}$ as in Figure 2.13(c). There is also a one-to-one correspondence between cuts in the graph $\mathcal{G}$ and *even-degree subgraphs* of $\mathcal{G}^*$, that is, subgraphs of $\mathcal{G}^*$ in which every node has even degree (equivalently, subgraphs formed as a union of edge-disjoint cycles). For example, in Figure 2.13(d) we show the even-degree subgraph corresponding to the cut seen in (b). Thus, in planar graphs, max-cut is equivalent to finding the maximum-weight even-degree subgraph of the dual graph, where the weight of each dual edge $E^* \in \mathcal{G}^*$ is equal to the weight of the edge $E \in \mathcal{G}$ that it cuts.

Consider MAP estimation for the *zero-field* Ising model, which has binary variables $x_v \in \{-1, +1\}$ and energy function:

$$f(x) = \sum_{\{u,v\} \in \mathcal{G}} \theta_{uv} x_u x_v \tag{2.82}$$

The edge parameters θ_{uv} may be positive or negative. In principle, the assumption of zero-field ($\theta_v = 0$ for all $v \in V$) does not result in any loss of generality. As illustrated in Figure 2.14, a general Ising model can be mapped to a zero-field model with one extra

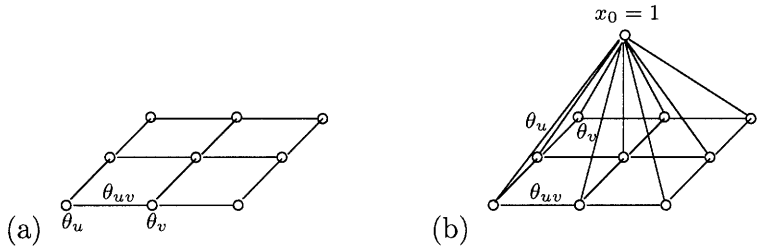

Figure 2.14. Illustration of method to convert a general Ising model into an equivalent zero-field Ising model with one auxiliary node. (a) The non-zero field Ising model on a 3×3 grid with node potentials θ_i and edge potentials θ_{ij}. (b) A zero-field model in which we have added an auxiliary node x_0 which is linked to each of the original variable nodes by edges $\{0, v\}$ for all $v \in \{1, \ldots, n\}$. The weights of these edges are defined $\theta_{0,v} = \theta_v$.

variable such that solving this augmented model is equivalent to solving the general Ising model.[14] Every assignment $x \in \mathbb{X}^V$ determines a cut:

$$K(x) = K(-x) = \{\{u, v\} \in \mathcal{G} | x_u x_v = -1\}. \tag{2.83}$$

The energy $f(x) = f(-x)$ is related to this cut $K(x)$ by:

$$f(x) = \sum_{E \notin K(x)} \theta_E - \sum_{E \in K(x)} \theta_E = \sum_{E \in \mathcal{G}} \theta_E - 2 \sum_{E \in K(x)} \theta_E \tag{2.84}$$

Then, MAP estimation is equivalent to finding the maximum-weight cut, with edge-weights defined $w(E) = -\theta_E$.

The max-cut problem may be formulated as an LP as follows. For each edge $\{u, v\}$ we define an edge variable $y_{u,v} \in \{0, 1\}$ that is equal to one if $E \in K$. Then, we have the continuous LP relaxation of max-cut:

$$\begin{array}{ll} \text{maximize} & w^T y \\ \text{subject to} & y \in \mathcal{K}(\mathcal{G}) \end{array} \tag{2.85}$$

where $y \in \mathbb{R}^{|\mathcal{G}|}$ is a vector of (continuous) edge variables and $\mathcal{K}(\mathcal{G})$ denotes the *cut polytope*, the convex hull of the set of valid cut vectors (i.e., where each element of y is either zero or one). This is an *exact* LP for the max-cut problem. However, it is generally intractable to characterize the cut polytope exactly. One simple relaxation of max-cut is only to constrain $y_E \in [0, 1]$. Tighter approximations to the cut polytope are obtained using the *odd cycle inequalities* [13]. Although it is not tractable to enumerate all of these constraints explicitly (e.g., in the simplex method) it is possible to implement an efficient *cutting plane method* [13]. This is an efficient method to check for violated

[14]The augmented model includes an auxiliary variable x_0 and corresponding node $0 \in V$. We also include extra edges, $\{0, v\} \in \mathcal{G}$ for all $v \notin V \setminus 0$, and corresponding edge potentials $\theta_v x_0 x_v$. Then, the maximum value f^* is the same for both problems, and each pair of MAP estimates, $(1, x^*)$ and $(-1, -x^*)$, of the augmented model corresponds to a MAP estimate x^* of the original model.

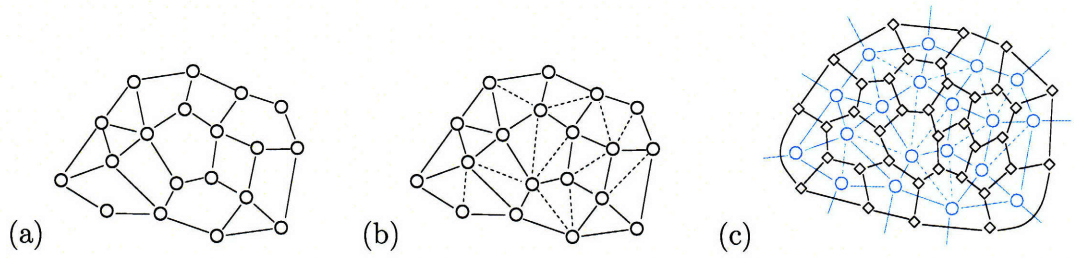

Figure 2.15. Illustration of procedure to obtain a triangular planar graph. (a) A planar graph. (b) We obtain a triangular version of this graph by adding edges to split each internal face into a set of triangles (faces with three sides). The weights on the new edges are set to zero. (c) The dual graph (we also split the outer face into triangles). Observe that all nodes of the dual graph are of degree three.

cycle inequalities, so that one may solve a series of tractable LPs by adding violated inequalities until there are no more violated constraints. In general, this approach is not guaranteed to converge to an integral solution. But, in the class of planar graphs, it has been shown that these odd-cycle inequalities provide a tight representation of the marginal polytope, so that it is tractable to solve max-cut in planar graphs (and, hence, MAP estimation in the zero-field Ising model on planar graphs is also tractable). For non-planar graphs, this cutting-plane method may or may not succeed in fully eliminating the integrality gap.

Maximum-Weight Perfect Matching in Planar Graphs

We discuss another method to solve the zero-field Ising model (or max-cut) on planar graphs by reduction to *maximum-weight perfect matching* [29, 102, 172, 203, 204]. This shows a connection to work in the statistical mechanics literature on tractable methods for computing the partition function of the so-called *dimer model* and the zero-field Ising model on planar graphs [79, 87, 128, 132]. This formulation is also interesting because it can be solved *directly* using a Gaussian elimination method with complexity $\mathcal{O}(n^{3/2})$ [164, 165].

We assume that zero-weight edges have been added to the planar graph $\mathcal{G}$ such that each face of $\mathcal{G}$ is now a triangle (that is, a face bounded by three edges). This procedure is shown in Figure 2.15(a) and (b). Now, all nodes of the dual graph $\mathcal{G}^*$ have degree three as seen in Figure 2.15(c). A number of works have shown that either max-cut or the zero-field planar Ising model can be solved by reduction to a matching problem [29, 102, 204]. We follow [172, 203], which employs a graphical method due to Kasteleyn [132] to reduce the problem to a maximum-weight perfect-matching problem defined on an auxiliary graph $\mathcal{G}_K^*$ based on the planar dual graph $\mathcal{G}^*$. The auxiliary graph $\mathcal{G}_K^*$ is obtained from the dual graph $\mathcal{G}^*$ by expanding each node of $\mathcal{G}^*$ into a fully-connected cluster of four nodes as seen in Figure 2.16(a). Each of the incoming edges is linked to a separate node within this cluster and the weights on these edges are copied from $\mathcal{G}^*$. The weights on the new edges within each cluster are set to zero. Now, it can be seen that there is a simple correspondence between (i) valid cuts in $\mathcal{G}$, (ii)

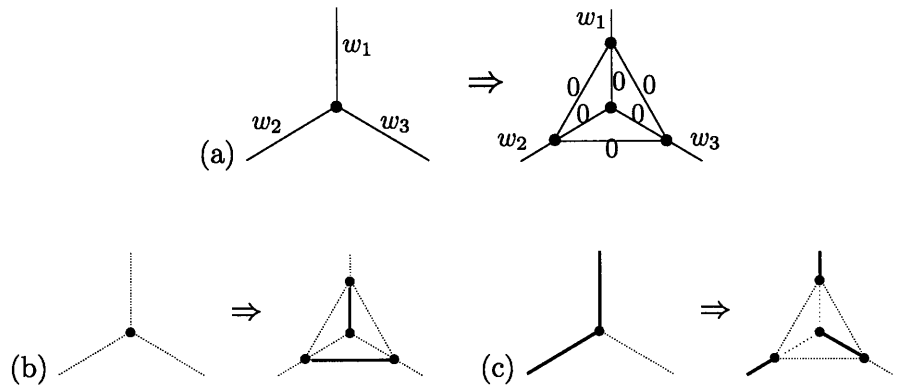

Figure 2.16. Illustration of construction of Kastelyn's graph $\mathcal{G}_K^*$ from a planar dual graph $\mathcal{G}^*$. We assume the $\mathcal{G}$ is a triangular, planar graph such that each node of $\mathcal{G}^*$ has degree three. (a) Each vertex of $\mathcal{G}^*$ (on the left) is replaced by a Kasteleyn cluster $\mathcal{G}_K^*$ (the four nodes on the right). This ensures that even-degree subgraphs of $\mathcal{G}^*$ map to perfect matchings in $\mathcal{G}_K^*$. (b) If an even-subgraph of $\mathcal{G}^*$ does not include any edges incident to this node, it corresponds to a perfect matching in $\mathcal{G}_K^*$ with two edges inside the cluster and no edges leaving the cluster, (c) If an even-subgraph in $\mathcal{G}^*$ includes two edges incident to this node, it corresponds to a perfect matching in $\mathcal{G}_K^*$ with two edges leaving the cluster and one edge inside the cluster. Thus, every perfect matching of $\mathcal{G}_K^*$ has an even number of edges leaving each cluster and therefore corresponds to an even-degree subgraph of $\mathcal{G}^*$.

even-degree subgraphs (unions of edge-disjoint cycles) of $\mathcal{G}^*$ and (iii) *perfect matchings* of $\mathcal{G}_K^*$, that is, any subset of edges $M \subset \mathcal{G}_K^*$ such that every node of $\mathcal{G}_K^*$ has exactly one of its edges in M. The correspondence between (ii) and (iii) is demonstrated in Figure 2.16(b) and (c). Thus, the problem of finding a maximum-weight cut of $\mathcal{G}$ is reduced to one of finding the maximum-weight perfect matching in the graph $\mathcal{G}_K^*$. This latter problem can be solved directly with $\mathcal{O}(n^{3/2})$ computation using the Gaussian elimination method of [165]. There are also iterative solution methods, based on the alternating paths method of Edmonds [57, 73], for perfect matching in general (non-planar) graphs. However, the reduction of MAP estimation to perfect matching only works for the zero-field planar Ising model.

Max-Flow/Min-Cut

As a final topic concerning combinatorial optimization methods, we review the max-flow/min-cut approach to MAP estimation in binary variable models.

The *max-flow* problem is defined on a *directed* graph with edge set $\mathcal{G} \subset V \times V$, and with *non-negative* edge capacities $w_E \geq 0$ for all $E \in \mathcal{G}$. Furthermore, two special nodes $s, t \in V$ are designated as the *source* and *sink* of the graph. We assume that there is one feedback edge $(t, s) \in \mathcal{G}$ with infinite capacity $w_{ts} = \infty$. Then, the *max-flow* problem is defined as follows. We define a *flow* variable $y_E \in [0, w_E]$ on each edge of the graph. Note that the flow across the edge is restricted by the edge capacity. We also require conservation of flow at each node, such that the total flow into a vertex is equal to the

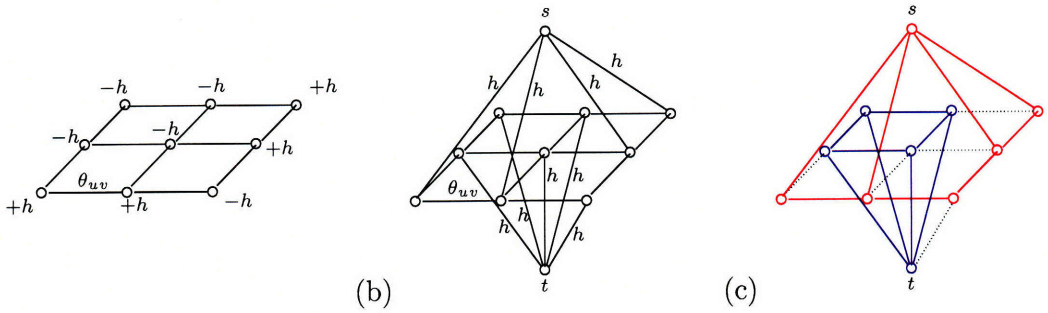

Figure 2.17. Illustration of min-cut/max-flow formulation of MAP estimation in binary models. (a) An Ising model with node potentials $\theta_v = \pm h$ ($h > 0$) and edge potentials $\theta_{u,v}$. (b) Auxiliary graph with two extra nodes s and t. Node s is connected to all nodes that had positive node potentials. Node t is connected to all nodes that had negative node potentials. The weight of the each new edge is set to the absolute value of the node potential. (c) The minimum-weight cut between s and t (dotted edges) also determines the MAP configuration. After cutting edges, nodes in the same component as s (the red subgraph) are set to $+1$ and those in the same component as t (the blue subgraph) are set to -1.

total flow out of the vertex. The net flow through the network is then equal to flow y_{ts} across the feedback edge (t, s). Then, we seek the maximum flow. The continuous LP relaxation of this problem is:

$$\begin{array}{ll} \text{maximize} & y_{ts} \\ \text{subject to} & y_E \in [0, w_E] \text{ for all } E \in \mathcal{G} \\ & \sum_{(u,v)\in\mathcal{G}} y_{u,v} = \sum_{(v,w)\in\mathcal{G}} y_{v,w} \text{ for all } v \in V \cup \{s,t\}. \end{array}$$

There are efficient, polynomial-time algorithms to solve the max-flow problem exactly in a finite number of steps [42, 171]. These methods are based on efficient algorithms to find *augmenting paths* from the source to the sink. The *min-cut problem* on this graph (not including the feedback edge) is defined as the minimum-weight cut of the graph such that the source and sink nodes are separated by the cut. This also can be formulated as an LP. Clearly, for any cut between s and t, the weight of the cut provides an upper-bound on the value of the any flow from s to t. In fact, the value of the minimum cut is equal to the maximum flow [82], so that max-flow algorithms can be used to solve this min-cut problem.

We now describe a min-cut formulation of MAP estimation in the ferromagnetic Ising model [98]. That is, we seek to maximize the energy function:

$$f(x) = \sum_{i\in V} \theta_i x_i + \sum_{\{i,j\}\in\mathcal{G}} \theta_{ij} x_i x_j \tag{2.86}$$

with respect to binary variables $x_i \in \{-1, +1\}$. It is required that the edge parameters θ_{ij} are positive for all $\{i, j\} \in \mathcal{G}$. However, the node parameters θ_i may be positive or negative. This is equivalent to solving the min-cut problem in an auxiliary graph based on $\mathcal{G}$. This auxiliary graph is based on vertices $V \cup \{s, t\}$. There are three types of edges

in this graph: (i) for each edge $\{i,j\} \in \mathcal{G}$, we define *two* directed edges, (i,j) and (j,i), with symmetric edge weights $w_{ij} = w_{j,i} = \theta_{ij}$, (ii) for each positive node parameter $\theta_i > 0$, we define an edge (s,i) with edge weight $w_{s,i} = \theta_i$, and (iii) for each negative node parameter $\theta_i < 0$, we define an edge (i,t) with weight $w_{i,t} = -\theta_i$. For example, the Ising model seen in Figure 2.17(a) may be solved as the min-cut problem shown in (b). Once the min-cut problem is solved, this determines a partitioning of the node set into two sets V_+ and V_-, which respectively include nodes s and t, for example, the red and blue nodes seen in Figure 2.17(c). The MAP estimate is then given by setting $x_i^* = +1$ for all $i \in V_+$ and $x_i^* = -1$ for all $i \in V_-$.

This method may be extended to solve a more general class of Markov random fields (with non-binary variables) if the model is defined by convex or submodular potentials [115, 136]. Also, there are a number of recently developed heuristic approaches, based on max-flow/min-cut methods, aimed at obtaining approximate solutions in the general case [38,138]. It is interesting to note that max-flow problems also arise as *dual problems* of MAP estimation [103].

■ 2.6 Inference in Gaussian Graphical Models

In this section we review a number of inference algorithms in Gaussian graphical models and also discuss the recently developed walk-sum view of Gaussian inference.

■ 2.6.1 The Information Form and Markov Structure

The *moment representation* of a Gaussian distribution is defined in terms of the *mean vector* $\hat{x} = \mathbb{E}\{x\}$ and *covariance matrix* $K = \mathbb{E}\{(x-\hat{x})(x-\hat{x})^T\}$. In terms of these parameters, the Gaussian distribution is defined

$$P(x) \propto \exp\{-\frac{1}{2}(x-\hat{x})^T K^{-1}(x-\hat{x})\}. \tag{2.87}$$

Here, K is a symmetric positive definite matrix ($K \succ 0$).[15] We also consider the *information form* defined by

$$P(x) = \exp\{-\frac{1}{2}x^T Jx + h^T x - \Phi(h,J)\} \tag{2.88}$$

where $J \succ 0$ is the *information matrix* and h the *potential vector.* The normalization constant $\Phi(h,J)$ is given by

$$\Phi(h,J) = \frac{1}{2}\left\{-\log\det J + h^T J^{-1} h + n\log 2\pi\right\} \tag{2.89}$$

[15]More generally, one may also consider degenerate Gaussian distributions in which the covariance matrix is only positive semi-definite ($K \succeq 0$), which corresponds to the random vector x being constrained to an affine subspace (e.g., defined by $Ax = b$) and being Gaussian distributed within this subspace. We focus on the non-degenerate case in the thesis.

Comparing these two forms, we see that they are equivalent, being related by:

$$K = J^{-1} \tag{2.90}$$
$$\hat{x} = J^{-1}h. \tag{2.91}$$

Gauss-Markov Structure

Let r_{ij} denote the *partial correlation coefficient* [145] between variables x_i and x_j, defined as:

$$r_{ij} \triangleq \frac{\text{cov}(x_i, x_j | x_{V\setminus ij})}{\sqrt{\text{var}(x_i|x_{V\setminus ij})\text{var}(x_j|x_{V\setminus ij})}} \tag{2.92}$$

In other words, r_{ij} represents the correlation coefficient between x_i and x_j with respect to the conditional distribution $P(x_i, x_j|x_{V\setminus ij})$, where $x_{V\setminus ij}$ denotes the set of all of variables except for x_i and x_j. A simple calculation shows that these coefficients are simply related to the information matrix [145]:

Proposition 2.6.1. *For a Gaussian distribution with information matrix J the partial correlation coefficients are given by*

$$r_{ij} = \frac{-J_{ij}}{\sqrt{J_{ii}J_{jj}}}. \tag{2.93}$$

Proof. It is simple to check that the conditional covariance of (x_i, x_j), after conditioning on $x_{V\setminus ij}$, is

$$K' \triangleq \begin{pmatrix} J_{ii} & J_{ij} \\ J_{ij} & J_{jj} \end{pmatrix}^{-1} = \frac{1}{\Delta}\begin{pmatrix} J_{jj} & -J_{ij} \\ -J_{ij} & J_{ii} \end{pmatrix} \tag{2.94}$$

where $\Delta = J_{ii}J_{ij} - J_{ij}^2$. Then, the correlation coefficient of this covariance matrix is given by

$$r_{ij} = \frac{K'_{ij}}{\sqrt{K'_{ii}K'_{jj}}} = \frac{-\Delta^{-1}J_{ij}}{\sqrt{(\Delta^{-1}J_{ii})(\Delta^{-1}J_{jj})}} = \frac{-J_{ij}}{\sqrt{J_{ii}J_{jj}}}. \tag{2.95}$$

□

As a consequence of this result, we see that the Markov structure of a Gaussian distribution is directly linked to the fill-pattern of the information matrix (see also [199]):

Proposition 2.6.2. *Let $\mathcal{G}$ be a pairwise graph. Then, a Gaussian distribution with information matrix J is Markov with respect to $\mathcal{G}$ if and only if $J_{ij} = 0$ for all $\{i, j\} \notin \mathcal{G}$.*

Thus, the graph $\mathcal{G}(J) \triangleq \{\{i, j\}|J_{ij} \neq 0\}$ describes the Markov structure of the Gaussian model with information matrix J. Also, the family of all Gauss-Markov models defined on $\mathcal{G}$ is represented by the set of all symmetric positive-definite matrices J that satisfy sparsity constraints: $J_{ij} = 0$ for all $\{i, j\} \notin \mathcal{G}$.

Gaussian Elimination and The Schur Complement

Given the information form (h, J), let us define the *marginal information form* $(\hat{h}_A, \hat{J}_A)$, on variables $A = V \setminus B$, by:

$$\begin{aligned} \hat{J}_A &= J_{A,A} - J_{A,B}(J_{B,B})^{-1}J_{B,A} \\ \hat{h}_A &= h_A - J_{A,B}(J_{B,B})^{-1}h_B \end{aligned} \tag{2.96}$$

The matrix $\hat{J}_A$ is known as the *Schur complement* of J with respect to the submatrix $J_{B,B}$ [110]. The marginal information form represents the result of eliminating the variables x_B in the system of equations $Jx = h$ to obtain the reduced set of equations $\hat{J}_A x_A = \hat{h}_A$ To see this, write $Jx = h$ as the system of equations:

$$J_{A,A}x_A + J_{A,B}x_B = h_A \tag{2.97}$$

$$J_{B,A}x_A + J_{B,B}x_B = h_B \tag{2.98}$$

To eliminate variables x_B from (2.97), we multiply both sides of (2.98) by $J_{A,B}(J_{B,B})^{-1}$, which gives

$$J_{A,B}(J_{B,B})^{-1}J_{B,A}x_A + J_{A,B}x_B = J_{A,B}(J_{B,B})^{-1}h_B. \tag{2.99}$$

Then, we subtract (2.99) from (2.97) to obtain the result:

$$\underbrace{[J_{A,A} - J_{A,B}(J_{B,B})^{-1}J_{B,A}]}_{\hat{J}_A} x_A = \underbrace{[h_A - J_{A,B}(J_{B,B})^{-1}h_B]}_{\hat{h}_A} \tag{2.100}$$

Thus, solving $\hat{J}_A x_A = \hat{h}_A$ for x_A gives us part of the solution to $Jx = h$, that is, it gives us $\hat{x}_A$. In practice, this reduction is accomplished using the well-known *Gaussian elimination* procedure [96], which essentially involves iterative application of (2.96) to eliminate the variables $(x_v, v \in B)$ one at a time.

By a similar argument, with respect to the matrix equation $JK = I$, we can also see that $[J_{A,A} - J_{A,B}(J_{B,B})^{-1}J_{B,A}]K_{A,A} = I_{A,A}$ and hence $K_{A,A} = [J_{A,A} - J_{A,B}(J_{B,B})^{-1}J_{B,A}]^{-1}$. Thus, we have demonstrated the following result:

Proposition 2.6.3. *Let $K = J^{-1}$ and $\hat{x} = J^{-1}h$. Then, it holds that $K_{A,A} = (\hat{J}_A)^{-1}$ and $\hat{x}_A = (\hat{J}_A)^{-1}\hat{h}_A$ with $\hat{J}_A$ and $\hat{h}_A$ defined by (2.96).*

In other words, $(\hat{h}_A, \hat{J}_A)$ represents the information form of the marginal statistics $(\hat{x}_A, K_{A,A})$. Because the marginals of a Gaussian distribution are also Gaussian, it is implied that variable elimination by integration reduces to Gaussian elimination:

$$P(x_A) \triangleq \int P(x_A, x_B)dx_B \propto \exp\left\{-\tfrac{1}{2}x_A^T\hat{J}_A x_A + \hat{h}_A^T x_A\right\} \tag{2.101}$$

It is also simple to verify that variable elimination by maximization (such as in the max-sum algorithm) likewise reduces to Gaussian elimination:

$$\max_{x_B}\left\{-\tfrac{1}{2}x^T Jx + h^T x\right\} = -\tfrac{1}{2}x_A^T\hat{J}_A x_A + \hat{h}_A^T x_A \tag{2.102}$$

This is shown by computing the gradient with respect to x_B, setting this gradient to zero and solving for x_B, which gives $x_B = -(J_{B,B})^{-1}(h_B + J_{B,A}x_A)$. Substituting this for x_B in the objective function, we obtain (2.102). Thus, inference in Gaussian graphical models (in either the sum-product or max-sum sense) reduces to Gaussian elimination calculations.

However, consistent with our earlier discussion of recursive inference methods (Section 2.4.1), one can see that variable elimination in Gaussian graphical models generally results in *fill edges*, due to the matrix inverse in (2.96) being a full matrix so as to create new pairwise interactions between neighbors of an eliminated node. This results in the computational complexity of inference being cubic in the tree-width of the graph.

■ 2.6.2 Gaussian Inference Algorithms

Junction Tree Algorithm

We now specify an efficient version of the junction tree recursive inference procedure for Gaussian graphical models. The upward sweep of this procedure is equivalent to Gaussian elimination but the downward sweep performs a recursive back-substitution procedure. This form of the algorithm is more efficient than using Gaussian elimination in both sweeps, and more closely follows standard methods developed in the linear algebra literature for solving sparse linear systems (e.g., by sparse Cholesky factorization, using Gaussian elimination, followed by back-substitution).

Let $\mathcal{T}$ be a junction tree of the chordal graph $\mathcal{G}$ (see Section 2.4.1). In this section, we use γ to denote a node of this junction tree, and write C_γ to denote the corresponding clique of $\mathcal{G}$. We obtain a directed version of $\mathcal{T}$ by selecting an arbitrary node to be the root of the tree and then orienting edges to point away from the root. In this directed tree, let $\pi(\gamma)$ denote the parent of node γ. Let $S_\gamma \triangleq C_\gamma \cap C_{\pi(\gamma)}$ denote the separator associated with edge $(\pi(\gamma), \gamma)$ of the junction tree. Also, we define $R_\gamma \triangleq C_\gamma \setminus C_{\pi(\gamma)}$ at each node. At the root node, $S_\gamma = \emptyset$ and $R_\gamma = C_\gamma$.

Now, we specify our two-sweep recursive inference procedure. The input to this procedure is the potential vector h and the sparse matrix J, which is defined over a chordal graph $\mathcal{G}$ with junction tree $\mathcal{T}$. The output of this procedure is the mean vector (MAP estimate) $\hat{x}$ and a sparse matrix K, defined over the same chordal graph $\mathcal{G}$, which then stores a subset of elements of the matrix inverse $K = J^{-1}$, that is, it only stores those diagonal elements of J^{-1} and off-diagonal elements corresponding to edges of $\mathcal{G}$.

Upward Sweep For each node γ of the junction tree, starting from the leaves of the tree and working upwards, we perform the following computations in the order shown:

$$\begin{aligned} Q_\gamma &= (J_{R_\gamma,R_\gamma})^{-1} \\ A_\gamma &= -Q_\gamma J_{R_\gamma,S_\gamma} \\ h_{S_\gamma} &\leftarrow h_{S_\gamma} + A_\gamma^T h_{R_\gamma} \\ J_{S_\gamma,S_\gamma} &\leftarrow J_{S_\gamma,S_\gamma} + J_{S_\gamma,R_\gamma} A_\gamma \end{aligned} \tag{2.103}$$

This upward pass performs Gaussian elimination in J. In each step, the subvector of h and principle sub-matrix of J indexed by S_γ are overwritten. We denote this operation by "$\leftarrow$" in the last two lines of (2.103). Note, that this "update" of those submatrices serves to propagate information up the tree as the procedure progresses, by virtue of the parent node $\pi(\gamma)$ subsequently accessing these submatrices corresponding to the shared variables $S_\gamma = C_\gamma \cap C_{\pi(\gamma)}$. Also, the matrices A_γ and Q_γ, computed at each node of the junction tree in this upward sweep, must be stored as they are used again in the subsequent downward sweep. These matrices specify the conditional probability distribution $P(x_{R_\gamma}|x_{S_\gamma})$, via an auto-regressive model:

$$x_{R_\gamma} = A_\gamma x_{S_\gamma} + w_\gamma \tag{2.104}$$

where $w_\gamma \sim \mathcal{N}(0, Q_\gamma)$. This model is used in the downward sweep to propagate moments (means and covariances) back down the tree.

Downward Sweep For each node γ of the junction tree, starting from the root node and working down the tree, we perform the following calculations in the order shown:

$$\hat{x}_{R_\gamma} \leftarrow A_\gamma \hat{x}_{S_\gamma} \tag{2.105}$$

$$\begin{aligned} K_{R_\gamma,S_\gamma} &\leftarrow A_\gamma K_{S_\gamma,S_\gamma} \\ K_{S_\gamma,R_\gamma} &\leftarrow (K_{R_\gamma,S_\gamma})^T \\ K_{R_\gamma,R_\gamma} &\leftarrow K_{R_\gamma,S_\gamma} A_\gamma^T + Q_\gamma \end{aligned} \tag{2.106}$$

Note that in this back-substitution form of Gaussian inference, the upward sweep operates on the information form (h, J) whereas the downward sweep is in terms of the moment parameterization $(\hat{x}, K)$. Each step of the downward sweep uses the auto-regressive model specified by (A_γ, Q_γ), constructed during the preceding upward sweep, to efficiently propagate clique moments back down the tree. That is, once the moments $(\hat{x}_{C_\gamma}, K_{C_\gamma,C_\gamma})$ have been set, this is sufficient to predict the moments at all of the children of γ in the junction tree, thereby implementing the downward propagation of moments. In traditional back-substitution methods [96], only the first line of these calculations is performed, as this suffices to compute the solution $\hat{x}$. The additional calculations compute the desired elements of the covariance matrix K. We also note that no additional matrix inverse calculations are required in the downward sweep. For this reason, the back-substitution method is somewhat more efficient than using Gaussian elimination.[16]

Once the downward sweep is completed, the vector $\hat{x}$ is equal to $J^{-1}h$ and the matrix K now stores a sparse subset of the elements of the inverse matrix J^{-1}, including all diagonal elements and those off-diagonal entries corresponding to edges of the chordal graph $\mathcal{G}$. This subset of elements determines the moment parameters η in the exponential family of Gaussian graphical models defined on $\mathcal{G}$. Thus, we have specified an efficient algorithm to implement the mapping $\Lambda : \theta \rightarrow \eta$ for Gaussian graphical models.

[16]We also comment that, in the computer code for this downward sweep, one may actually write the outputs $\hat{x}$ and K to the same data structure that initially stored the inputs h and J so as to make efficient use of memory.

Gaussian Belief Propagation

We describe a parametric form of iterative belief propagation in Gaussian graphical models. The Gaussian distribution may be factored into node and edge factors based on its information form:

$$P(x) \propto \prod_{v \in V} \psi_v(x_v) \prod_{\{u,v\} \in \mathcal{G}} \psi_{uv}(x_u, x_v) \tag{2.107}$$

with

$$\begin{aligned} \psi_v(x) &= \exp\{-\tfrac{1}{2} J_{vv} x_v^2 + h_v x_v\} \\ \psi_{uv}(x_u, x_v) &= \exp\{-J_{uv} x_u x_v\} \end{aligned} \tag{2.108}$$

We specify functional messages $\mu_{u \to v}(x_v)$ on the edges of the graph as

$$\mu_{u \to v}(x_v) \propto \exp\{-\frac{1}{2} \Delta J_{u \to v} x_v^2 + \Delta h_{u \to v} x_v\}. \tag{2.109}$$

In practical terms, it is the *parameters* $(\Delta h_{u \to v}, \Delta J_{u \to v})$ of these functions that actually serve as messages in Gaussian belief propagation. We initially set these parameters to zero, $\Delta h_{u \to v} = 0$ and $\Delta J_{u \to v} = 0$, corresponding to uninformative initial messages $\mu_{u \to v}(x_v) = 1$ for all x_v.

Now, we perform iterative belief propagation (in either the sum-product or max-product sense) using the factorization (2.108) based on the information form. This reduces to the following operations in terms of the message parameters. To calculate the message from node u to v, given the other messages into node u, we first combine messages at node v with the node factor ψ_v. In terms of information parameters, this becomes:

$$\begin{aligned} \hat{h}_{u \backslash v} &= h_u + \sum_{w \in \partial u \backslash v} \Delta h_{w \to u} \\ \hat{J}_{u \backslash v} &= J_{u,u} + \sum_{w \in \partial u \backslash v} \Delta J_{w \to u} \end{aligned} \tag{2.110}$$

Then, multiplying by $\psi_{u,v}$ and integrating (or maximizing!) over x_u, the message $\mu_{u \to v}(x_v)$ has parameters:

$$\begin{aligned} \Delta h_{u \to v} &= -J_{v,u} (\hat{J}_{u \backslash v})^{-1} \hat{h}_{u \backslash v} \\ \Delta J_{u \to v} &= -J_{v,u} (\hat{J}_{u \backslash v})^{-1} J_{u,v} \end{aligned} \tag{2.111}$$

Equivalently, this procedure may also be viewed as performing *Gaussian elimination* with respect to the information form defined on the computation tree of the graph.[17]

[17] In fact, this latter perspective suggests that Gaussian belief propagation might also be useful for solving more general linear systems of equations (e.g., non-symmetric or indefinite systems), even though this no longer corresponds to a probabilistic inference procedure. Here, however, we focus only on the symmetric, positive-definite case (corresponding to inference in a Gaussian graphical model).

One may also view Gaussian belief propagation as being parameterized by the information parameters $(\hat{h}_{u\backslash v}, \hat{J}_{u\backslash v})$, which specify the conditional distribution $P(x_u|x_v = 0)$. This latter representation can be expressed in terms of the moment parameters:

$$\begin{aligned} \hat{x}_{u\backslash v} &\triangleq (\hat{J}_{u\backslash v})^{-1}\hat{h}_{u\backslash v} \\ K_{u\backslash v} &\triangleq (\hat{J}_{u\backslash v})^{-1} \end{aligned} \tag{2.112}$$

In this representation, the Gaussian belief propagation equations are:

$$\begin{aligned} K_{u\backslash v} &= (J_{u,u} - \sum_{w\in\partial u\backslash v} J_{u,w}K_{w\backslash u}J_{w,u})^{-1} \\ \hat{x}_{u\backslash v} &= K_{u\backslash v}(h_u - \sum_{w\in\partial u\backslash v} J_{u,w}\hat{x}_{w\backslash u}) \end{aligned} \tag{2.113}$$

Some may prefer this representation as it is expressed in terms of the familiar moment representation of Gaussian messages. However, these various forms are equivalent, being related by a simple change of variables.

In any case, estimates of the marginal distribution at each node are ultimately obtained by combining all of the messages to a node. In the information form, this results in adding messages:

$$\begin{aligned} \hat{h}_v &= h_v + \sum_{u\in\partial v} \Delta h_{u\to v} \\ \hat{J}_v &= J_{v,v} + \sum_{u\in\partial v} \Delta J_{u\to v} \end{aligned} \tag{2.114}$$

The final estimates of marginal moments are then given by:

$$\begin{aligned} K_{v,v} &= (\hat{J}_{v,v})^{-1} \\ \hat{x}_v &= K_{v,v}\hat{h}_v \end{aligned} \tag{2.115}$$

As is well-known, if Gaussian belief propagation converges, then the mean estimates $\hat{x}_v$ are correct [217]. This then solves the MAP estimation problem in Gaussian models. However, the variance estimates are generally incorrect in loopy graphs, but may still provide a useful approximation. A sufficient condition for convergence of Gaussian belief propagation is given in [217] that is equivalent to the information matrix J being diagonally dominant, that is, $J_{ii} > \sum_{j\neq i} |J_{ij}|$ for all i. Generalizations of this condition are developed in later work [124, 157], which we describe further in Section 2.6.3.

Iterative Methods and Embedded Trees

We briefly discuss linear methods for iterative solution of $Jx = h$ where J is a sparse matrix [96, 206]. These methods are based on a *preconditioner* M^{-1}, which is a linear operator that approximates multiplication by J^{-1} and is easy to compute; more precisely, there is a fast algorithm for solving $Mx = b$ for x given b. Then, starting from

any initial guess $\hat{x}^{(0)}$, the method proceeds by computing a sequence of estimates $\hat{x}^{(s)}$ according to the equation:

$$\hat{x}^{(s+1)} = \hat{x}^{(s)} + M^{-1}(h - J\hat{x}^{(s)}) \quad (2.116)$$

$$\triangleq M^{-1}(h + K\hat{x}^{(s)}) \quad (2.117)$$

In the second line we define $K \triangleq M - J$ (it should also be tractable to apply K to a vector). For instance, the classical *Gauss-Jacobi method* [96] chooses $M = \text{Diag}(J)$, the diagonal part of J, so that it is tractable to multiply by M^{-1} (because M is a diagonal matrix) or by $K = M - J$ (because K is a sparse matrix).

This procedure may be viewed as an iterative correction strategy. At each step, we seek to improve the previous estimate based on the residual error $h^{(s)} \triangleq h - J\hat{x}^{(s)}$. The exact estimation error $e^{(s)} \triangleq J^{-1}h - \hat{x}^{(s)} = J^{-1}h^{(s)}$ solves the *defect equation*: $Je^{(s)} = h^{(s)}$. We compute an approximate correction by instead solving $M\tilde{e}^{(s)} = h^{(s)}$, adding $\tilde{e}^{(s)} = M^{-1}h^{(s)}$ to the previous estimate to obtain $\hat{x}^{(s+1)}$. Using the same preconditioner at every iteration, this defines a linear system that converges if $\rho(M^{-1}K) < 1$.[18] If the method does converge, it then yields the correct solution (any fixed point of the algorihtm has zero residual error). One may also use non-stationary cyclic iterations that iteratively cycle over a set of preconditioners $M_1, \ldots, M_L$. Then, the method converges if $\rho(M_L^{-1}K_L \cdots M_1^{-1}K_1) < 1$ where $K_s = M_s - J$.

The *embedded trees (ET) algorithm* [201], and related methods [45, 66], were developed to take advantage of fast algorithms for solving $Jx = h$ when $\mathcal{G}(J)$ is a tree (or some other thin subgraph for which exact inference is fast). In such thin models, there are efficient solution techniques with linear complexity in the number of variables (for example, sparse Cholesky factorization followed by back-substitution [96]). This suggests the use of iterative methods for solving problems on sparse loopy graphs using preconditioners based on spanning trees of the graph. That is, given an embedded tree $\mathcal{T} \subset \mathcal{G}$, one defines the preconditioner M to be equal to J on the diagonal and on off-diagonal entries corresponding to edges of the tree. The remaining elements of M are set to zero. It was also proposed [201] to use cyclic iterations based on a collection of embedded trees that collectively cover all edges of the graph. It was found that such methods were often able to rapidly solve large, sparse linear systems and that the performance of these methods is competitive with more standard methods such as the conjugate gradients algorithm.

■ 2.6.3 Walk-Sum View of Gaussian Inference

We briefly summarize our work on the walk-sum view of Gaussian inference, done in collaboration with D. Malioutov [124, 157] and V. Chandrasekaran [45, 47].

[18] We let $\rho(A)$ denotes the *spectral radius* of the matrix A, which is the maximum of the absolute values of the eigenvalues of A [110].

Definition of Walk-Sums

The main idea, introduced in [122], is to express inference (e.g., computation of means and variances) as computing *walk-sums* in the Gaussian graphical model, that is, computing weighted sums of walks in the graph, where a *walk* is defined to be any sequence of vertices $(w_0, \ldots, w_\ell)$ such that $\{w_{s-1}, w_s\} \in \mathcal{G}$ for $s = 1, \ldots, \ell$. This paradigm is based on the *Neumann series* for the inverse of the matrix $J = I - R$,

$$(I - R)^{-1} = I + R + R^2 + R^3 + \ldots, \tag{2.118}$$

which is a generalization of the geometric series and holds if $\rho(R) < 1$. We also note that this series is closely linked to the *Gauss-Jacobi method.*

Here, we assume that J is rescaled to have unit-diagonal ($J_{vv} = 1$ for all $v \in V$). Then, $R = I - J$ is zero-diagonal ($R_{vv} = 0$ for all v) and has the partial correlation coefficient r_{uv} as its off-diagonal elements. Thus, R is sparse according to $\mathcal{G}$ and powers of the matrix R simply accumulate sums over walks in $\mathcal{G}$, that is, $(R^\ell)_{uv}$ is a sum over all ℓ-step walks in $\mathcal{G}$ from nodes u to v, with the weight of a walk $w = (w_0, \ldots, w_\ell)$ defined as the product of edge-weights $\phi(w) = \prod_{s=0}^{\ell-1} r_{v_s, v_{s+1}}$. This suggests the following formal interpretation of inference in Gaussian graphical models:

$$K_{uv} = \sum_{w: u \to v} \phi(w) \tag{2.119}$$

$$\hat{x}_v = \sum_{w: * \to v} h_{w_0} \phi(w) \tag{2.120}$$

The walk-sum in the first line is taken over the set of all walks that begin at node u and end at node v. In particular, variances K_{vv} correspond to the sum over all *self-return* walks of at node v, that is, walks that begin and end at v. The walk-sum in the second line is taken over the set of all walks which end at node v, where we multiply the weight of each walk by h_{w_0}, the value of h at the starting point of the walk. It is important to note that, because walks may revisit nodes multiple times, there are *infinitely* many walks in each of these sums (in connected graphs). Hence, walk-sums may fail to converge, and convergence may depend upon the order in which walks are included in the sum. We recall from basic analysis [184] that the value of a series $\sum_{k=1}^{\infty} a_k$ is invariant to reordering of its terms if and only if the series converges *absolutely*, that is, if and only if $\sum_k |a_k|$ converges. Hence, we say that a Gaussian graphical model is *walk-summable* if the walk-sum for K_{uv} converges absolutely for all $u, v \in V$. As it turns out, the condition $\rho(R) < 1$ is necessary but *not sufficient* for this walk-summable property to hold. Instead, we need the following more restrictive condition:

Proposition 2.6.4 (Walk-Summability). *All of the following conditions are equivalent:*

- *The model $J = I - R$ is walk-summable, that is, the formal walk-sums defined in (2.119) are well-defined (converge absolutely).*

- *The spectral radius condition $\rho(|R|) < 1$, where $|R|$ denotes the matrix of absolute values of elements of R.*[19] *Equivalently, the matrix series $\sum_\ell |R|^\ell$ converges.*

- *The model $J' = I - |R|$ is valid ($J' \succ 0$). In other words, we may negate any negative edge weights in R, to obtain an* attractive *model with all positive edge weights, and this attractive model is still a valid Gaussian model.*

- *The matrix J is* pairwise normalizable, *that is, there exists a decomposition $J = \sum_{E \in \mathcal{G}} J_E$ in which each J_E is non-zero only on a two-by-two submatrix corresponding to edge E and this submatrix is positive definite. This includes the class of diagonally dominant models.*

The proof uses standard results of analysis [184] and iterative methods [206] and also makes use of Perron-Frobenius theory [110]. As a corollary, we also have that all valid *non-frustrated* models (that can be transformed to an attractive model by negating a subset of variables) are walk-summable, which includes attractive models and cycle-free models (e.g., trees) as special cases. We refer the reader to [157] for complete proofs.

The main use of this walk-sum picture is that it provides a graphical approach to analysis of the convergence of iterative algorithms. This idea was first suggested in [122] and has since been expanded upon to analyze Gaussian belief propagation [124, 157] and a wide class of iterative linear methods including the embedded trees algorithm [45, 47, 122]. The basic recipe here involves demonstrating the following: (1) the iterative algorithm can be interpreted as computing a sequence of walks-sums $\phi_s = \phi(W_s) = \sum_{w \in W_s} \phi(w)$ over walk-sets W_s for $s = 1, 2, \ldots$; (2) the sets $\{W_s\}$ are *nested* $W_s \subset W_{s+1}$; and (3) the sets $\{W_s\}$ are *complete* such that $\cup_{s=1}^\infty W_s$ is equal to the set of walks W for which we would compute the walk-sum $\phi(W)$. Once this is done, the sum-partition theorem for absolutely convergent series,[20] implies that the walk-sums computed by the algorithm converge to the correct value, that is, $\lim_{s \to \infty} \phi(W_s) = \phi(\cup_{s=0}^\infty W_s) = \phi(W)$. The simplest (trivial) example of this idea is the Gauss-Jacobi algorithm itself, which is guaranteed to converge for walk-summable models. However, the ability to reorder walks arbitrarily (in walk-summable models) allows the walk-sum idea to be applied to a much broader class of algorithms.

Walk-Sum Interpretation of The Embedded-Trees Algorithm

We now describe the walk-sum interpretation of the embedded trees (ET) algorithm [45, 47]. Recall that ET uses a sequence of spanning trees $\mathcal{T}_s \subset \mathcal{G}$ to define preconditioners in the iterative method:

$$\hat{x}^{(s)} = M_s^{-1}(h + K_s \hat{x}^{(s-1)}) \tag{2.121}$$

[19] It holds that $\rho(R) \leq \rho(|R|)$, so that walk-summability implies $\rho(R) < 1$ and convergence of the Neumann series $\sum_\ell R^\ell$. Also, $\rho(R) < 1$ implies that $J = I - R \succ 0$ so that walk-summability implies validity of the model.

[20] Basically, the sum-partition theorem allows us to assert that, for walk-summable models, it holds that $\sum_{k=1}^\infty \sum_{w \in W_k} \phi(w) = \sum_{w \in \cup_k W_k} \phi(w)$ where the sets $\{W_k\}$ are mutually disjoint.

To show that this procedure computes walk-sums, we define the analogous walk-sets $W_k^{(s)}$ for each $k \in V$:

$$W_k^{(s)} = \cup_{j \in V} \left(\left(\cup_{(i,j) \notin \mathcal{T}_s} W_i^{(s-1)} \times (i,j) \right) \cup \{(j)\} \right) \times \{ j \to k | \mathcal{T}_s \} \tag{2.122}$$

Here, $\{j \to k|\mathcal{T}_s\}$ denotes the set of all walks from j to k in $\mathcal{T}_s$ and "$\times$" represents the operation of concatenating together pairs of walks to form longer walks (provided the second walk begins where the first walk ended). The construction of these walk-sets directly mirrors the algebraic form of (2.121). It then follows that $\hat{x}_v^{(s)} = \phi(W_v^{(s)})$ for each $v \in V$, which provides the walk-sum interpretation of ET. The main insight here is that applying the preconditioner M_s^{-1} corresponds to performing inference over the embedded tree $\mathcal{T}_s$, which generates all walks through this tree. More precisely, the set W_s is composed of the union of two sets: (1) walks that live entirely within $\mathcal{T}_s$ and (2) extensions of previously collected walks that are extended by stepping across one of the cut edges $(i,j) \in \mathcal{G} \setminus \mathcal{T}_s$ and then continue on in $\mathcal{T}_s$.

Another point to note here concerns validity of the preconditioner M_s based on $\mathcal{T}_s$. In general, ET may be *ill-posed* if the information matrix M_s based on $\mathcal{T}_s$ is indefinite or singular. However, in walk-summable models, it holds that every embedded tree is walk-summable (because absolute convergence of walk-sums in $\mathcal{G}$ implies absolute convergence of walk-sums within any subgraph of $\mathcal{G}$). This implies that all ET preconditioners are positive definite, because walk-summability of the tree $\mathcal{T}$ implies that $\rho(R_T) < 1$, where R_T is the matrix of edge-weights over $\mathcal{T}$, and $M \triangleq I - R_T \succ 0$. Thus, ET is well-posed in walk-summable models. Next, to show convergence of ET (for walk-summable models), we need to show that the sets $W_v^{(s)}$ are nested for each v, which is not at all obvious in the general case of non-stationary iterations. However, this was shown in [47], which also analyzes a more general class of iterative methods and proposes a method of *adaptively* selecting trees within the ET algorithm. In [45], this method was extended to adaptively select thin subgraph preconditioners.

Walk-Sum Interpretation of Gaussian Belief Propagation

We now discuss the walk-sum interpretation of Gaussian belief propagation [124, 157]. The main tool here is again the computation tree. Because belief propagation is equivalent to exact inference in the computation tree, we immediately obtain the following walk-sum interpretation of the marginal estimates produced by belief propagation:

$$\hat{x}_v^{(s)} = \sum_{w: * \to v | \mathcal{T}_v^{(s)}} h_{w_0} \phi(w) \tag{2.123}$$

$$\hat{K}_v^{(s)} = \sum_{w: v \to v | \mathcal{T}_v^{(s)}} \phi(w) \tag{2.124}$$

Here, both walk-sums are taken in $\mathcal{T}_v^{(s)}$, the k-step *computation tree* rooted at node v. The first sum is over all walks which terminate at the root node of this computation

tree (but may start anywhere in the tree) and the second sum is over the set of self-return walks which begin and end at the root node of the computation tree. These walk-sets are nested, as are the computation trees, so that, for walk-summable models, BP estimates converge to walk-sums taken in the infinite computation tree. In the case of the mean estimates, one can see that there is one-to-one correspondence between walks that end at the root node of the $\mathcal{T}_v$ and walks in $\mathcal{G}$ that end at node v. Hence, the BP means converge to the correct means. The variances, however, are another story. Every self-return walk at the root node of $\mathcal{T}_v$ is also a self-return walk at node v in $\mathcal{G}$. However, some self-return walks of $\mathcal{G}$ are *not* self-return walks in the computation tree (e.g., walks which go around a cycle once and then stop). Hence, BP only captures a subset of the self-returns walks in $\mathcal{G}$ and the final variance estimates are therefore approximate in loopy graphs.

Note that this analysis relies on the walk-sum interpretation of inference in the computation tree. This is valid for walk-summable models defined on $\mathcal{G}$, because walk-summability in $\mathcal{G}$ implies walk-summability of each of its computation trees (and hence positive definiteness of the information matrices defined on these computation trees). It is also interesting to note that the messages within Gaussian belief propagation also have a simple walk-sum interpretation:

$$\Delta J_{u\to v} = \sum_{w:u\to v|\mathcal{T}_{u\backslash v}} \phi(w) \tag{2.125}$$

$$\Delta h_{u\to v} = \sum_{w:*\to v|\mathcal{T}_{u\backslash v}} h_{w_0}\phi(w) \tag{2.126}$$

where $\mathcal{T}_{u\backslash v}$ denotes the subtree rooted at u but excluding its neighbor v. The walk-sum for $\Delta J_{u\to v}$ is taken over the set of all *single-revisit* self-return walks at node v into the subtree $\mathcal{T}_{u\backslash v}$. The walk-sum for $\Delta h_{u\to v}$ includes all *single-visit* walks to node v that start somewhere in $\mathcal{T}_{u\backslash v}$ and then end at node v (never visiting v before).

We also remark that, based on this walk-sum interpretation of inference in Gaussian models defined on trees, it is possible to explicitly derive a two-pass algorithm for computing the necessary walk-sums at each node. However, upon comparing the resulting equations to the Gaussian BP equations, we see that they are essentially equivalent. Hence, this provides a walk-sum interpretation of the individual messages in Gaussian BP. We refer the reader to [124, 157] for further details.

■ 2.7 Learning Graphical Models

In this final section of the chapter, we review methods for learning graphical models from sample data. We first focus on the problem of parameter estimation over a given graph and then discuss some ideas concerning learning the graph structure.

■ 2.7.1 Maximum-Likelihood and Information Projection

To begin, let us assume that the graph $\mathcal{G}$ of the graphical model is given and we wish to select the potential functions over this graph to maximize the likelihood of a set of independent samples $\tilde{x}^s \in \mathbb{X}^V$ for $s = 1, \ldots, N$ of the random variables $x = (x_v, v \in V)$ that we are modeling. This is equivalent to minimizing the KL-divergence $D(\tilde{P}, P)$ of our graphical model P relative to the sample distribution $\tilde{P}$.[21] In exponential family graphical models, the sample moments $\tilde{\eta} = \mathbb{E}_{\tilde{P}}\{\phi(x)\} = \frac{1}{N}\sum_{s=1}^{N}\phi(\tilde{x}^s)$ are *sufficient statistics* of the sample data and learning reduces to the following convex optimization problem over the parameters $\theta \in \Theta(\mathcal{G})$ (the potential specification of the model):

$$\begin{array}{ll} \text{minimize} & \mathcal{F}(\theta) \triangleq \Phi(\theta) - \tilde{\eta}^T\theta \\ \text{subject to} & \theta \in \Theta(\mathcal{G}) \end{array} \tag{2.127}$$

This corresponds to the information projection (2.31) of $\tilde{P}$ to the e-flat submanifold determined by requiring sparsity with respect to $\mathcal{G}$. Recall that $\Phi(\theta)$ is the normalization function (log-partition function) of the graphical model, also known as free energy in statistical physics, satisfying the moment-generating property $\nabla\Phi(\theta) = \Lambda(\theta) = \eta$. Thus, learning amounts to solving the moment-matching condition $\Lambda(\theta) = \tilde{\eta}$. In discrete-variable graphical models, this is equivalent to *marginal-matching* with respect to $\mathcal{G}$, that is, requiring that $P(x_E) = \tilde{P}(x_E)$ for all $E \in \mathcal{G}$. Indeed, in the over-parameterized representation of the model, the moments are precisely marginal distributions. However, even using minimal representations (e.g., for the Boltzmann and Ising models) moment-matching is equivalent to marginal-matching.

In Gaussian models, learning involves matching *means*, *variances* and *edge-wise correlations* over the pairwise graph $\mathcal{G}$. That is, given sample statistics $\tilde{x} = \frac{1}{N}\sum_s \tilde{x}^s$ and $\tilde{K} = \frac{1}{N}\sum_s(\tilde{x}^s - \tilde{x})(\tilde{x}^s - \tilde{x})^T$, we seek the Gaussian graphical model specified by (h, J), with J being sparse with respect to $\mathcal{G}$, for which the corresponding moments $\hat{x} = J^{-1}h$ and $\hat{K} = J^{-1}$ satisfy the constraints:

$$\hat{x}_v = \tilde{x}_v \text{ and } K_{vv} = \tilde{K}_{vv} \tag{2.128}$$

for all $v \in V$ and

$$K_{uv} = \tilde{K}_{uv} \tag{2.129}$$

for all $\{u, v\} \in \mathcal{G}$. These moment-matching conditions are equivalent to marginal-matching in Gaussian models (that is, if we take $\tilde{P}$ to be the Gaussian distribution with moments $\tilde{x}$ and $\tilde{K}$). Note that only a partial specification of the matrix $\tilde{K}$ is used here, the correlations $\tilde{K}_{uv}$ for $\{u, v\} \notin \mathcal{G}$ are not enforced and do not need to be computed.

[21] The sample distribution is defined $\tilde{P}(x) \triangleq \frac{1}{N}\sum_{s=1}^{N}\delta(x; \tilde{x}^s)$ where $\delta(x; \tilde{x}^s)$ is a probability distribution over x concentrated at $\tilde{x}^s$ (the Kronecker-delta function for discrete $\mathbb{X}$ or the Dirac-delta function for continuous $\mathbb{X}$). Minimizing $D(\tilde{P}, P) = -(H(\tilde{P}) + \frac{1}{N}\sum_s \log P(\tilde{x}^s))$ is equivalent to maximizing $\log P(\tilde{x}^1, \ldots, \tilde{x}^N) = \sum_s \log P(\tilde{x}^s)$ for independent samples $\{\tilde{x}^s\}$.

Projection to Chordal Graphs

In the special case that we seek to learn a Markov model over a *chordal* graph $\mathcal{G}$, the projection problem has a closed-form solution. Using the junction-tree factorization, any probability distribution $P(x)$ that is Markov on $\mathcal{G}$ can be factored in terms of its marginal distributions on cliques $C \in \mathcal{C}(\mathcal{G})$ and separators $S \in \mathcal{S}(\mathcal{G})$ of a junction tree of the graph:

$$P(x) = \prod_C P_C(x_C) \prod_S P_S^{-1}(x_S) \tag{2.130}$$

$$\triangleq \prod_C \psi_C(x_C) \prod_S \psi_S(x_S) \tag{2.131}$$

Moreover, any set of probability distributions $\{P_C\}$ and $\{P_S\}$, that are consistent with respect to marginalization,[22] may be plugged into this formula and it defines a consistent probability distribution having these distributions as its marginals. In terms of potential functions, this gives an energy function

$$f(x) = \sum_C \theta_C(x_C) + \sum_S \theta_S(x_S) \tag{2.132}$$

with potentials defined by $\theta_C(x_C) = \log P_C(x_C)$ and $\theta_S(x_S) = -\log P_S(x_S)$. Thus, using an over-parameterized representation of the graphical model, this gives a simple "projection" formula.[23] We simply plug-in the sample marginals $\tilde{P}_C$ and $\tilde{P}_S$ in the preceding formulas to obtain the projection of $\tilde{P}$ to $\Theta(\mathcal{G})$.

This same idea can be extended to compute projections using a minimal representation of the model. From the junction-tree factorization, we take logarithms on both sides and compute the expectation with respect to P to obtain the entropy decomposition:

$$H(P) = \sum_C H(P_C) - \sum_S H(P_S) \tag{2.133}$$

Using the moment-parameterization η of a graphical model, we write $H(\eta)$ for the global entropy and write $H_C(\eta_{[C]})$ for the marginal entropy on a clique (or separator). Here, $\eta_{[C]}$ denotes the subset of moment parameters associated with features defined on x_C

[22] We say that the marginal specification is consistent if any two distributions P_A and P_B that share variables $S = A \cap B$ are consistent with respect to these variables, that is, they have the same marginal distribution $P_S(x_S) \triangleq \sum_{x_{A \setminus B}} P_A(x_S, x_{A \setminus B}) = \sum_{x_{B \setminus A}} P_B(x_S, x_{B \setminus A})$. In junction trees, owing to the running intersection property, it is actually sufficient to check this condition only on the edges of the junction tree. However, in the case of learning (or information projection), if the marginals are all computed from a given distribution (e.g., the sample distribution $\tilde{P}$ or some other higher-order model that we would project to $\mathcal{G}$), then these marginals are, by definition, consistent and we do not have to worry about checking for consistency.

[23] Note that because we are using an over-parameterized representation here, the "projection" is not really unique, there are many equivalent reparameterizations of the model and many different projection formulas one could give that all project to different points within the same subspace of equivalent reparameterizations.

or any subset of these variables. In the Boltzmann model we define $\eta_{[C]} = \{\eta_E, E \subseteq C\}$ where $\eta_E = \mathbb{E}\{\phi_E(x)\} = \mathbb{E}\{\prod_{v \in E} x_v\}$. The notation $\eta_{[C]}$ is used to avoid confusion with η_C, which represents a *single* moment-parameter of the Boltzmann model. Thus, in the Boltzmann model, $[C]$ represents the set of all subsets of C. Now, we have the following formula for $H(\eta)$:

$$H(\eta) = \sum_C H_C(\eta_{[C]}) - \sum_S H_S(\eta_{[S]}) \tag{2.134}$$

By conjugate duality, we recall that the gradient of entropy is equal to (minus) the inverse moment map, that is, $\nabla H(\eta) = -\Lambda^{-1}(\theta)$. Using this relation, and the above decomposition, we obtain the projection formula:

$$\theta = \Lambda^{-1}(\eta) = \sum_C \Lambda_C^{-1}(\eta_{[C]}) - \sum_S \Lambda_S^{-1}(\eta_{[S]}) \tag{2.135}$$

where $\Lambda_C^{-1}(\eta_{[C]}) = -\nabla H_C(\eta_{[C]})$ denotes the (tractable) mapping between the η and θ representations within a fully-connected subset of $|C|$ nodes. In the Boltzmann model, one may use a recursive Möbius transform [30, 121, 123] to compute the mapping Λ_C^{-1} within each clique (or separator) with complexity $\mathcal{O}(w2^w)$ where w is the clique size. We describe this transform and its applications to Boltzmann machines in Appendix C. These calculations are generally linear in the number of nodes but exponential in the tree-width of the graph.

In the Gaussian model, this projection step reduces to the following calculations. Given the mean vector $\hat{x}$ and partial specification of the covariance matrix K, over the graph $\mathcal{G}$, we compute the information form (h, J) as follows. Let K_A denote the principle submatrix of K corresponding to nodes $A \subset V$. The information matrix is formed as

$$J = \sum_C [(K_C)^{-1}]_{V \times V} - \sum_S [(K_S)^{-1}]_{V \times V} \tag{2.136}$$

where $[\hat{J}_C]_{V \times V}$ denotes zero-padding to an $n \times n$ matrix (indexed by V) having $\hat{J}_C = (K_C)^{-1}$ as the appropriate principle submatrix corresponding to indices $C \subset V$. Note that the result is a sparse J matrix that respects the Markov structure specified by $\mathcal{G}$. The potential vector h vector is determined by:

$$h = \sum_C [(K_C)^{-1} \hat{x}_C]_V - \sum_S [(K_S)^{-1} \hat{x}_S]_V \tag{2.137}$$

where $[\hat{h}_C]_V$ denotes zero-padding to an n-vector having $\hat{h}_C = (K_C)^{-1} \hat{x}_C$ as the appropriate subvector. Note that h can also be obtained by sparse matrix multiplication: $h = J\hat{x}$. The computational complexity of the Gaussian projection is linear in the number of nodes and cubic in the tree-width of the graph.

Unfortunately, these projection methods only work for chordal graphs. In practice, the graph $\mathcal{G}$ is often not chordal, and solution of the projection problem generally requires iterative methods to solve the convex optimization problem (2.127).

Iterative Scaling Algorithm

We now discuss an iterative method that can be used to solve for the projection to a thin, non-chordal graph. We still need that the graph to which we wish to project is thin because this method requires that we perform inference with respect to the graphical model, which is only tractable in thin graphs.

The *iterative scaling* algorithm [62, 63], also known as *iterative proportional fitting* [114,186], is a simple procedure that iteratively cycles over the (generalized) edges $E \in \mathcal{G}$ (e.g., these may be taken to be the cliques of a pairwise graph) of our graphical model and, at each step, modifies the edge potential $\theta_E(x_E)$ to enforce the constraint that the corresponding marginal distribution $P(x_E)$ should be equal to the sample distribution $\tilde{P}(x_E)$. In order to impose this constraint, we must first compute the current marginal distribution $P(x_E)$ (e.g., using recursive inference methods defined over a junction tree of the graph, such as discussed in Section 2.4.1). Thus, the method is only tractable for thin graphs (where exact inference is tractable). Then, the probability distribution $P(x)$ is modified by scaling it by the ratio of the sample marginal $\tilde{P}(x_E)$ to the current marginal $P(x_E)$:

$$P'(x) = P(x) \times \frac{\tilde{P}(x_E)}{P(x_E)} \tag{2.138}$$

Here, we use P' to denote the value of the distribution immediately after this correction step. This does indeed have the desired effect, such that the new marginal $P'(x_E)$ is now equal to the sample marginal $\tilde{P}(x_E)$, as shown by:

$$P'(x_E) = \sum_{x_{V \setminus E}} P'(x_E, x_{V \setminus E}) = \cancel{\left(\sum_{x_{V \setminus E}} P(x_E, x_{V \setminus E}) \right)} \times \frac{\tilde{P}(x_E)}{\cancel{P(x_E)}} = \tilde{P}(x_E)$$

Using an over-parameterized representation, this modification of P is realized simply by absorbing the correction factor into the edge factor $\psi_E(x_E)$:

$$\psi'_E(x_E) = \psi_E(x_E) \times \frac{\tilde{P}(x_E)}{P(x_E)}, \tag{2.139}$$

or, equivalently, by correcting the edge potential:

$$\theta'_E(x_E) = \theta_E(x_E) + \left(\log \tilde{P}_E(x_E) - \log P_E(x_E) \right). \tag{2.140}$$

A similar method works in minimal representations of the model. Then, at each step, we modify the appropriate subset of model parameters $\theta_{[E]}$ according to:

$$\theta'_{[E]} = \theta_{[E]} + \left(\Lambda_E^{-1}(\tilde{\eta}_{[E]}) - \Lambda_E^{-1}(\eta_{[E]}) \right) \tag{2.141}$$

For instance, in the Boltzmann model the mapping Λ_E^{-1} can be computed using the Möbius transform described in Appendix C. In the Gaussian model, these corrections are

performed within the information form and reduce to local matrix inverse calculations [199]:

$$J'_{E,E} = J_{E,E} + [(\tilde{K}_E)^{-1} - (K_E)^{-1}]_{V \times V} \quad (2.142)$$

$$h'_E = h_E + [(\tilde{K}_E)^{-1}\tilde{x}_E - (K_E)^{-1}\hat{x}_E]_V \quad (2.143)$$

Here, $(\hat{x}_E, K_E)$ denotes the current marginal moments, computed for (h, J), and $(\tilde{x}_E, \tilde{K}_E)$ denote sample statistics.

The important thing to note about this update procedure is that, immediately after the update on edge $E \in \mathcal{G}$, it holds that the *new* marginal on edge E is equal to the specified marginal. However, as we proceed to correct the other edges of the graph, this has the effect of also changing the marginals on previously-corrected edges, thereby spoiling the moment-matching condition for those edges.[24] Hence, it is generally necessary to *repeatedly* cycle over the all the edges of the graph until we reach a fixed-point of the algorithm for which all the marginal-matching constraints are simultaneously satisfied. In practice, we terminate this iterative procedure once the marginal discrepancies are found to be sufficiently small.

Block-Coordinate Descent There are two complementary interpretations of the iterative scaling algorithm. The first (simpler) interpretation is that it performs *block coordinate descent*, a standard method of optimization theory [24], with respect to the convex optimization problem (2.127). This is, at each step, the subset of model parameters $\theta_{[E]}$ are jointly updated so as to minimize the objective $\mathcal{F}(\theta) = \Phi(\theta) - \tilde{\eta}^T\theta$ with respect to just the variables $\theta_{[E]}$ (with the other elements of θ being held fixed at their previous values). Computing the gradient with respect to $\theta_{[E]}$, and setting this to zero, we obtain the optimality condition: $(\Lambda^{-1}(\theta))_{[E]} = \tilde{\eta}_{[E]}$, which is satisfied immediately after the iterative scaling correction on edge E. Note also that the iterative scaling update on edge E does not modify any parameters except for $\theta_{[E]}$. Therefore, it is equivalent to one step of block-coordinate descent. For a convex and differentiable objective function, such as $\mathcal{F}(\theta)$, coordinate descent is guaranteed to converge to the global minima [24]. Hence, the iterative-scaling method is guaranteed to converge to the global minimum of $\mathcal{F}(\theta)$.

Csiszar-Bregman Iterative Projection Method A geometric interpretation of the iterative scaling algorihtm has been provided by Csiszar using ideas of information geometry and information projection [61,62]. In fact, owing to the Bregman distance interpretation of relative entropy, this interpretation actually shows that that iterative scaling algorithm is a special case of the general method due to Bregman for projection onto convex sets [39].

[24]The only exception to this rule seems to be the case where $\mathcal{G}$ represents the set of cliques of a chordal graph (e.g., the pairwise edges of a Markov chain or tree). Then, one can show that by performing the iterative scaling updates in an appropriate order, the procedure actually yields the exact information projection in a finite number of steps (by one pass through the junction tree of the graph). This is consistent with the closed-form formula for projection onto a chordal graph using the junction tree factorization.

In Section 2.3.3 we saw that associated with every m-projection problem (such as maximum-likelihood) there is an equivalent class of dual e-projection problems that lead to the same solution. In the context of maximum-likelihood learning, this means that the maximum-likelihood projection $\theta^* \in \Theta(\mathcal{G})$, of $\tilde{P}$ to our family of graphical models defined on $\mathcal{G}$, can be equivalently described as follows: given *any* point $\theta^{(0)} \in \Theta(\mathcal{G})$, the maximum-likelihood parameter θ^* is equivalent to the e-projection of $\theta^{(0)}$ to the m-flat submanifold $\mathcal{M} = \cap_{E \in \mathcal{G}} \mathcal{M}_E$ defined as the intersection of m-flat submanifolds $\mathcal{M}_E = \{\eta \mid \eta_{[E]} = \tilde{\eta}_{[E]}\}$. In other words, each submanifold $\mathcal{M}_E$ represents the set of all probability distributions that satisfy a single marginal-matching condition $\eta_{[E]} = \tilde{\eta}_{[E]}$. Then, the intersection of all of these manifolds defines the set of probability distributions which have the same marginal moments as $\tilde{P}$ with respect to the exponential family defined on $\mathcal{G}$. Then, Proposition 2.3.2 asserts that the m-projection of $\tilde{P}$ to $\mathcal{G}$ is equivalent to the e-projection of $\theta^{(0)} \in \Theta(\mathcal{G})$ to $\mathcal{M}$.

$$\eta^* = \arg\min_{\eta \in \mathcal{M}} d(\eta, \theta_0) \tag{2.144}$$

Csiszar realized that, rather than directly projecting onto $\mathcal{M}$, one may instead use an iterative method that involves projecting onto these sets $\mathcal{M}_E$ *sequentially.* That is, we generate a sequence of solutions $\eta^{(k)}$, where, at each step we take the previous solution $\eta = \eta^{(k)}$ and project it onto one of the the sets $\mathcal{M}_E$ to obtain the next iterate $\eta' = \eta^{(k+1)}$ such that

$$\eta' = \arg\min_{\mu \in \mathcal{M}_E} d(\mu, \theta) \tag{2.145}$$

where θ are the parameters of η. Introducing Lagrange multipliers to enforce the linear moment constraints, one finds that the optimal solution to this e-projection problem must be of the form

$$P'(x) = P(x) \exp\{\lambda^T \phi_{[E]}(x)\} \tag{2.146}$$

where the multipliers λ are chosen to satisfy the moments constraints $\eta' \in \mathcal{M}_E$. Comparing this to the iterative scaling algorithm, we identify the optimal choice of Lagrange multipliers λ with the parameter correction in (2.141), that is:

$$\lambda = \theta'_{[E]} - \theta_{[E]} = \Lambda^{-1}(\tilde{\eta}_{[E]}) - \Lambda^{-1}(\eta_{[E]})$$

Thus, each step of the iterative scaling method can also be viewed as performing the e-projection of the previous solution to the next m-flat manifold.

■ 2.7.2 Structure Learning

We now briefly consider some approaches to also learn the graph structure $\mathcal{G}$ from sample data.

Chow-Liu Trees and Thin Chordal Graphs

Using the fact that there is a closed form solution for projection to trees and thin chordal graphs (junction trees), it is then natural to seek the best tree (or best tree-width k

chordal graph) to approximate a given distribution P (e.g., the sample distribution $\tilde{P}$). Our presentation closely follows [56, 131].

In chordal graphs, the projection formula we described earlier (in terms of cliques and separators over a junction tree of the graph) can equivalently be described by the following recursively defined set of potential functions defined on *all* cliques of the graph:

$$\theta_C(x_C) \triangleq \log P(x_C) - \sum_{C' \subsetneq C} \theta_{C'}(x_{C'}) \tag{2.147}$$

$$= \sum_{C' \subseteq C} (-1)^{|C \setminus C'|} \log P(x_{C'}) \tag{2.148}$$

In terms of these potentials, the projection to *any chordal graph* $\mathcal{G}$ is given by:

$$P_{\mathcal{G}}(x) = \exp\left\{\sum_{C \in \mathcal{G}} \theta_C(x_C)\right\} \tag{2.149}$$

This projection formula has the advantage that it does not explicitly refer to a junction tree of the graph, treating all cliques of the chordal graph uniformly.[25] In other words, the *same* potential $\theta_C(x_C)$ may be used in *any* chordal graph $\mathcal{G}$ containing C as a clique. From this latter chordal projection formula, we can show the following clique-wise decomposition of the relative entropy $D(P, P_{\mathcal{G}})$ between the distribution P and its projection $P_{\mathcal{G}}$:

$$D(P, P_{\mathcal{G}}) \triangleq \mathbb{E}_P\left\{\log \frac{P(x)}{P_{\mathcal{G}}(x)}\right\} = -(H(P) + \sum_{C \in \mathcal{G}} \mathbb{E}_P\{\theta_C(x_C)\}) \tag{2.150}$$

Thus, selecting $\mathcal{G}$ to minimize relative entropy (over some class of chordal graphs) is equivalent to maximizing $w(\mathcal{G}) \triangleq \sum_{C \in \mathcal{G}} w_C$ with edge weights defined by:

$$w_C \triangleq \mathbb{E}_P\{\theta_C(x_C)\} = -\sum_{C' \subseteq C} (-1)^{|C \setminus C'|} H(P_{C'}) \tag{2.151}$$

We also note that the weight $w(\mathcal{G})$ is equal to the negative entropy $-H(P_{\mathcal{G}})$ of the projection to $\mathcal{G}$. This is analogous to the entropy decomposition given earlier for junction trees. Thus, maximizing $w(\mathcal{G})$ is equivalent to selecting $\mathcal{G}$ to minimize $H(P_{\mathcal{G}})$. Intuitively, this comes about as we seek the graph that retains the most information (the least uncertainty), thereby minimizing the information loss $D(P, P_{\mathcal{G}}) = H(P_{\mathcal{G}}) - H(P)$ relative to P. Then, we are left with the problem of finding the maximum-weight (generalized) graph over some class of chordal graphs (e.g., the class of tree-width k graphs). Note that, if we only consider tree-width k graphs, then we only need to compute the

[25] On the other hand, the projection formula we gave earlier is much more efficient to compute, as it only requires computations on each *maximal* clique and each separator (rather than on all cliques).

edge weights w_C on sets C up to size $|C| < k+1$. In the case of trees ($k = 1$), this is just the max-weight spanning tree problem, which can be efficiently solved using a greedy algorithm [141]. Unfortunately, for $k \geq 2$, selecting the max-weight chordal graph among the set of tree-width k graphs is NP-hard [131], and a number of approximation strategies have therefore been developed [8, 49, 69, 131, 167].

The approach reviewed so far in this section is based on the philosophy that, because exact inference is only tractable in thin graphs, one should only try to learn thin graphical models (to ensure that the learned model is tractable). While this may well be a reasonable approach in some contexts, there are two plausible arguments against this point of view:

1. The processes that we actually wish to model in practice are often not well-approximated by any thin graphical model. Hence, if the goal of learning is to actually build realistic models, learning thin models is probably not a good idea in general.

2. While it is true that *exact* inference is only tractable in thin graphs, there is much evidence that *approximate* inference is tractable in a much broader class of model, and the difficulty of approximate inference does not seem to be correlated with tree-width.

This suggests developing principled learning methods based on other measures of model complexity (besides tree-width) to help regularize our choice of graphical model.

Learning Sparse Graphical Models

In classical approaches to model-order selection, such as Akaike's information criteria [2], one seeks to strike a balance between providing a good fit to the data and having a low-order model (with a small number of model parameters). The purpose of having a low-order model is to avoid over-fitting the data. Essentially, this involves using a penalized likelihood function or, equivalently, minimizing a penalized divergence measure:

$$\mathcal{F}(\theta) = D(\tilde{P}, P_\theta) + \gamma \|\theta\|_0 \tag{2.152}$$

Here, θ denotes the vector of all possible model parameters in some high-dimensional exponential family model, $\|\theta\|_0$ denotes the ℓ_0-norm of the parameter vector (the number of non-zero parameters), P_θ is the exponential family distribution and $D(\tilde{P}, P_\theta)$ is its divergence relative to the distribution of sample data. The parameter $\gamma > 0$ controls how much importance we place on obtaining a low-order model relative to obtaining a model that provides a good fit to the sample data.

It is worth noting that, because the ℓ_0-norm does not depend on the absolute value of the non-zero parameters (it only depends on the number of non-zero parameters), the optimal solution to this problem always corresponds to the maximum-likelihood solution within the subspace of non-zero parameters. Akaike demonstrated that the effect of penalizing the model order is to reduce the expected value of the divergence

$D(P_{\text{true}}, P_\theta)$ relative to the true (unknown) distribution P_{true} that the samples are drawn from. His analysis suggest that the best choice of γ is approximately $\frac{1}{2N}$ where N is the number of samples. In exponential family graphical models, this complexity measure $\|\theta\|_0$ seems very appropriate as it is also correlated with the number of edges of the graph $\mathcal{G}$. For example, in the Gaussian or Boltzmann models, requiring that the model is Markov on a graph $\mathcal{G}$ is equivalent to setting $\theta_E = 0$ if $E \notin \mathcal{G}$. Because the model complexity $\|\theta\|_0$ is non-convex, it is not generally tractable to (provably) find the best graphical model under this criterion. Nonetheless, a number of greedy heuristic methods have been proposed, which incrementally add or remove edges in the graph to search for a local minimum of this objective [67, 126, 177, 199].

Recently, another approach to model selection in graphical models has been proposed, which may be regarded as the convex relaxation of the above objective, using the (convex) ℓ_1-norm in place of the (non-convex) ℓ_0-norm to measure model complexity [9, 147]. That is, we select the model parameters θ to minimize

$$\mathcal{F}(\theta) = D(\tilde{P}, P_\theta) + \gamma\|\theta\|_1, \tag{2.153}$$

which is now a convex optimization problem that is tractable to solve. Using the ℓ_1-norm $\|\theta\|_1 = \sum_k |\theta_k|$ tends to select sparse solutions when doing so does not drastically effect the divergence. Note that using the ℓ_1-norm does not lead to an exact maximum-likelihood model in some subspace, as even the non-zero parameters tend to be deflected towards zero.

This approach also has a dual interpretation, as a relaxed version of the maximum entropy principle [72]:

$$\begin{array}{ll} \text{maximize} & H(\eta) \\ \text{subject to} & \|\eta - \tilde{\eta}\|_\infty \leq \gamma \end{array} \tag{2.154}$$

This also is a convex optimization problem, now formulated in terms of the moment parameters (marginal distributions) of the graphical model. In comparison to the traditional maximum entropy principle, with equality constraints $\eta = \tilde{\eta}$, this formulation allows for some error in specification of the moments $\tilde{\eta}$, only requiring that the moments of the model η are close to the sample moments. Those inequality constraints that are active (satisfied with equality) in the optimal solution η^* of this problem then determine the sparsity of the solution in the θ^* parameters (and, hence, the graphical structure of the resulting model). In other words, the features associated with inactive constraints (that are strictly satisfied) are dropped from the model, with the corresponding elements in θ^* being zero.

Chapter 3

Lagrangian Relaxation for Discrete MRFs

■ 3.1 Introduction

In this chapter we develop a general Lagrangian relaxation (LR) approach to MAP estimation in MRFs with discrete variables. Our general strategy is based on reformulating an intractable estimation problem, defined on a graph having large tree-width, as an equivalent estimation problem defined on a larger graph of smaller tree-width but subject to additional complicating constraints. This is done by creating multiple copies of nodes and edges in the new graph and decomposing node and edge potentials of the original graph between these copies. The simplest examples of this approach involve block decompositions, in which we break the graph up into its individual edges or small subgraphs. However, it can be more efficient computationally to allow larger subgraphs that are thin, such as spanning trees similar to the tree-reweighted approach [211]. In any case, an equivalent MAP estimation problem is obtained by imposing constraints that all copies of a node or edge should be assigned consistently in the new MRF. Then, introducing Lagrange multipliers to relax these constraints, we obtain a tractable estimation problem defined in the relaxed MRF that is now defined on a thin graph. The value of this relaxed MAP problem provides an upper-bound on the value of the original MAP estimation problem. It then remains to minimize this upper-bound with respect to the Lagrange multipliers, which is equivalent to optimizing over all valid decompositions of the potentials among multiple copies of a node or edge. For some models, this minimization results in a consistent MAP estimate satisfying all of the relaxed constraints. Then, there is no duality gap (the primal and dual problems have the same value) and the correct MAP estimate is obtained.

In order to minimize the Lagrangian dual problem, we develop a statistical-physics-based approach, using a Gibbs distribution defined on the decomposed model, which becomes equivalent to Lagrangian relaxation in the limit as the temperature of the Gibbs distribution approaches zero. To solve this low-temperature relaxation we develop a simple iterative algorithm which involves marginal computations within each subgraph and exchanging messages between copies of a node or edge so as to enforce equality of their marginal distributions. This procedure has an intuitive geometric in-

terpretation as a special case of the iterative scaling method, traditionally used for fitting graphical models to data. In our formulation, each step of the procedure is interpreted as a minimum relative entropy projection to a linear subspace where all copies of a specific node or edge are required to have consistent marginal distributions. Alternatively, the procedure can also be described simply as a block coordinate descent method for minimizing the free energy of the Gibbs distribution over the subspace of valid potential decompositions. Using this optimization approach at each temperature, we then gradually reduce the temperature to approach the solution to the Lagrangian dual problem. This is shown to correspond to an interior point method for solving a relaxed version of the MAP problem based on the pseudo-marginal polytope, where the entropy of the decomposed model serves as a barrier function.

An added advantage of this method of gradually reducing the temperature is that it provides a simple way to obtain approximate solutions in cases where there is a duality gap. This method, which we refer to as *low-temperature estimation*, produces an estimate at each temperature by selecting each variable to maximize its approximate low-temperature marginal probability distribution. We compute such an estimate at each temperature and then select the best one over a range of temperatures. This can also serve to initialize a greedy algorithm, such as the *iterated conditional modes* (ICM) method [28]. If there is no duality gap, this method still recovers the correct MAP estimate at low enough temperatures. However, when there is a duality gap, the zero-temperature marginals exhibit ties (non-unique MAP estimates in the relaxed problem) so that some or all variables becomes ambiguous. The low-temperature estimation approach provides a way to "break" these ties. While the approach may seem a bit simplistic, we have found that it often produces surprisingly good results. In fact, in most of our simulation studies so far, based on randomly-generated spin-glass models, we have found that this method consistently provides optimal or near-optimal solutions even in examples that exhibit a large duality gap. We provide a geometric interpretation to help explain why this occurs.

We also consider methods to enhance the LR formulation in cases where a duality gap occurs. This involves detecting inconsistent subgraphs, where the optimality conditions arising in the optimal dual decomposition are incompatible, and including these subgraphs (or maximal cliques of triangulated versions of these subgraphs) in LR so as to eliminate these inconsistencies. This approach is closely related to cutting-plane methods used to solve linear programs with a large number of constraints [12, 13] and recent related work of Sontag *et al* [198]. However, in our approach, rather than introducing additional constraints, we instead introduce additional degrees of freedom in the dual formulation. Also, our approach is based on the characterization of strong duality as a satisfiability problem, which is to determine if the set of MAP solutions on each subgraph of the decomposition are globally consistent. We propose a simple version of this idea to identify inconsistent cycles in binary variable models, and to remove these inconsistencies by including additional blocks in the decomposition.

In the final section of the chapter, we perform a computational study of the per-

formance of LR for solving for the MAP estimate of ferromagnetic and frustrated Ising models on planar and non-planar 2D lattices.

■ 3.1.1 Road-Map of Various Related Problems

In our development and analysis we discuss a number of inter-related optimization problems, corresponding to different representations of the Lagrangian dual problem (and smoothed versions of these) arising in our Lagrangian relaxation method and related linear-programming relaxations of MAP estimation (and maximum-entropy regularized versions of these). The dual problem may be formulated in terms of either a set of Lagrange multipliers or as an optimization over the set of valid potential decompositions arising in one of our graphical decomposition methods. The LP relaxation method is formulated in terms of the moment parameterization of the graphical model (corresponding to its marginal specification) taken over the pseudo-marginal polytope of the graphical model. In our analysis, we relate our dual methods for MAP estimation to the LP relaxation method by demonstrating that the LP relaxation method is recovered by taking the "dual of the dual", that is, by again taking the Lagrangian dual of our dual problem.[1] However, our algorithms ultimately focus on solving these problems in the dual domain, that is, as either an optimization over Lagrange multipliers or valid potential decompositions.

To help the reader navigate the chapter, we provide a road-map of these various problems, and the connections between these, in Figure 3.1. Here is a brief synopsis:

1. **MAP-$\mathcal{G}$ and LP-$\mathcal{M}$** (Section 3.1.2): This is the MAP estimation problem that we wish to solve. It is formulated with respect to a graphical model defined on a (generalized) graph $\mathcal{G}$. (MAP-$\mathcal{G}$) refers to the problem of explicitly maximizing $f(x) = \theta^T \phi(x)$ over all $x \in \mathbb{X}^n$. (LP-$\mathcal{M}$) refers to the equivalent formulation of the MAP estimation problem as solving a linear program (LP) over the marginal polytope $\mathcal{M}(\mathcal{G})$, which is the convex hull of the set $\{\phi(x), x \in \mathbb{X}^n\}$.

2. **LP-$\hat{\mathcal{M}}$** (Section 3.2.5): This denotes the LP relaxation of MAP estimation, obtained by replacing the (intractable) marginal polytope $\mathcal{M}(\mathcal{G})$ by the pseudo-marginal polytope $\hat{\mathcal{M}}(\mathcal{G})$, which provides an outer-bound on $\mathcal{M}(\mathcal{G})$. The LP relaxation method was reviewed previously in Section 2.5.2, and is also discussed in Section 3.2.5.

3. **MAP-$\mathcal{G}^\dagger$** (Section 3.2.3): To apply the Lagrangian relaxation method, we first reformulate (MAP-$\mathcal{G}$) as a constrained MAP estimation problem on a thin graph $\mathcal{G}^\dagger$. The manner in which $\mathcal{G}^\dagger$ is defined is discussed in the beginning of Section 3.2. The problem (MAP-$\mathcal{G}^\dagger$) is defined in Section 3.2.3. Although defined on larger set of problem variables, this problem is subject to constraints that ensure it is still equivalent to (MAP-$\mathcal{G}$).

[1] We note that the method of considering the "dual of the dual" is a standard approach to derive convex relaxations of intractable combinatorial optimization problems [25, 26].

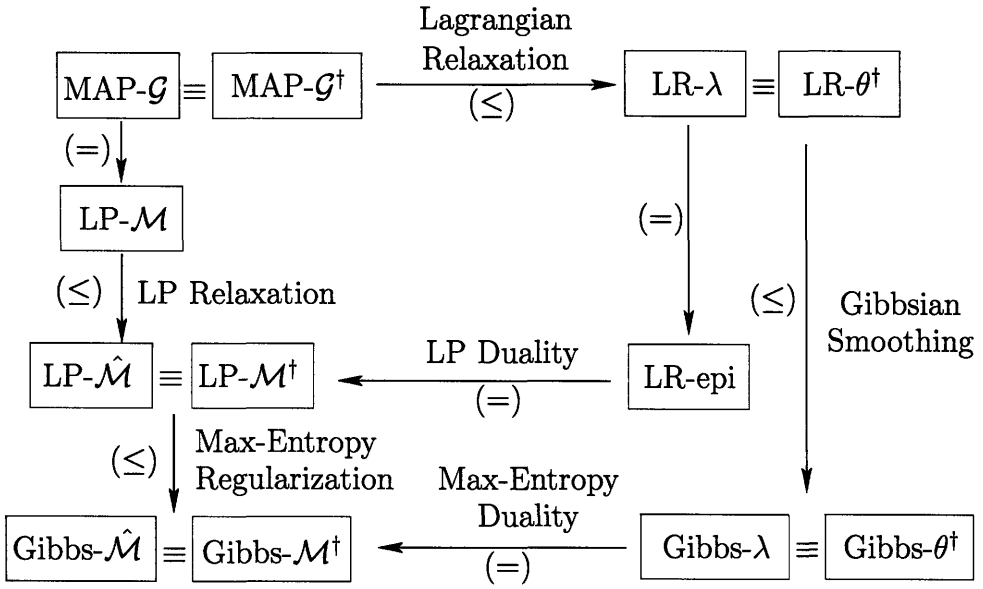

Figure 3.1. Road-map of various related problems that are discussed in the chapter.

4. **LR-λ and LR-$\theta^\dagger$** (Section 3.2.3): Then, we introduce Lagrange multipliers λ to relax the constraints of (MAP-$\mathcal{G}^\dagger$), which leads to the Lagrangian dual problem (LR-λ). This is shown to be equivalent to an optimization over all valid decompositions $\theta^\dagger$ of the original graphical model with respect to $\mathcal{G}^\dagger$. We denote this latter formulation of the problem as (LR-$\theta^\dagger$).

5. **LR-epi and LP-$\hat{\mathcal{M}}^\dagger$** (Section 3.2.5): To relate these (LR) problems to (LP-$\hat{\mathcal{M}}$), we reformulate the problem (LR-λ) as an LP over the epigraph of the convex, piece-wise linear dual function being minimized in (LR-λ). This problem is denoted (LR-epi). Then, we show that the LP dual of (LR-epi) is an LP over the marginal polytope of the decomposed graphical model defined on $\mathcal{G}^\dagger$. We denote this dual LP by (LP-$\mathcal{M}^\dagger$) and show that is it actually equivalent to the LP-relaxation (LP-$\hat{\mathcal{M}}$). Thus, this establishes a connection between our work, which focuses on dual algorithms for solving (LR), and methods aimed at directly solving (LP-$\hat{\mathcal{M}}$).

6. **Gibbs-λ and Gibbs-$\theta^\dagger$** (Section 3.3.1): In our approach to solving the (LR) problems, we actually solve smoothed version of these, using a Gibbsian smoothing method inspired by statistical physics. This involves using the free energy (log-partition function scaled by temperature) of a Gibbs distribution based on the decomposed potential specification over $\mathcal{G}^\dagger$ (at a specified temperature) to provide a smooth, convex, upper-bound approximation to the non-smooth Lagrangian dual function. We denote the smoothed versions of the (LR) problems

as either (Gibbs-λ) or (Gibbs-$\theta^\dagger$), depending on which parameterization is used. We solve the dual problem by gradually lowering the temperature (annealing) and re-optimizing the potential decomposition at each temperature so as to minimize the free energy. In the limit of zero-temperature, this recovers the correct solution of the dual problem. This may also be viewed as an interior-point method for solving (LR-epi).

7. **Gibbs-$\mathcal{M}^\dagger$ and Gibbs-$\hat{\mathcal{M}}$** (Section 3.3.2): Finally, we show that using the Gibbsian smoothing method in the dual problem is equivalent to using a maximum-entropy regularization method in the LP relaxation method. This involves using maximum-entropy duality to show that the Lagrangian dual of (Gibbs-$\theta^\dagger$) is a maximum-entropy regularized version of the (LP-$\mathcal{M}^\dagger$), which we denote by (Gibbs-$\mathcal{M}^\dagger$). This problem then reduces to (Gibbs-$\hat{\mathcal{M}}$), which uses a block-wise entropy function to regularize (LP-$\hat{\mathcal{M}}$). In the temperature annealing method, this essentially amounts to using the block-wise entropy function as a *barrier function* to provide an interior-point method for solving (LP-$\hat{\mathcal{M}}$).

The reader may wish to refer back to this road-map as they encounter each of these problems, to recall the progress made so far and to understand how these various pieces fit together.

■ 3.1.2 MAP Estimation

We briefly restate the MAP problem in graphical models (see Chapter 2). We are given a graph-structured objective function $f(x)$, also called the energy, based on a (generalized) graph $\mathcal{G}$:

$$f(x) = \sum_{E \in \mathcal{G}} f_E(x_E) \tag{3.1}$$

The problem variables $x = (x_1, \ldots, x_n)$ are identified with the vertices of $\mathcal{G}$ and the potentials $f_E(x_E)$ are defined over the edges of $\mathcal{G}$. In this chapter we assume that each variable $x_v \in \mathbb{X}$ is a discrete variable with a small number of possible states, e.g., $\mathbb{X} = \{0, 1\}$ or $\mathbb{X} = \{-1, +1\}$ for binary-variable models. This defines a Gibbs distribution $P(x) \propto \exp\{f(x)\}$. Then, we solve for the *maximum a posteriori* (MAP) estimate $\hat{x}$ to maximize $P(x)$ or, equivalently, to maximize the objective:

(MAP-$\mathcal{G}$)	maximize	$f(x) = \sum_{E \in \mathcal{G}} f_E(x_E)$
	subject to	$x \in \mathbb{X}^n$

Using an exponential family representation of $P(x) \propto \exp\{\theta^T \phi(x)\}$ (Section 2.3) with binary-valued features (e.g. the Ising or Boltzmann models or the over-parameterized representation of general discrete models using indicator functions as features), this may be equivalently stated as a linear program (LP) over the marginal polytope $\mathcal{M}(\mathcal{G})$

of the graphical model:

$$\boxed{\begin{array}{ll} \text{(LP-}\mathcal{M}\text{)} & \begin{array}{ll} \text{maximize} & \theta^T \eta \\ \text{subject to} & \eta \in \mathcal{M}(\mathcal{G}) \end{array} \end{array}}$$

Because the marginal polytope is defined as the convex hull of the point set $\{\phi(x), x \in \mathbb{X}^n\}$, we see that (with binary-valued features) this is equivalent to (MAP-$\mathcal{G}$). However, because this corresponds to an integer programming problem, it is NP-hard to solve in general. The problem (LP-$\mathcal{M}$) is intractable owing to the marginal polytope having exponentially many faces.

■ 3.2 Graphical Decomposition Methods

In this section, we develop the main ideas concerning the formulation and interpretation of the Lagrangian relaxation approach to MAP estimation. We begin by describing a number of ways that an intractable graphical model can be relaxed to a tractable graphical model or decomposed into a set of separate tractable graphical models. We generally consider methods that involve *duplicating* variables to define this relaxed graphical model. Nodes and edges of the graph must also be duplicated, so as to allow potentials of the original model to be mapped to corresponding potentials in the relaxed model. In this decomposition step, every node and edge of $\mathcal{G}$ must be represented at least once in the graph $\mathcal{G}^\dagger$ that describes the relaxed or decomposed graphical model.

We describe two types of graphical decomposition methods: *block decompositions* and *subgraph decompositions*. In fact, the block decomposition method is really a special case of the subgraph decomposition method. However, block decompositions are somewhat simpler to describe and therefore serve as a useful pedagogical tool to convey the essential mathematical concepts. Also, we show that block decompositions are just as powerful as the class of subgraph decompositions insofar as, for every tractable subgraph decomposition (using thin graphs) there is a corresponding tractable block decomposition (using small blocks) that is just as good as the subgraph decomposition in regard to the MAP problems that it can solve without a duality gap (Corollary 3.2.2, Section 3.2.6). For these reasons, we briefly introduce the general subgraph decomposition method in Section 3.2.2, but then focus our mathematical development on the special case of block decompositions for the remainder of the chapter. However, the block decomposition method is less efficient computationally than the subgraph decomposition method, because block decompositions introduce many more degrees of freedom in the dual problem that must be solved. For this reason, we also provide additional details of the subgraph decomposition approach in Appendix A. The development of this more general method is almost identical to that for block decompositions, with some notational complications and a few additional caveats that we explain in the appendix.

■ 3.2.1 Block Decompositions

We begin by describing the simplest case of *block decompositions.* In this approach, the objective function is decomposed as a sum of local potential functions defined over the edges of a generalized graph $\mathcal{G}$ (these edges are the "blocks" of the decomposition):

$$f(x) = \sum_{E \in \mathcal{G}} f_E(x_E) \tag{3.2}$$

It is important to note that the manner in which the objective is split among edges of $\mathcal{G}$ is not unique. Given any two edges $A, B \in \mathcal{G}$ which share variables corresponding to $S = A \cap B$, we could add an arbitrary function $\lambda(x_S)$ to one potential and subtract it from the other to obtain another valid decomposition of $f(x)$. This degree of freedom is critical in our approach.

The starting point for our development is the simple observation that if we now *independently* maximize each block separately we obtain an *upper-bound* on the value of the MAP problem:

$$f^* \triangleq \max_x \left\{ \sum_{E \in \mathcal{G}} f_E(x_E) \right\} \leq \sum_{E \in \mathcal{G}} \max_{x_E} f_E(x_E) \tag{3.3}$$

To formalize this a bit, let us define a set of auxiliary variables x^E on each edge of $\mathcal{G}$ and denote the complete set of these auxiliary variables by $x^\dagger = (x^E, E \in \mathcal{G})$. Then, each valid decomposition of f over $\mathcal{G}$ defines an auxiliary objective function:

$$f^\dagger(x^\dagger) = \sum_{E \in \mathcal{G}} f_E(x^E) \tag{3.4}$$

This function may also be regarded as defining the energy of a *decomposed graphical model* based on an auxiliary graph $\mathcal{G}^\dagger$, in which each edge $E \in \mathcal{G}$ maps to a separate connected component within $\mathcal{G}^\dagger$. Thus, this auxiliary graph has an expanded vertex set $V^\dagger$ containing multiple duplicates of each vertex $v \in V$, one duplicate for each edge $E \in \mathcal{G}$ containing that vertex. This extended vertex set then represents the auxiliary variables $x^\dagger = (x^E, E \in \mathcal{G}) \equiv (x^\dagger_v, v \in V^\dagger)$.

This idea is shown in the simplest case of a pairwise graph in Figure 3.2(a). On the left, we circle the set of "blocks", which in this case are just the pairwise edges of the graph, and on the right we show the corresponding graphical decomposition $\mathcal{G}^\dagger$. We also call this a *pairwise decomposition.*

More generally, block decompositions are not restricted to only use the original set of edges of the graph. Instead, we may redefine $\mathcal{G}$ to be a set of larger blocks, such that every edge of the original graphical model is covered by some block in $\mathcal{G}$.[2] This method

[2]To avoid introducing further notation, we assume that $\mathcal{G}$ has already been redefined so that it edges represent the blocks that we wish to use in the decomposition method and the potentials of the original graphical model have been absorbed into the block potentials of this new graph.

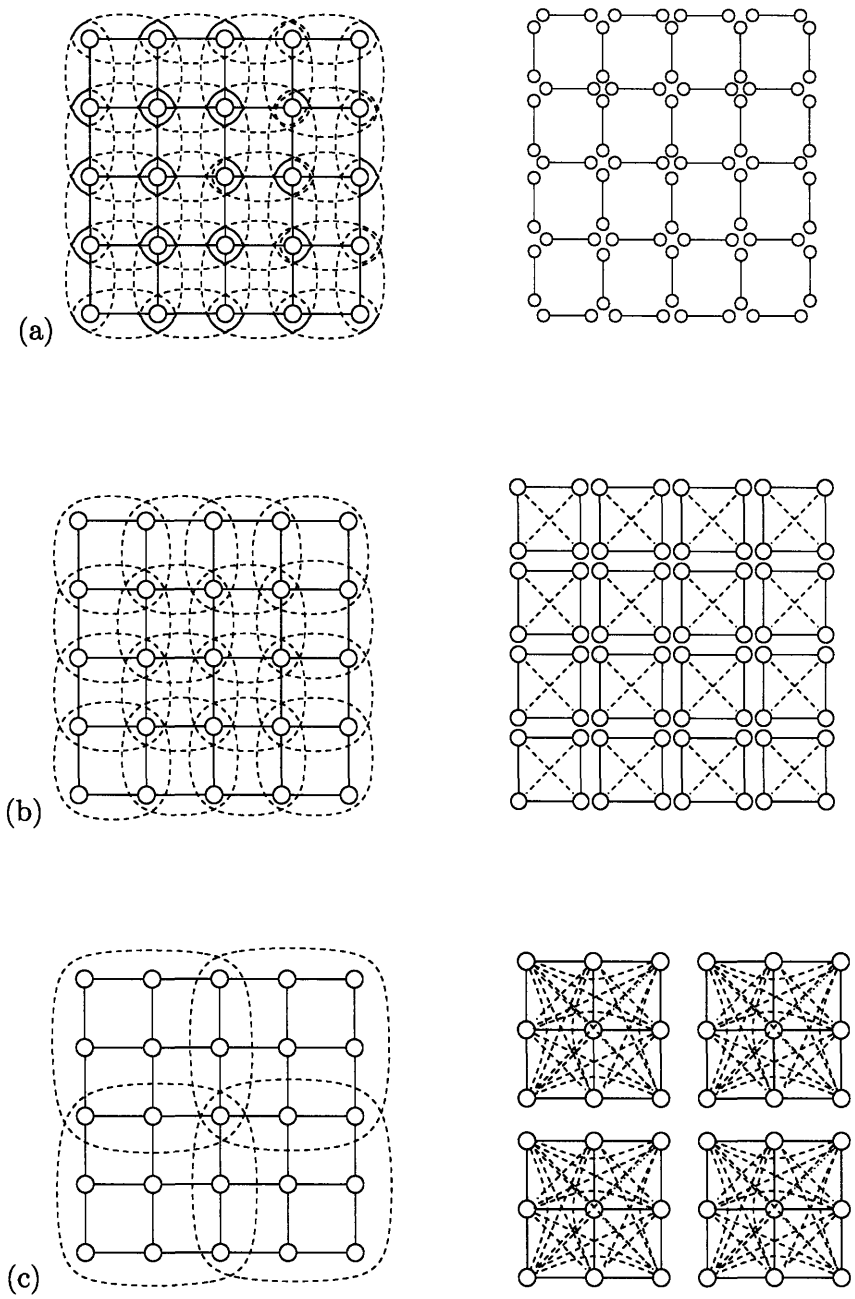

Figure 3.2. Illustration of several simple block decomposition of a 5×5 grid: (a) the pairwise decomposition using the edges of this pairwise graphical model, (b) decomposition into 2×2 blocks corresponding to the faces of the (planar) graph, (c) decomposition into 3×3 blocks. On the left in each figure, the dashed lines enclose the "blocks" of nodes of the graph. On the right we show the corresponding decomposed graph $\mathcal{G}^\dagger$ that is used to obtain tractable upper-bound on the value of the MAP problem on $\mathcal{G}$. In block decompositions, there is no assumption that each block must satisfy any Markov structure. Hence, each block is drawn as being a completely connected subgraph in $\mathcal{G}^\dagger$.

is illustrated in Figure 3.2(b) and (c), using either 2×2 or 3×3 blocks of nodes within the grid. Using larger blocks allows one to obtain tighter bounds of the value of the MAP problem. Note however, that in the block decomposition method, we presume no further structure within each block, that is, the block-wise potentials are not required to further decompose into edge-wise potentials defined over the corresponding subgraphs. This essentially means that we treat the graphical model as though each of those blocks were a clique of the graph to start with. Because of this, the block decomposition method will only be tractable if we keep the block sizes sufficiently small.

■ 3.2.2 Subgraph Decompositions

Next, we describe the *subgraph decomposition* method. Similar to the block decomposition method, this also involves splitting up the potentials of a graphical model defined on $\mathcal{G}$ among corresponding connected components within $\mathcal{G}^\dagger$. However, now each of the components corresponds to a subgraph of $\mathcal{G}$, rather than a completely connected "block". Let $\{\mathcal{G}^{(k)}\}$ denote this collection of (not necessarily disjoint) subgraphs, which are indexed by k. We require that every edge $\mathcal{G}$ is contained in at least one of these subgraphs: $\mathcal{G} \subset \cup_k \mathcal{G}^{(k)}$. Then, $\mathcal{G}^\dagger$ is defined by this set of subgraphs but where each subgraph now has it own set of nodes $V^{(k)}$ and corresponding variables $x^{(k)}$. Starting with the energy function (3.2) defined on $\mathcal{G}$, the potentials functions f_E defined on edges of $\mathcal{G}$ are then absorbed into (or split among) the corresponding components of $\mathcal{G}^\dagger$ that include the corresponding edges. We denote the decomposed potentials by $f_E^{(k)}(x_E)$, which are chosen subject to the constraints:

$$\sum_{k:E\in\mathcal{G}^{(k)}} f_E^{(k)}(x_E) = f_E(x_E) \quad \text{for all } x_E. \tag{3.5}$$

That is, if we re-sum potentials on all duplicates of edge E, in different components of $\mathcal{G}^\dagger$ that include that edge, we must recover the original potential f_E. This then defines an energy function on each subgraph:

$$f^{(k)}(x^{(k)}) = \sum_{E\in\mathcal{G}^{(k)}} f_E^{(k)}(x_E^{(k)}) \tag{3.6}$$

For valid subgraph decompositions, which satisfy (3.5), it holds that $f(x) = \sum_k f^{(k)}(x)$ for all x. Allowing each subgraph its own set of independent variables, we then obtain an auxiliary objective function $f^\dagger(x^\dagger)$ over $\mathcal{G}^\dagger$:

$$f^\dagger(x^\dagger) = \sum_k f^{(k)}(x^{(k)}) \tag{3.7}$$

As in the block-decomposition method, we obtain an upper-bound on the value of the MAP problem using any such subgraph decomposition of $f(x)$ defined on $\mathcal{G}$ with respect to this collection of subgraphs $\{\mathcal{G}^{(k)}\}$:

$$f^* \triangleq \max_x f(x) \leq \max_{x^\dagger} f^\dagger(x^\dagger) = \sum_k \max_{x^{(k)}} f^{(k)}(x^{(k)})$$

Moreover, if we require that each of the subgraphs $\mathcal{G}^{(k)}$ is thin, then it is tractable to compute this upper-bound as the maximum over each separate subgraph can then be efficiently computed using recursive inference methods.

We illustrate several subgraph decompositions of a pairwise model $f(x)$ defined on the 5×5 grid seen in Figure 3.3(a). Again, we may break the model up into small subgraphs, similar to the block decomposition method. For example, one may decompose the graph up into a set of short cycles as seen in (b). Such relaxations can often be improved using larger induced subgraphs such as seen in (c) where we have used 3×3 subgraphs such that the nodes along the boundary of each subgraph are duplicated between adjacent regions. In such cases, including additional edges along the boundary of these subgraphs, such as the dotted edges in (d), can enhance the relaxations that we consider. These method are similar to the block decomposition method, except that we do retain some sparse structure within each component, rather than simply treating the entire blocks of nodes as a completely connected subgraph.

The subgraph decomposition also allows us to use much larger subgraphs, provided that we require that each subgraph is thin, so as to be tractable by recursive inference methods. For example, as seen in (e) and (f), we may decompose the grid into a set of thin, horizontal strips such that adjacent strips share duplicated nodes along their boundary. Again, it can be useful to include extra edges in the overlap of these subgraphs as in (f) to enhance the relaxation, although this increases the width of the subgraph and affects the computational complexity of our methods. Another example of this method is to decompose the graphical model among a set of *spanning trees* as seen in (g). The tree-reweighted max-product (TRMP) method is based on a similar idea (using convex combinations of trees) [211].

In principle, one could also consider relaxations in which the resulting thin graph $\mathcal{G}^{\dagger}$ is connected. For example, by taking a spanning tree of the graph and then adding an extra leaf node for each missing edge we obtain the graph seen in (d). However, we should mention that, although our general formalism would allow such relaxations, the iterative scaling method we develop is substantially complicated in such cases, so we focus on solving decompositions that do not allow multiple copies of a variable within the same connected component of the relaxed problem. But we do point out that these types of relaxations could be handled within our basic framework, although different optimization methods may then be more appropriate to solve the resulting dual problem (for example, subgradient descent or steepest descent to minimize the free energy in our statistical-physics approach).

A Preliminary Example of Lagrangian Relaxation To briefly convey the basic idea, consider a simple pairwise Ising model defined on a 3-node cycle $\mathcal{G}$ represented in Fig. 3.4. To relax this model, we define a thin graph $\mathcal{G}^{\dagger}$ (the 4-node chain seen on the right in the figure)[3] where node 4 is a copy of node 1. Essentially, we have "broken"

[3]Note that this preliminary example is not a subgraph decomposition as it does not break $\mathcal{G}$ into a set of subgraphs. We use this example because it only introduces a single Lagrange multiplier in the dual problem and thus serves as a simple illustration of the Lagrangian relaxation method.

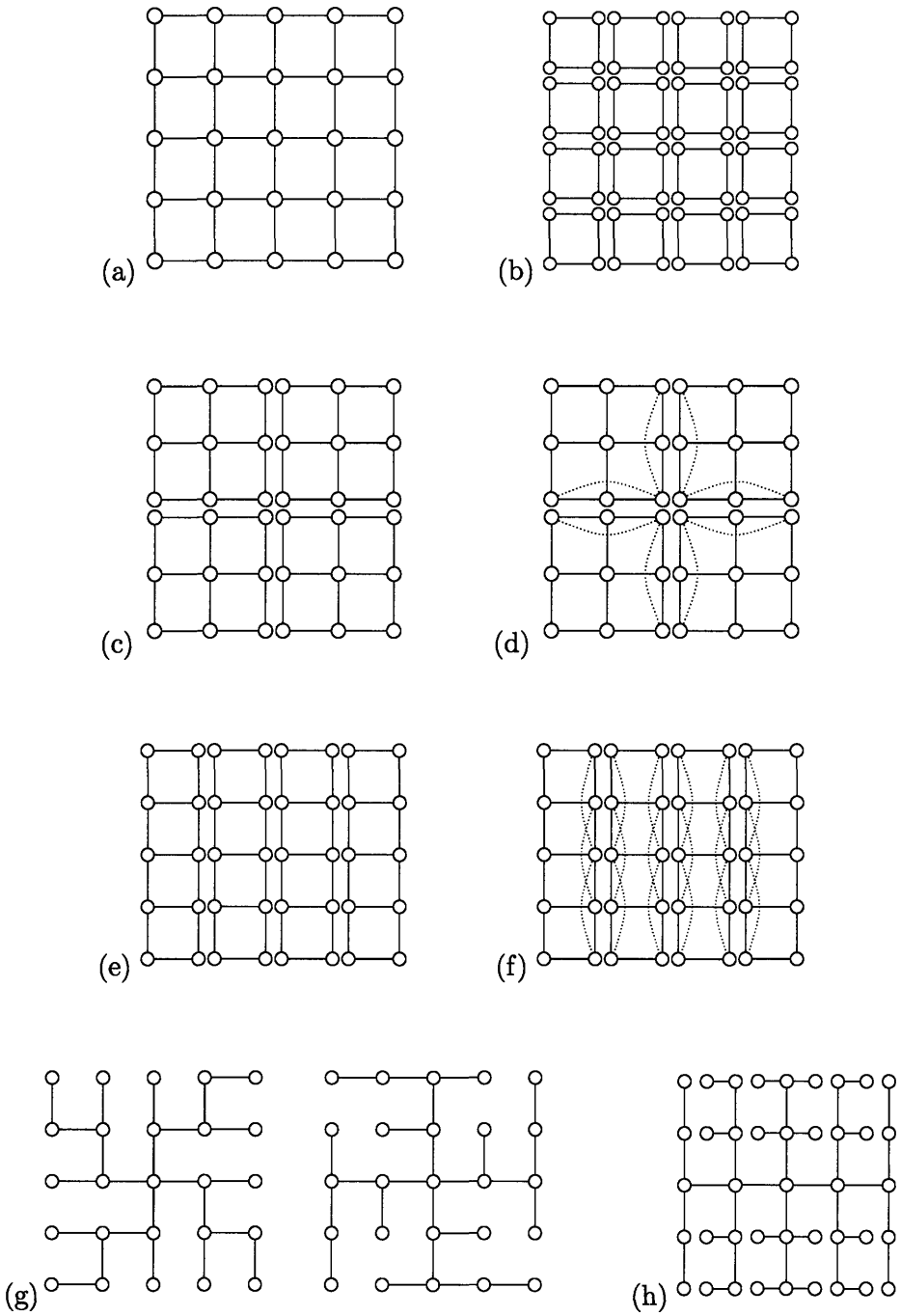

Figure 3.3. Illustrations of a variety of possible ways to obtain a tractable relaxation of a MRF defined over a 5×5 grid shown in (a). The simplest versions of this approach involve edge-wise decompositions as in (b) or small subgraphs as in (c) and (d). One may also break up the graph into thin subgraphs, as in (e) or (f). The TRMP method, using convex combinations of trees, is essentially equivalent to this subgraph decomposition method using embedded trees as in (g). It is also possible "unroll" the graph to obtain a tree as shown in (h).

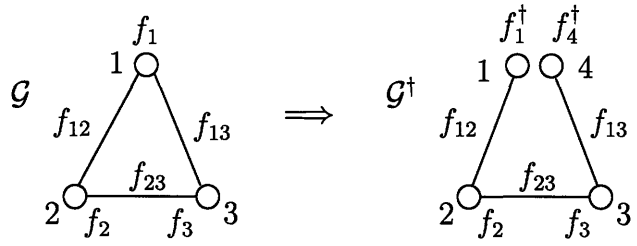

Figure 3.4. A simple preliminary example of Lagrangian relaxation.

the cycle by duplicating one node, thereby reducing the tree-width of the graph from two to one. We then map potentials defined on nodes and edges in $\mathcal{G}$ to corresponding nodes and edge in $\mathcal{G}^\dagger$. For the duplicated variables, $x_1^\dagger$ and $x_4^\dagger$, we must split the node potential f_1 between $f_1^\dagger$ and $f_4^\dagger$ such that $f_1(x_1) = f_1^\dagger(x_1) + f_4^\dagger(x_1)$ for all x_1. Now the problem of maximizing $f(x)$ is equivalent to maximizing $f^\dagger(x^\dagger)$ subject to the constraint $x_1^\dagger = x_4^\dagger$. To solve the latter we relax the constraint using Lagrange multipliers: $L(x^\dagger, \lambda) = f^\dagger(x^\dagger) + \lambda(x_1^\dagger - x_4^\dagger)$. The additional term $\lambda(x_1^\dagger - x_4^\dagger)$ modifies the self-potentials: $f_1^\dagger \leftarrow f_1^\dagger(x_1^\dagger) + \lambda x_1^\dagger$ and $f_4^\dagger \leftarrow f_4^\dagger(x_4^\dagger) - \lambda x_4^\dagger$, parameterizing a family of models on $\mathcal{G}^\dagger$ all of which are equivalent to f under the constraint $x_1^\dagger = x_4^\dagger$. For a fixed λ, solving $\max_x L(x, \lambda) \triangleq g(\lambda)$ gives an upper bound on $f^* = \max_x f(x)$, so by optimizing λ to minimize $g(\lambda)$, we find the tightest bound $g^* = \min_\lambda g(\lambda)$. If the constraint $x_1^\dagger = x_4^\dagger$ is satisfied in the final solution, then there is strong duality $g^* = f^*$ and we obtain the correct MAP assignment for $f(x)$.

■ 3.2.3 Lagrangian Relaxation and Dual Problem

Now, let us proceed with developing the Lagrangian relaxation method and resulting dual problem in the context of block decompositions. As discussed in Section 3.2.1, each decomposition of $f(x)$ over the blocks in $\mathcal{G}$ corresponds to a distinct objective function $f^\dagger(x^\dagger)$ on this extended set of auxiliary variables corresponding to vertices of $\mathcal{G}^\dagger$ and it holds that $\max f \leq \max f^\dagger$ for all valid decompositions. Let us define the *dual function* for a given decomposition over $\mathcal{G}$ by:

$$g(f^\dagger) \triangleq \max_{x^\dagger} f^\dagger(x^\dagger) = \sum_E \max_{x^E} f_E(x^E) \tag{3.8}$$

Then, the *dual problem* is to minimize this dual function over all valid decompositions of f with respect $\mathcal{G}$. Let g^* denote the minimum value of this problem and let $f^* = \max_x f(x)$ denote the MAP value. Then, $g^* \geq f^*$ for all f and we say that *strong duality holds* if $g^* = f^*$ or that there is a *duality gap* if $g^* > f^*$.

To be more precise about how the class of valid decompositions is represented, consider the exponential family representation of the graphical model based on the

energy function:

$$f(x) = \theta^T \phi(x) = \sum_{\alpha \in \mathcal{I}} \theta_\alpha \phi_\alpha(x_\alpha) \tag{3.9}$$

where ϕ is the set of features defining the model, θ are the model parameters, and $\alpha \in \mathcal{I}$ is used to index this set of features and corresponding parameters. To define a graphical model, we restrict each feature ϕ_α to be defined on a subset of variables $x_\alpha \triangleq x_{S(\alpha)}$ where $S(\alpha) \subset V$ indicates the *support* of feature $\phi_\alpha(x) = \phi_\alpha(x_\alpha)$. To specify how the objective $f(x)$ is split among the edges of $\mathcal{G}$, we define and edge-wise parameter vector θ^E on each edge, which is only defined over the subset of features $[E] \subset \mathcal{I}$ that have support within that edge. This represents a valid decomposition if the components θ^E sum up to θ:

$$\sum_{E \in \mathcal{G}} [\theta^E]_{\mathcal{I}} = \theta \tag{3.10}$$

Here, $[\theta^E]_{\mathcal{I}}$ denotes the sparse "zero padded" representation of θ^E with respect to full set of features $\mathcal{I}$. Thus, the collection $\theta^\dagger = (\theta^\dagger, E \in \mathcal{G})$ represents the decomposition:

$$f^\dagger(x^\dagger) = \sum_{E \in \mathcal{G}} f_E(x_E)$$

with

$$f_E(x^E) = (\theta^E)^T \phi_{[E]}(x^E).$$

The overall parameter vector $\theta^\dagger$ also defines the exponential family parameters of the graphical model defined on $\mathcal{G}^\dagger$, with respect to a corresponding set of auxiliary features $\phi^\dagger(x^\dagger) = (\phi^E(x^E), E \in \mathcal{G})$.

Let D denote the linear operator mapping x to a consistent representation in the auxiliary variable $x^\dagger$, that is, $x^\dagger = Dx$. For consistent $x^\dagger$, it also holds that $\phi^\dagger(x^\dagger)$ is a linear function of $\phi(x)$, which we denote by $\phi^\dagger(Dx) = \hat{D}\,\phi(x)$. The condition of valid decomposition is equivalent to the linear constraint:

$$\hat{D}^T \theta^\dagger = \theta \tag{3.11}$$

Then, one can easily verify the property that $f^\dagger(Dx) = f(x)$ for all x:

$$\begin{aligned} f^\dagger(Dx) &= (\theta^\dagger)^T \phi^\dagger(Dx) \\ &= (\theta^\dagger)^T \hat{D} \phi(x) \\ &= (\hat{D}^T \theta^\dagger)^T \phi(x) \\ &= \theta^T \phi(x) \\ &= f(x) \end{aligned} \tag{3.12}$$

This also verifies that $\max f^\dagger \geq \max f$, because every value of $f(x)$ is also a value of $f^\dagger(Dx)$. Now, we have that the optimal value of the MAP problem is

$$f^* = \max_x \theta^T \phi(x) \tag{3.13}$$

and it is bounded above by

$$\boxed{g(\theta^\dagger) = \max_{x^\dagger} \left\{ (\theta^\dagger)^T \phi^\dagger(x^\dagger) \right\} = \sum_{E \in \mathcal{G}} \max_{x^E} \left\{ (\theta^E)^T \phi_{[E]}(x^E) \right\}} \tag{3.14}$$

for all $\theta^\dagger$ such that $\hat{D}^T \theta^\dagger = \theta$. Hence, the dual problem can be simply stated as:

$$\boxed{\begin{array}{ll} \text{(LR-}\theta^\dagger\text{)} & \begin{array}{ll} \text{minimize} & g(\theta^\dagger) \\ \text{subject to} & \hat{D}^T \theta^\dagger = \theta \end{array} \end{array}}$$

This problem is minimizing a convex, piece-wise linear objective function subject to linear constraints. It is tractable to solve, owing to the fact that $g(\theta^\dagger)$ is tractable to compute for a given $\theta^\dagger$ because the maximization over $x^\dagger$ is separable and the maximum over each block is tractable by direct enumeration, provided the blocks sizes are kept small.

Derivation Using Lagrange Multipliers

At this point, it may not yet be apparent to the reader how this dual problem corresponds to the standard method of Lagrangian relaxation. We now rederive the dual problem from this perspective.

First, let us restate the MAP estimation problem defined on $\mathcal{G}$ as a constrained MAP estimation problem with respect to the auxiliary graph $\mathcal{G}^\dagger$. A key idea is that the constraint that $x^\dagger$ is consistent, that is, that $x^\dagger = Dx$ for some x, can be encoded as a set of linear constraints on the feature vector $\phi^\dagger(x^\dagger)$. For a given feature ϕ_α let $\zeta(\alpha)$ denote the set of feature in $\phi^\dagger$ that are duplicates of ϕ_α in the mapping $\phi^\dagger(Dx) = \hat{D}\phi(x)$. Now, we impose constraints that all elements of $\phi^\dagger(x^\dagger)$ corresponding to duplicates of the same feature must be equal:

$$\phi^\dagger_\alpha(x^\dagger) = \phi^\dagger_\beta(x^\dagger) \text{ for all } \alpha, \beta \in \zeta(\gamma) \text{ for some } \gamma \in \mathcal{I} \tag{3.15}$$

We define the matrix C to encode these constraints as:

$$C\phi^\dagger(x^\dagger) = 0 \tag{3.16}$$

The kth row of C corresponds to a pair (α_k, β_k) of duplicated features and contains two non-zero entries: $C_{k,\alpha_k} = +1$ and $C_{k,\beta_k} = -1$.

Let $\mathcal{R}(A)$ and $\mathcal{N}(A)$ respectively denote the *range space* and *null space* of A.[4] The matrices C and $\hat{D}$ satisfy the following:

Lemma 3.2.1. $\mathcal{N}(C) = \mathcal{R}(\hat{D})$.

[4]The *range space* of A is the set of all vectors y that are equal Ax for some vector x. The *null space* of A is the set of all vectors x such that $Ax = 0$. For an $m \times n$ matrix A, these are vector subspaces of (respectively) $\mathbb{R}^m$ and $\mathbb{R}^n$.

Proof. If $C\eta^\dagger = 0$, then $\eta^\dagger_{\alpha_k} - \eta^\dagger_{\beta_k} = 0$ for all k so that $\eta^\dagger$ is consistent and hence $\eta^\dagger = D\eta$ for some η. This shows that $\mathcal{N}(C) \subseteq \mathcal{R}(\hat{D})$. If $\eta^\dagger = D\eta$, then $\eta^\dagger_{\alpha_k} = \eta^\dagger_{\beta_k}$ and $(D\eta^\dagger)_k = \eta^\dagger_{\alpha_k} - \eta^\dagger_{\beta_k} = 0$ for all k. This shows that $\mathcal{R}(\hat{D}) \subset \mathcal{N}(A)$. Hence, $\mathcal{N}(C) = \mathcal{R}(\hat{D})$. □

This asserts that $C\eta^\dagger = 0$ if and only if $\eta^\dagger = \hat{D}\eta$ for some η.

Let $\tilde{\theta}^\dagger$ be any valid decomposition of θ. We can reformulate the MAP problem defined on $\mathcal{G}$ as the following *constrained* MAP estimation problem defined on $G^\dagger$:

$$\boxed{\begin{array}{ll} \text{(MAP-}\mathcal{G}^\dagger\text{)} & \begin{array}{ll} \text{maximize} & \tilde{f}^\dagger(x^\dagger) \triangleq (\tilde{\theta}^\dagger)^T\phi^\dagger(x^\dagger) \\ \text{subject to} & C\phi^\dagger(x^\dagger) = 0 \end{array} \end{array}}$$

Now, to obtain the dual problem, we use the Lagrangian relaxation method. That is, we relax the linear constraints on $\phi^\dagger(x^\dagger)$, introducing Lagrange multipliers to impose penalties for constraint violations in the Lagrangian objective:

$$\begin{aligned} L(x^\dagger;\lambda) &\triangleq (\tilde{\theta}^\dagger)^T\phi^\dagger(x^\dagger) + \lambda^T C\phi^\dagger(x^\dagger) \\ &= \left(\tilde{\theta}^\dagger + C^T\lambda\right)^T \phi^\dagger(x^\dagger) \end{aligned} \tag{3.17}$$

The main point here is that the Lagrange multipliers serve to parameterize the space of valid decompositions. This is based on the following observation:

Lemma 3.2.2. $\mathcal{R}(C^T) = \mathcal{N}(\hat{D}^T)$.

Proof. This is a consequence of Lemma 3.2.1. Given a vector subspace $\mathbb{V} \subset \mathbb{R}^n$, the *orthogonal complement* is the set $\mathbb{V}^\perp$ of all vectors $u \in \mathbb{R}^n$ such that $u^Tv = 0$ for all $v \in \mathbb{V}$. Note that $(\mathbb{V}^\perp)^\perp = \mathbb{V}$. We recall a fundamental result of linear algebra: $\mathcal{R}(A)^\perp = \mathcal{N}(A^T)$. Then, we have: $\mathcal{R}(C^T) = \mathcal{N}(C)^\perp = \mathcal{R}(\hat{D})^\perp = \mathcal{N}(\hat{D}^T)$, where we used Lemma 3.2.1 in the second step. □

In other words, $D^T\Delta\theta^\dagger = 0$ if and only if $\Delta\theta^\dagger = C^T\lambda$ for some λ. Given some valid decomposition $\tilde{\theta}^\dagger$, it follows that $D^T(\theta^\dagger - \tilde{\theta}^\dagger) = 0$ if and only if $\theta^\dagger - \tilde{\theta}^\dagger = C^T\lambda$. Using $D^T\tilde{\theta}^\dagger = \theta$, this shows that $D^T\theta^\dagger = \theta$ if and only if $\theta^\dagger = \tilde{\theta}^\dagger + C^T\lambda$ for some λ. Thus, $\theta^\dagger$ is a valid decomposition of θ if and only if it may be expressed as $\tilde{\theta}^\dagger + C^T\lambda$ for some λ.

The Lagrangian dual function is defined by the maximum value of the Lagrangian objective over all $x^\dagger$ for a given value of λ:

$$\boxed{g(\lambda) \triangleq \max_{x^\dagger} \left\{ (\tilde{\theta}^\dagger + C^T\lambda)^T\phi^\dagger(x^\dagger) \right\}} \tag{3.18}$$

Then, the Lagrangian dual problem is simply to minimize $g(\lambda)$ with respect to λ.

$$\boxed{\text{(LR-}\lambda\text{)} \qquad \text{minimize } g(\lambda)}$$

This is equivalent to minimizing the dual function $g(\theta^\dagger)$ in (3.14) over the subspace of all valid decompositions of θ. Note that there is a slight abuse of notation in that we allow $g(\cdot)$ to represent either the dual function as a function of Lagrange multipliers λ or as a function of valid decompositions $\theta^\dagger$.

■ 3.2.4 Dual Optimality, Strong Duality and Constraint Satisfaction

We now consider optimality conditions, both for minimizing the dual function $g(\theta^\dagger)$ (3.14) in (LR-$\theta^\dagger$) over valid decompositions and for testing for strong duality. We state and prove these conditions for block decompositions, but analogous results hold in the general subgraph decomposition method using essentially the same proofs (further comments on subgraph decompositions may be found in Appendix A).

For a model θ, let $f^* = \max_x f(x)$ denote the value of the MAP problem and let $\hat{X} = \{x | f(x) = f^*\}$ denote the set of MAP estimates. Given a decomposition $\theta^\dagger$, let $f_E^* = \max_{x^E} f_E(x^E)$ and define the *block-wise MAP sets* by

$$\hat{X}^E = \{x^E | f_E(x^E) = f_E^*\}. \tag{3.19}$$

That is, $\hat{X}^E$ represents the set of optimal MAP estimates with respect to the component f_E of the block decomposition. Note, this determines the set of MAP estimates in the decomposed graphical model as a Cartesian product of the local block-wise MAP sets on each connected component of $\mathcal{G}^\dagger$:

$$\hat{X}^\dagger = \hat{X}^{E_1} \otimes \cdots \otimes \hat{X}^{E_m} \tag{3.20}$$

By an abuse of notation, we write $x \in \hat{X}^E$ to indicate that $x_E \in \hat{X}^E$ or $x \in \hat{X}^A \cap \hat{X}^B$ to indicate that both $x_A \in \hat{X}^A$ and $x_B \in \hat{X}^B$. Roughly speaking, we are letting $\hat{X}^E$ actually represent the set $\{x | x_E \in \hat{X}^E\}$ in such expressions. We say that the collection of block MAP sets $\{\hat{X}^E\}_{E \in \mathcal{G}}$ is *pairwise satisfiable* if for every pair of edges $A, B \in \mathcal{G}$ it holds that $\hat{X}^A \cap \hat{X}^B$ is non-empty. Note that the word "pairwise" in "pairwise satisfiable" does *not* refer to pairwise edges or pairwise graphs, but rather it refers to *pairs of edges* in $\mathcal{G}$ and these can be generalized edges of any size. We say that this collection of sets is *satisfiable* if $\hat{X}_{\text{LR}} \triangleq \cap_{E \in \mathcal{G}} \hat{X}^E$ is non-empty. This is equivalent to the condition that there exists an $x^\dagger \in \hat{X}^\dagger$ which is *consistent*, such that $x^\dagger = Dx$ for some x. Note that pairwise satisfiability does not imply satisfiability.

We say that a valid decomposition $\theta^\dagger$ is *dual optimal*, or that this value of $\theta^\dagger$ corresponds to a *dual optimal decomposition*, if it is a solution of the dual problem, that is, if $g(\theta^\dagger) = g^* \triangleq \min g$. We begin by proving the following *necessary* condition for dual optimality:

Lemma 3.2.3. *Every dual optimal decomposition is pairwise satisfiable.*

Proof. We show that if the decomposition is not pairwise satisfiable, then there exists a valid perturbation of the decomposition (a perturbation of the Lagrange multipliers) that decreases the dual function. If the decomposition is not pairwise satisfiable, then

there exists two intersecting edges A, B such that the $\hat{X}^A \cap \hat{X}^B$ is empty. Let $S = A \cap B$ denote the intersection of these two edges. Let $f_A(x_A)$ and $f_B(x_B)$ denote the potentials on edges A and B determined by $\theta^\dagger$. We define a perturbation of these as follows. For all x let

$$\begin{aligned} \tilde{f}_A(x_S, x_{A\setminus S}) &= f_A(x_S, x_{A\setminus S}) - \lambda(x_S) \\ \tilde{f}_B(x_S, x_{B\setminus S}) &= f_B(x_S, x_{B\setminus S}) + \lambda(x_S) \end{aligned}$$

For any choice of $\lambda(x_S)$, this is a valid decomposition since $\tilde{f}_A(x)+\tilde{f}_B(x) = f_A(x)+f_B(x)$ for all x. Now, let $\lambda(x_S) = \epsilon$ for all x_S that occurs in any optimal configuration in $\hat{X}_A$ and $\lambda(x_S) = 0$ otherwise. Note that these values of x_S which are optimal with respect to $\hat{X}^A$ are not optimal in any configuration of $\hat{X}^B$. For $\epsilon \geq 0$ sufficiently small, we have that $\max \tilde{f}_A = \max f_A - \epsilon$ (because we have decreased all optimal configurations on A by ϵ) and $\max \tilde{f}_B = \max f_B$ (because no configuration that is optimal on A is also optimal on B). Therefore,

$$\max \tilde{f}_A + \max \tilde{f}_B = \max f_A + \max f_B - \epsilon,$$

and $g(\tilde{f}) = g(f) - \epsilon > g(f)$, which contradicts optimality of the original decomposition. Hence, any dual optimal decomposition must be pairwise satisfiable. □

It must be emphasized that the above local consistency is *not sufficient* to ensure dual optimality. This is because these conditions correspond to coordinate-wise optimality, which is not in general sufficient to ensure that a non-differentiable function is minimized. This notion of local consistency is closely related to the concept of weak tree agreement in the TRMP method [134, 135, 211].

Note that requiring that $\hat{X}^A \cap \hat{X}^B$ does not imply that the set of optimal configurations of x_S that occur in $\hat{X}^A$ is the same as those that occur in $\hat{X}^B$, but only that these two sets have non-empty intersection. Let us say that the collection $\{\hat{X}^E\}$ is *pairwise consistent* if, for all pairs of intersecting edges $A \cap B = S$, it holds that $\hat{X}^A$ and $\hat{X}^B$ give *exactly the same* sets of optimal configuration on x_S. Thus, every pairwise consistent decomposition is also pairwise satisfiable, but the reverse is untrue. However, we now show that there always exists a pairwise consistent dual optimal decomposition.

Lemma 3.2.4. *There exists a pairwise consistent dual optimal decomposition.*

Proof. By the preceding lemma, any dual optimal decomposition is pairwise satisfiable. Using this property, we show that any dual optimal decomposition can be perturbed slightly to obtain another dual optimal decomposition that is also pairwise consistent. For a given pair of inconsistent (but satisfiable) edges A, B, there exists some x'_S that is optimal in $\hat{X}^A$ but not $\hat{X}^B$. Then, we again perturb f_A and f_B by $\lambda(x_S)$ (with opposite signs), with $\lambda(x'_S) = \epsilon$ and $\lambda(x_S) = 0$ if $x_S \neq x'_S$. Then, for ϵ sufficiently small, we still have $\max f_A = \max f_A$ (because there must exist another element of $\hat{X}_A$ that maximizes f_A with $x_S \neq x'_S$, owing to pairwise satisfiability). Also, $\max f'_B = \max f_B$ (because x_S was not optimal with respect to B). Therefore $g' = g$,

and this new decomposition is still dual optimal. Furthermore, the new optimal set $\hat{X}^A$ no longer includes any configurations with $x_S = x'_S$. Thus, we have removed this inconsistency. By repeating this procedure for every such inconsistency, we eventually obtain a dual optimal decomposition that is pairwise consistent. □

Later, we use this lemma to help demonstrate strong duality for several classes of tractable graphical models. Next, we prove the following result characterizing when strong duality occurs:

Proposition 3.2.1. *Strong duality holds if and only if there exists a decomposition such that the block-wise MAP sets are jointly satisfiable. When strong duality holds, the set $\hat{X}_{\text{LR}} = \cap_{E \in \mathcal{G}} \hat{X}^E$ is identical to the set of all optimal solutions of the MAP problem. Also, satisfiability implies dual optimality and is necessary if strong duality holds.*

Proof. First, we prove that existence of a satisfiable decomposition implies strong duality. It always holds, by construction of the dual problem, that $f^* \leq g^*$. We must show that $f^* \geq g^*$. By satisfiability, there exists $\hat{x}$ such that $f_E(\hat{x}_E) = f_E^*$ for all $E \in \mathcal{G}$. Then,

$$\begin{aligned} f(\hat{x}) &= f^\dagger(D\hat{x}) \\ &= \sum_{E \in \mathcal{G}} f_E(\hat{x}_E) \\ &= \sum_{E \in \mathcal{G}} f_E^* \\ &= g^* \end{aligned}$$

Since $f(\hat{x})$ is a lower-bound for f^*, this implies that $g^* \leq f^*$. Hence, $g^* = f^*$. This also show that every satisfiable decomposition is dual optimal and every $x \in \hat{X}_{\text{LR}}$ is an optimal solution to the MAP problem.

Next, we prove that strong duality $g^* = f^*$ implies existence of a decomposition that is satisfiable. Recall that the Lagrangian (3.17) is given by

$$L(x^\dagger, \lambda) = \tilde{f}(x^\dagger) + \lambda^T C \phi^\dagger(x^\dagger),$$

which is specified relative to a valid decomposition $\tilde{f}(x^\dagger)$ of the objective function $f(x)$. As is well-known [24], strong duality obtains in the Lagrangian method if and only if there exists a saddle point of the Lagrangian, that is a pair $(\hat{x}^\dagger, \hat{\lambda})$ such that:

$$L(\hat{x}^\dagger, \hat{\lambda}) = \max_{x^\dagger} L(x^\dagger, \hat{\lambda}) \tag{3.21}$$

and

$$L(\hat{x}^\dagger, \hat{\lambda}) = \min_{\lambda} L(\hat{x}^\dagger, \lambda). \tag{3.22}$$

The condition (3.22) implies $\nabla_\lambda L(\hat{x}^\dagger, \lambda) = 0$, which is equivalent to $C\phi^\dagger(x^\dagger) = 0$. This condition holds if and only if $\hat{x}^\dagger = D\hat{x}$ for some $\hat{x}$. We use this $\hat{x}$ to show that the

decomposition $\hat{f}^\dagger$ specified by $\hat{\theta}^\dagger = \tilde{\theta}^\dagger + C^T\hat{\lambda}$ is satisfiable. First, note that $L(\hat{x}^\dagger, \hat{\lambda}) = \sum_E \hat{f}_E(\hat{x}_E)$. The condition (3.21) requires that $\sum_E \hat{f}_E(\hat{x}_E) = \sum_E \max_{x^E} \hat{f}_E(x^E)$. Hence, we must have $\hat{f}_E(\hat{x}_E) = \hat{f}_E^*$ for all $E \in \mathcal{G}$. Thus, this decomposition $\hat{f}^\dagger$ is satisfiable based on $\hat{x}$.

It is also a standard result of Lagrangian duality [24] that, if there exists a saddle point, then the set of all saddle point is the Cartesian product of the set of primal solutions and the set of dual solutions. Hence, when strong duality obtains, the set $\hat{X}_{\mathrm{LR}} = \cap_E \hat{X}^E$ is equal to $\hat{X}$, the set of MAP estimates, and *every* dual optimal decomposition is satisfiable. □

In other words, strong duality is equivalent to existence of a decomposition such that all blocks are simultaneously maximized by some x. If there is a duality gap, then minimizing the dual function must lead to a dual optimal decomposition for which the block-wise MAP sets are pairwise satisfiable but not jointly satisfiable. We must emphasize that, even if strong duality is obtained, this does not imply that every solution $x^\dagger \in \arg\max f^\dagger$ is consistent (so as to determine a solution of the MAP problem). This difficulty implies that it is in general intractable to determine if there is a duality gap and to obtain an MAP solution when there is no duality gap. The intractability is due to the fact that the general constraint satisfaction problem in NP-complete. However, it most cases it is tractable to detect when strong duality obtains based on the following lemma (this generalizes a result derived previously for TRMP [211], which corresponds to tree-based decompositions within our framework).

Corollary 3.2.1. *Let $\theta^\dagger$ be an optimal decomposition in the dual problem. If each edge has a unique MAP solution, that is, if for each $E \in \mathcal{G}$ the set $\hat{X}^E$ contains a single element, then strong duality obtains and $\hat{X}^E = \{\hat{x}_E\}$ for some $\hat{x}$ and all $E \in \mathcal{G}$. Then, this $\hat{x}$ is the unique solution to the MAP estimation problem.*

Proof. From Lemma 3.2.3, every optimal block decomposition is pairwise satisfiable. If each block also has a unique MAP estimate, then pairwise satisfiability implies that these local estimates are all consistent. Hence, there is a global $\hat{x}$ with maximizes all blocks. Then, by Proposition 3.2.1, strong duality holds and this is the unique MAP estimate. □

There is a sense in which this is the only likely outcome if strong duality holds. Essentially, the subset of model parameters θ for which strong duality obtains and there exists a dual optimal decomposition that does not satisfy the above uniqueness condition is a zero-volume set in the space of model parameters. We provide some intuition for why this is so in the following section when we discuss the linear programming interpretation of the dual problem. Hence, if we chose θ at random from a *probability density* on the space of model parameters it holds almost surely that either: (i) strong duality holds and the optimal block decomposition leads to a unique MAP estimate, or (ii) there is a duality gap and the dual optimal decomposition is non-unique. In many applications, this condition is met by virtue of conditioning on noisy continuous measurements of

x. If that is not the case, for example in combinatorial problems with integer-valued θ parameters, we can add a small random perturbation to the model parameters so as to ensure that this condition is met. Then, by solving the perturbed model, we know whether or not strong duality holds in the perturbed model and recover a near-optimal solution to the original problem in the case that strong duality does hold. In fact, if θ is integer valued, we can ensure that the unique MAP estimate of the perturbed problem is also an MAP estimate of the original problem by making the perturbation small enough.

■ 3.2.5 Linear Programming Interpretation and Duality

Let us now consider the linear programming (LP) interpretation of the Lagrangian dual problem. This both serves to give some geometric intuition for the results derived in the preceding section, and to demonstrate the connection to LP relaxation methods for MAP estimation.

Geometric Interpretation of the Lagrangian Dual Problem

We illustrate a geometric interpretation of the Lagrangian dual function $g(\lambda)$ in Figure 3.5. The dual function is the maximum over a finite set of linear functions in λ indexed by $x^\dagger$. For each $x^\dagger \in \mathbb{X}^\dagger$, there is a linear function $g(\lambda; x^\dagger) = a(x^\dagger)^T\lambda + b(x^\dagger)$, with $a(x^\dagger) = C\phi^\dagger(x^\dagger)$ and $b(x^\dagger) = (\tilde{\theta}^\dagger)^T\phi^\dagger(x^\dagger)$. The graph of each of these functions defines a hyper-plane in $\mathbb{R}^{d+1}$, where d is the number of Lagrange multipliers. The flat hyper-planes, with $a(x^\dagger) = 0$, correspond to consistent $x^\dagger$. The remaining sloped hyper-planes represent inconsistent $x^\dagger$. Hence, the highest flat hyper-plane corresponds to the optimal MAP estimate, with height equal to f^*. The dual function $g(\lambda)$ is defined by the maximum height over the set of all hyper-planes (both the consistent and inconsistent ones) for each λ, and is therefore convex, piece-wise linear and greater than or equal to f^* for all λ.

Consider again the main point of the previous section from this perspective. If there is a duality gap, as seen in Figure 3.5(a), the minimum of the dual function occurs at a single point corresponding to an intersection of two sloped hyper-planes that "hide" the flat hyper-planes. This corresponds to the fact that all MAP estimates $x^\dagger$ of the relaxed model defined over $\mathcal{G}^\dagger$ are inconsistent configurations, such that $C\phi^\dagger(x^\dagger) \neq 0$, and the dual optimal decomposition is not satisfiable. If strong duality holds, as seen in Figure 3.5(b), then the minimum is defined by the highest flat hyper-plane corresponding to a consistent assignment. This then is the optimal MAP estimate. Its intersection with slanted hyper-planes defines the polytope of optimal Lagrange multipliers over which the maximum flat hyper-plane is exposed. Thus, there is generally a convex, polyhedral set of optimal λ's that minimize the dual function. We can now see that, for randomly chosen model parameters θ, these are the two likely outcomes of minimizing the dual function. Either there is no duality gap and a unique MAP estimate, in which case the set of optimal Lagrange multipliers is a set of non-zero volume, or there is a duality gap and the set of optimal Lagrange multipliers has zero volume. However, there is

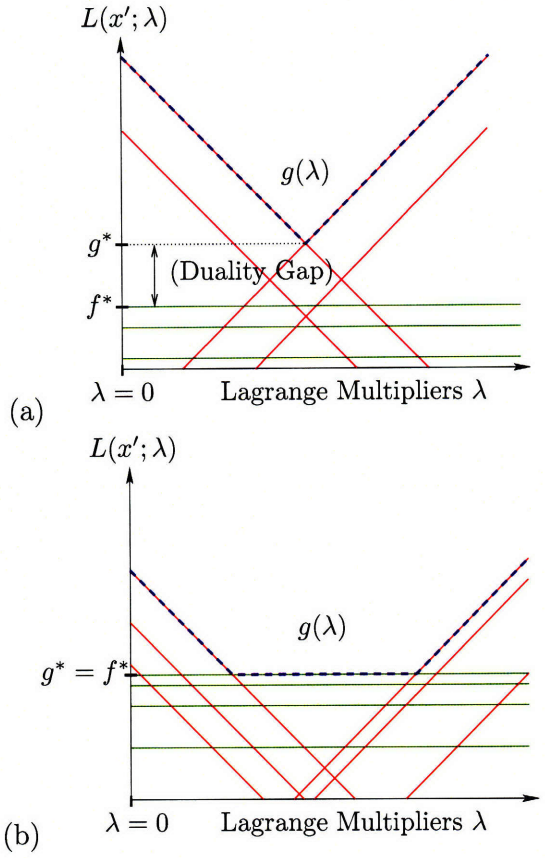

Figure 3.5. Illustration of the Lagrangian duality in the cases that (a) there is a duality gap and (b) there is no duality gap (strong duality holds).

also the exceptional (unlikely) case that the maximum flat hyper-plane passes exactly through the minimum defined by an intersection of sloped hyper-planes or there are multiple optimal MAP estimates. In these exceptional cases, it becomes difficult to detect when strong duality has been achieved or to recover a consistent MAP estimate. Even then, a small random perturbation of θ causes all these hyper-planes to be shifted slightly to avoid this exceptional case (and by repeating this procedure enough times we should find a case where strong duality occurs and thereby obtain an MAP estimate). Thus, the main failure mode of the Lagrangian relaxation method is the occurrence of a duality gap.

Description as an LP

The problem of minimizing $g(\lambda)$ can be equivalently stated as the linear programming problem of finding the lowest-point of the *epigraph* of $g(\lambda)$, defined as the set of all points in $\mathbb{R}^{d+1}$ that lie "above" the graph of $g(\lambda)$, that is, all points (λ, h) such that

$g(\lambda) \leq h$. This epigraph may be specified by the set of constraints: $g(\lambda; x^\dagger) \leq h$ for all $x^\dagger$. Each of these constraints is a linear inequality constraint on (λ, h) and therefore defines a "half-space" constraint, such that the epigraph of $g(\lambda)$ is a polytope. Thus, we arrive at the following LP formulation of the dual problem:

$$\boxed{\begin{array}{ll} \text{(LR-epi)} & \begin{array}{ll} \text{minimize} & h \\ \text{subject to} & (\tilde{\theta}^\dagger + C^T\lambda)^T\phi^\dagger(x^\dagger) \leq h \text{ for all } x^\dagger. \end{array} \end{array}}$$

Note that the minimization is over (λ, h). This can also be stated as an optimization over valid decompositions $\theta^\dagger$:

$$\begin{array}{ll} \text{minimize} & h \\ \text{subject to} & \hat{D}^T\,\theta^\dagger = \theta \\ & (\theta^\dagger)^T\phi^\dagger(x^\dagger) \leq h \text{ for all } x^\dagger. \end{array} \tag{3.23}$$

Here, the problem variables are $(\theta^\dagger, h)$ and we have the extra linear constraint $\hat{D}^T\theta^\dagger = \theta$. Of course, these LPs are intractable to solve explicitly, as they specify an exponential number of constraints. We only introduce this formulation to consider its dual LP.

The Dual LP

Using linear-programming duality, we show that the dual of this LP, obtained by relaxing the inequality constraints, reduces to the following LP relaxation of the MAP estimation problem:

$$\boxed{\begin{array}{ll} \text{(LP-}\hat{\mathcal{M}}\text{)} & \begin{array}{ll} \text{maximize} & \theta^T\eta \\ \text{subject to} & \eta \in \hat{\mathcal{M}}(\mathcal{G}) \end{array} \end{array}}$$

where

$$\hat{\mathcal{M}}(\mathcal{G}) = \cap_{E\in\mathcal{G}}\mathcal{M}_E = \{\eta \mid \eta_{[E]} \in \mathcal{M}_E\}$$

and $\mathcal{M}_E$ is the set of realizable moments $\eta_{[E]}$. The key point here is that the relaxation of the marginal polytope that one uses in the LP relaxation of MAP estimation is linked to the block structure used in the block decomposition method. Thus, adding additional blocks in the block decomposition method results in tighter approximations to the marginal polytope and correspondingly tighter bounds on the value of the MAP problem. We have already described how a duality gap in the block decomposition method appears as a non-unique MAP estimate in the optimal decomposition. Such "ties" in the block decomposition method are related to the occurrence of fractional solutions in the LP relaxation method, in which some or all node marginals contain fractional probabilities (between zero and one). We say that the LP has an *integrality gap* if there are no integral (non-fractional) solutions.

Proposition 3.2.2. *Problems (3.23) and (LP-$\hat{\mathcal{M}}$) are related by LP duality and have equal values. Thus, there is a duality gap for the optimal block decomposition over $\mathcal{G}$ if and only if there the LP relaxation based on $\hat{\mathcal{M}}(\mathcal{G})$ has an integrality gap.*

Proof. We take the linear-programming dual of problem (3.23) and show that it reduces to (LP-$\hat{\mathcal{M}}$). First, we define the LP Lagrangian, introducing Lagrange multipliers $\mu(x^\dagger) \geq 0$ associated to the inequality constraints (we do not relax the linear constraint):

$$\begin{aligned} L(h, \theta^\dagger; \mu) &= h + \sum_{x^\dagger} \mu(x^\dagger) \left[(\theta^\dagger)^T \phi^\dagger(x^\dagger) - h \right] \\ &= (1 - \sum_{x^\dagger} \mu(x^\dagger)) h + (\theta^\dagger)^T \left(\sum_{x^\dagger} \mu(x^\dagger) \phi^\dagger(x^\dagger) \right) \end{aligned}$$

Then, the dual function is defined as the maximum over all $(h, \theta^\dagger)$ for each $\mu \geq 0$:

$$f(\mu) = \max_{h, \theta^\dagger : \hat{D}^T \theta^\dagger = \theta} L(h, \theta^\dagger; \mu) \tag{3.24}$$

Note that the value of the dual function is $+\infty$ unless $\sum_{x^\dagger} \mu(x^\dagger) = 1$. Hence, μ defines a probability distribution over $x^\dagger$ and $\eta^\dagger(\mu) \triangleq \sum_{x^\dagger} \mu(x^\dagger) \phi^\dagger(x^\dagger)$ defines a set of realizable moments $\eta^\dagger(\mu) \in \mathcal{M}(\mathcal{G}^\dagger)$ with respect to the auxiliary graphical model arising from the block decomposition. Also, because we maximize over all $\theta^\dagger$ in the null-space of $\hat{D}$, the value of the dual function is $+\infty$ unless $\eta^\dagger(\mu)$ is orthogonal to this null space, which is equivalent to the range space of C. This leads to the consistency constraint $C\eta^\dagger(\mu) = 0$ in the dual problem. For $\eta^\dagger(\mu)$ that satisfy this constraint, the value of the dual function is equal to $(\tilde{\theta}^\dagger)^T \eta^\dagger(\mu)$ for any valid block decomposition $\tilde{\theta}^\dagger$. Thus, the dual problem can be reduced to:

(LP-$\mathcal{M}^\dagger$)	maximize	$(\tilde{\theta}^\dagger)^T \eta^\dagger$
	subject to	$\eta^\dagger \in \mathcal{M}(\mathcal{G}^\dagger)$
		$C\eta^\dagger = 0$

Finally, we recall that $C\eta^\dagger = 0$ implies that $\eta^\dagger = \hat{D}\eta$ for some η. Thus, the objective transforms to $(\theta^\dagger)^T \hat{D}\eta = (\hat{D}^T \theta^\dagger)^T \eta = \theta^T \eta$ and the dual LP reduces to:

$$\begin{aligned} &\text{maximize} && \theta^T \eta \\ &\text{subject to} && \eta \in \hat{\mathcal{M}}(\mathcal{G}) \end{aligned} \tag{3.25}$$

Thus, we have shown that (3.23) and (LP-$\hat{\mathcal{M}}$) are LP dual problem. Then, by strong duality for linear programs, we have that the value of the two problems are equal. □

Consider the geometric interpretation of the relation between the marginal polytope on $\mathcal{G}^\dagger$ and the pseudo-marginal polytope on $\mathcal{G}$. We refer to Figure 3.6 to illustrate this discussion. Each vertex of the marginal polytope $\mathcal{M}(\mathcal{G}^\dagger)$ represents one of the points $\phi^\dagger(x^\dagger)$ for some $x^\dagger$. In Figure 3.6(a), these points are shown as green and red vertices, which respectively represent consistent and inconsistent states. The marginal polytope $\mathcal{M}(\mathcal{G})$ is the convex hull of all these points (the outline of which is drawn in blue in

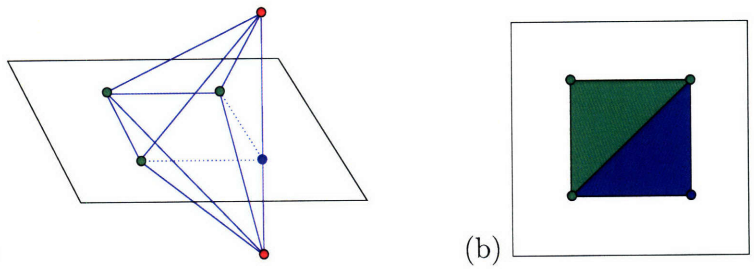

Figure 3.6. Illustration of relation between Lagrangian relaxation and LP relaxation of MAP estimation. (a) Depiction of the marginal polytope $\mathcal{M}(\mathcal{G}^\dagger)$ (drawn in blue) and the subspace of consistent moments $\eta^\dagger = D\eta$. The green vertices represent consistent states, the red vertices represent inconsistent ones. (b) Illustration of the intersection of $\mathcal{M}(\mathcal{G}^\dagger)$ with the subspace of consistent moments. This defines the pseudo-marginal polytope $\hat{\mathcal{M}}(\mathcal{G})$ (green and blue areas), which has a pseudo-vertex (blue) corresponding to a convex combination of inconsistent vertices of $\mathcal{M}(\mathcal{G}^\dagger)$.

the figure). As shown in the proof of the preceding proposition, the dual interpretation of the Lagrangian dual problem is an LP over this polytope, but subject to the linear constraint $C\eta^\dagger = 0$. In Figure 3.6(a), we have drawn a plane to indicate this subspace of points, which includes all of the vertices of $\mathcal{M}(G^\dagger)$ corresponding to consistent states $x^\dagger$ (the green vertices). The marginal polytope $\mathcal{M}(\mathcal{G})$ corresponds to the convex hull of this set of consistent points. The inconsistent points (seen in red) lie outside of this subspace. Note that the intersection of the marginal polytope $\mathcal{M}(\mathcal{G}^\dagger)$ with the consistent subspace defines another polytope, drawn separately in 3.6(b). However, this "slice" of $\mathcal{M}(\mathcal{G}^\dagger)$ is larger then the $\mathcal{M}(\mathcal{G})$, it contains additional vertices (the blue vertex in the figure) that do not correspond to any vertex of $\mathcal{M}(\mathcal{G}^\dagger)$. Rather, they represent a convex combination of two or more inconsistent vertices of $\mathcal{M}(\mathcal{G}^\dagger)$, chosen such that the *averaged* constraint violation is zero. It is this intersection of $\mathcal{M}(\mathcal{G}^\dagger)$ with the subspace defined by the consistency constraint $C\eta^\dagger = 0$ that defines the pseudo-marginal relaxation $\hat{\mathcal{M}}(\mathcal{G})$.

When there is a duality gap in the Lagrangian dual method, this corresponds to a vertex of the pseudo-marginal polytope $\hat{\mathcal{M}}(\mathcal{G})$ that corresponds to a convex combination of two or more inconsistent states on $\mathcal{G}^\dagger$. The weight placed on each of these inconsistent states corresponds to the Lagrange multipliers $\mu(x^\dagger)$ in the LP-dual of our Lagrangian dual problem. Based on the complementary-slackness condition, the optimal μ must have support only over those $x^\dagger$ that are optimal MAP estimates of $\hat{\mathcal{G}}^\dagger$ at the minimum of the dual function, that is, those configurations that correspond to the slanted hyperplanes in Figure 3.5(b) that define the minimum of the dual function. Thus, the η vector that arises from solving the LP relaxation must correspond to a fractional solution, that is, a solution in which at least one node has a probability distribution with some fractional values.

■ 3.2.6 Some Tractable Problems

We now consider two cases where strong duality can be guaranteed: (1) block relaxations based on the maximal cliques of a chordal graph and (2) pairwise relaxation of ferromagnetic Ising models.

Thin Chordal Graphs

First, we consider the block relaxation based on the set of maximal cliques of a chordal graph.

Proposition 3.2.3 (chordal duality). *Let $\mathcal{G}$ be the set of maximal cliques of some chordal graph and let θ be a Markov model on $\mathcal{G}$ (or any subgraph of $\mathcal{G}$). Then, strong duality holds in the block decomposition based on $\mathcal{G}$.*

Proof. A graph is chordal if and only if there exists a junction tree of the graph. This is a tree whose nodes correspond to maximal cliques of the graph and with edges defined such that the running intersection property holds (the intersection of any two cliques is included by every clique along the path between them in the junction tree). Using this property, we show that pairwise consistency of the clique-wise optimal sets implies global consistency in chordal graphs. Then, by Proposition 3.2.1, strong duality obtains. We construct a global configuration $\hat{x}$ satisfying all of the clique optimality conditions as follows. Pick an arbitrary starting node C_0 and define this to be the root of the junction tree. At the root node, we assign the sub-vector $\hat{x}_{C_0}$ to be any element of the set $\hat{X}^{C_0}$. Now, we order the remaining cliques according to their distance from the root node in the junction tree, such that the parent of each clique precedes its children. For each clique C_k, we sequentially assign the sub-vector $\hat{x}_{C_k}$ to be an element of the set $\hat{X}^{C_k}$ that is consistent with any preassigned values. By the running intersection property, any preassigned nodes lie in the intersection of C_k with its parent clique. Then, by Proposition 3.2.3, there exists an optimal configuration of x_{C_k} that is consistent with the parent's configuration. □

As a special case of this general result we see that strong duality is obtained for pairwise LR on trees. More generally, any thin graph can be solved using a block LR algorithm based on the maximal cliques in a triangulated version of the graph (a chordal super-graph). Of course, in trees and thin graphs, we can instead simply solve the MAP problem directly using dynamic programming techniques (e.g., the max-product algorithm). Nonetheless, it is good to see that the LR approach is at least as powerful as dynamic programming.

Also, as a corollary of this result, we infer the following:

Corollary 3.2.2. *Consider a subgraph decomposition $\mathcal{G} = \cup_k \mathcal{G}^{(k)}$ in which each $\mathcal{G}^{(k)}$ includes the set of maximal cliques of a chordal graph as its edges (this chordal graph may be different for each k). Then, the values of the dual problem, using either the block decomposition based on $\mathcal{G}$ or the subgraph decomposition based on $\{\mathcal{G}^{(k)}\}$, are equal.*

Proof. Let g^*_{block} and g^*_{sub} denote the values of the dual problems using either the block or subgraph decomposition. First, we show that $g^*_{\text{sub}} \leq g^*_{\text{block}}$. This follows as we have for each subgraph:

$$\max_{x^{(k)}} f^{(k)}(x^{(k)}) \leq \sum_{E \in \mathcal{G}^{(k)}} \max_{x_E} f^{(k)}(x_E)$$

Thus, any subgraph decomposition can be relaxed to a block decomposition of greater or equal value. However, if each subgraph is chordal, then by Proposition 3.2.3, there exists a block decomposition of the subgraph that is tight, so that the value of the block decomposition is equal to that of the subgraph decomposition. □

For example, this shows that the value of any dual optimal subgraph decomposition using spanning trees of a pairwise graph is equal to the value of the optimal pairwise decomposition using just the edges of the graph. Similarly, the subgraph decomposition using all cycles of a pairwise graph is equivalent to the "triangle" relaxation, using all subsets of three nodes as blocks in the decomposition. This follows as each cycle can be "triangulated" (by adding a few pairwise edge to make the cycle chordal as illustrated in Figure 2.10) and decomposed into a set of triangles (corresponding the the cliques of the chordal version of the cycle). In fact, the same result holds if we only include the set of triangles that are contained within the maximal cliques of a chordal super-graph of the graph. This is because, in chordal graphs, every cycle can be decomposed into triangles within the maximal cliques of the chordal graph. This is the reason that we have focused our analysis mainly on block decompositions. However, that does not imply that there is no advantage to using subgraph decompositions, the complexity of solving the corresponding block decomposition is often much higher due to introducing many more degrees of freedom in the dual problem.

Ferromagnetic and Non-Frustrated Ising Models

In this section, we consider binary variable models $x_i \in \{-1, +1\} \equiv \{-, +\}$ with pairwise ferromagnetic interactions. This means that the potential can be expressed in the form

$$f(x) = \sum_i \theta_i x_i + \sum_{\{i,j\} \in \mathcal{G}} \theta_{ij} x_i x_j \tag{3.26}$$

where $\theta_{ij} > 0$ for all edges. Note that the "field" parameters θ_i defining the node potentials can be positive or negative. In physics, this is called the *ferromagnetic* Ising model. In fact, any binary model with pairwise interactions of the form

$$f(x) = \sum_i f_i(x_i) + \sum_{ij} f_{ij}(x_i, x_j) \tag{3.27}$$

is equivalent (up to an irrelevant additive constant) to an Ising model with parameters:

$$\theta_i = \frac{1}{2}\sum_x x_i f(x) = \frac{1}{2}\{[f_i(+) - f_i(-)] + \sum_j [f_{ij}(++) + f_{ij}(+-) - f_{ij}(-+) - f_{ij}(--)]\}$$

$$\theta_{ij} = \frac{1}{4}\sum_x x_i x_j f(x) = \frac{1}{4}\{f_{ij}(++) + f_{ij}(--) - f_{ij}(+-) - f_{ij}(-+)\}$$

Hence,

$$\sigma(f_{ij}) \triangleq f_{ij}(--) + f_{ij}(++) - f_{ij}(-+) - f_{ij}(+-)$$

is *invariant* with respect to reparameterizations of the model, and we therefore say that a general pairwise interaction $f_{ij}(x_i, x_j)$ is *ferromagnetic* if $\sigma(f_{ij}) > 0$. This condition is also called *submodular* in the graphical modeling literature [220].

It is known that this class of models is tractable by a number of methods including methods which convert MAP estimation in such models to a max-flow/min-cut problem [98, 115, 136],] roof-duality [103] and the TRMP method [134, 135, 211]. We now show that it is also solvable using the simplest version of Lagrangian relaxation, where we split the graph into a set of pairwise edges. Indeed, this version of LR is essentially equivalent to TRMP, which has also been shown to be tight for ferromagnetic models, but we find it instructive to provide another proof of strong duality from the perspective of LR, based on the characterization of strong duality as being equivalent to satisfiability.

Proposition 3.2.4 (ferromagnetic duality). *If there is a duality gap in the pairwise relaxation of the Ising model (or any other binary variable model with pairwise potentials), then there must exist at least one anti-ferromagnetic interaction, that is, an edge $\{i, j\} \in \mathcal{G}$ such that $\sigma(f_{ij}) < 0$. Hence, strong duality obtains in ferromagnetic models using pairwise relaxations.*

The proof may be found in Appendix B. It is based on Proposition 3.2.1 and a graphical method for checking satisfiability in the 2-SAT problem [7].

Non-Frustrated Ising Model This strong duality result is easily extended to the following generalization of the ferromagnetic model class. Let us say that a cycle is *frustrated* if it includes an odd number of anti-ferromagnetic edges. If there are no frustrated cycles then we say that the model is *non-frustrated*. This condition is equivalent to the existence of a partitioning of the vertex set $V = V_1 \cup V_2$ (where $V_1 \cap V_2 = \emptyset$) such that negating all variables in either subset (e.g., $x_i \rightarrow -x_i$ for all $i \in V_1$), which is equivalent to simply relabeling the nodal states (swapping + and − labels), results in a ferromagnetic model. Note that negating one endpoint of an edge reverses the edge's sign $\sigma(f_{ij}) \rightarrow -\sigma(f_{ij})$, but negating both endpoints preserves the sign. Thus, this condition is met if there exists V_1 and V_2 such that all "cut" edges between these two sets have $\sigma(f_{ij}) < 0$ and the remaining edges have $\sigma(f_{ij}) > 0$. Clearly, if there exists such a cut then it is unique. Thus, it is easy to check if a given model is non-frustrated. Simply select a spanning tree of the graph and then determine a sign flips that make

the tree ferromagnetic (this is always possible in trees, by a linear-time algorithm). Then, the model is non-frustrated if and only if the remaining edges are also made ferromagnetic by the same set of sign flips.

Corollary 3.2.3 (non-frustrated duality). *For binary models with pairwise interactions, a duality gap implies the existence of at least one frustrated cycle. Equivalently, strong duality holds in non-frustrated models.*

Proof. If the model is non-frustrated, then there exists a bi-partition of V into two disjoint sets V_+ and V_- such that by simply swapping the state values at all nodes in one of these sets (say V_-) we obtain a ferromagnetic model. That is, by re-defining the variables of the Ising model as $x'_v = x_v$ for all $v \in V_+$ and $x'_v = -x_v$ for all $v \in V_-$ (and making corresponding changes in the definitions of potentials) we obtain an equivalent model θ' that is ferromagnetic and for which strong duality holds. Clearly, the values of both the MAP problem and the dual problem are invariant to simply relabeling state values, so that strong duality must also hold in non-frustrated models. □

■ 3.3 A Statistical Physics Approach to Solving the Dual Problem

In this section we develop our approach to solution of the dual problem arising in these decomposition methods.

■ 3.3.1 Gibbsian Smoothing Technique

The dual function in (LR-$\theta^\dagger$) is given by:

$$g(\theta^\dagger) = \max_{x^\dagger} \left\{ \theta^\dagger \phi^\dagger(x^\dagger) \right\} \tag{3.28}$$

This is convex, piece-wise linear function. Hence, direct minimization of this function would require optimization methods for non-differentiable objectives, such as the sub-gradient method. We propose another approach that allows one to use methods for smooth optimization to solve the dual.

The starting point for our approach is the log-sum-exp "soft-max" function:

$$\boxed{\hat{g}_\beta(\theta^\dagger) \triangleq \beta^{-1} \log \sum_{x^\dagger} \exp\{\beta \theta^\dagger \phi^\dagger(x^\dagger)\}} \tag{3.29}$$

As shown by Proposition 2.5.1, this is a smooth, convex function and it holds that $\hat{g}_\beta(\theta^\dagger) \geq g(\theta^\dagger)$ for all $\theta^\dagger$ and $\beta \geq 0$. Moreover, the smoothed dual function $\hat{g}_\beta$ converges uniformly to g as $\beta \to \infty$.

This also gives our approach a connection to statistical physics, as this smoothed dual function is equal to the free-energy (log-partition function scaled by temperature) of the Gibbs distribution:

$$P(x^\dagger; \theta^\dagger, \beta) = \exp\left\{ \beta \left[(\theta^\dagger)^T \phi^\dagger(x^\dagger) - \hat{g}_\beta(\theta^\dagger) \right] \right\} \tag{3.30}$$

Thus, β^{-1} represent the temperature of the Gibbs distribution. As the temperature becomes small, this distribution puts more weight on those configurations $x^\dagger$ for which the energy is near the maximum value. Thus, this provides a natural way "smooth out" the dependence of the dual function over the set of near-optimal configurations, rather than just the maximum as in $g(\theta^\dagger)$.

In the block decomposition method, the smoothed dual function is tractable to compute, it is simply evaluated as:

$$\hat{g}_\beta(\theta^\dagger) = \sum_{E \in \mathcal{G}} \hat{g}_\beta^E(\theta^E) \tag{3.31}$$

where each block is computed as

$$\hat{g}_\beta^E(\theta^E) = \beta^{-1} \log \sum_{x^E} \exp\{\beta \theta^E \phi_{[E]}(x^E)\}. \tag{3.32}$$

Then, for a specified value of β, we can solve the following *smoothed dual problem* to obtain an approximation to the solution of the non-differentiable dual problem:

$$\boxed{\begin{array}{ll} \text{(Gibbs-}\theta^\dagger\text{)} & \begin{array}{ll} \text{minimize} & \hat{g}_\beta(\theta^\dagger) \\ \text{subject to} & \hat{D}^T\theta^\dagger = \theta \end{array} \end{array}}$$

Let the minimum value of this problem be denoted by $g^*(\beta)$. If we have chosen a minimal parameterization of the exponential family, as in the Boltzmann and Ising models, then this function is strictly convex and we obtain a unique solution $\theta^{\dagger *}(\beta)$. The smoothed dual problem can also be expressed in terms of Lagrange multipliers λ. By an abuse of notation, let $\boxed{\hat{g}_\beta(\lambda) \equiv \hat{g}_\beta(\tilde{\theta}^\dagger + C^T\lambda)}$. Then, the smoothed dual problem is equivalently expressed as:

$$\boxed{\begin{array}{ll} \text{(Gibbs-}\lambda\text{)} & \text{minimize } \hat{g}_\beta(\lambda) \end{array}}$$

Using the moment-generating property of the log-partition function, we show that the optimality of the decomposition $\theta^\dagger$ is equivalent a set of moment-matching conditions among duplicated variables and subsets of variables in $\mathcal{G}^\dagger$:

Proposition 3.3.1. *A decomposition $\theta^\dagger$ minimizes the smoothed dual function $\hat{g}(\theta^\dagger)$, over the subspace of valid decompositions of θ, if and only if it holds that*

$$C\, \eta_\beta^\dagger(\theta^\dagger) = 0 \tag{3.33}$$

where

$$\eta_\beta^\dagger(\theta^\dagger) \triangleq \sum_{x^\dagger} P(x^\dagger; \theta^\dagger, \beta) \phi^\dagger(x^\dagger) \tag{3.34}$$

That is, it must hold that the moments $\eta^\dagger$ in the decomposed model are consistent, such that, $\eta_\alpha^\dagger = \eta_\beta^\dagger$ for all pairs of features (α, β) that are duplicates of some common feature γ of the original model.

Proof. Minimizing $\hat{g}_\beta(\theta^\dagger)$ over valid decompositions is equivalent to minimizing the smooth, convex function $\hat{g}_\beta(\lambda)$ with respect to λ. Minimizing $\hat{g}_\beta(\lambda)$ is equivalent to solving the condition $\nabla \hat{g}_\beta(\lambda) = 0$. Using the chain rule and moment-generating property of the log-partition function, we obtain:

$$\nabla \hat{g}_\beta(\lambda) = C\Lambda_\beta^\dagger(\tilde{\theta}^\dagger + C^T\lambda)$$

where $\Lambda_\beta^\dagger(\theta^\dagger) \triangleq \mathbb{E}_{\beta\theta^\dagger}\{\phi^\dagger(x^\dagger)\}$ calculates the moments of the Gibbs distribution at temperature β^{-1}. Then, setting $\theta^\dagger = \tilde{\theta}^\dagger + C^T\lambda$ and $\eta^\dagger = \Lambda_\beta^\dagger(\theta^\dagger)$, we obtain the result that the gradient of $\hat{g}_\beta(\lambda)$ is zero if and only if the moments $C\eta^\dagger = 0$, that is, if and only the moments $\eta^\dagger$ are consistent. This is also equivalent to the condition that the gradient of $\hat{g}_\beta(\theta^\dagger)$ is orthogonal to the null-space of $\hat{D}^T$, which is the condition for a valid decomposition $\theta^\dagger$ to minimize $\hat{g}_\beta(\theta^\dagger)$ subject to the constraint $\hat{D}^T\theta^\dagger = \theta$. □

Because, for discrete graphical models, the moment parameters correspond to marginal specification of the graphical model, this moment consistency condition is also equivalent to the following marginal-matching conditions: For all blocks $A, B \in \mathcal{G}$ of the decomposition, we require that the probability distributions on these blocks must have consistent marginals with respect to the nodes $S = A \cap B$. That is, given the distributions:

$$\begin{aligned} P_\beta^A(x^A) &= \exp\{\beta\theta^A\phi^A(x^A) - \log Z_A\} \\ P_\beta^B(x^B) &= \exp\{\beta\theta^B\phi^B(x^B) - \log Z_B\} \end{aligned}$$

we must have that

$$\sum_{x_{A\setminus S}} P_\beta^A(x_S, x_{A\setminus S}) = \sum_{x_{B\setminus S}} P_\beta^B(x_S, x_{B\setminus S}) \tag{3.35}$$

for all x_S.

Temperature Annealing Method

To solve the dual problem, we actually solve a sequence of smoothed versions of the dual problem, using the soft-max approximation based on the Gibbs distribution for a sequence of temperature $\tau_k = \beta_k^{-1}$ approaching zero. This idea is illustrated in Figure 3.7(a).

The simplest approach to specify the reduction in temperature is to give a rate parameter $\rho \in (0, 1)$ and specified tolerance ϵ on the accuracy of the solution. Then, the overall procedure is structured as follows. Let $\tau_0 = 1.0$ and $\lambda_0 = 0$. For $k = 1, 2, \ldots$, solve the smoothed Lagrangian dual problem (Gibbs-λ), minimizing $\hat{g}_\beta(\lambda)$ with respect to λ for $\beta = (\rho^k\tau_0)^{-1}$, using a local descent method initialized from λ_{k-1}. Terminate the local descent method once all moment matching conditions are met within a tolerance of ϵ. There are two modifications of this procedure that sometimes help to significantly accelerate the overall rate of convergence. The first is to adapt the rate at which

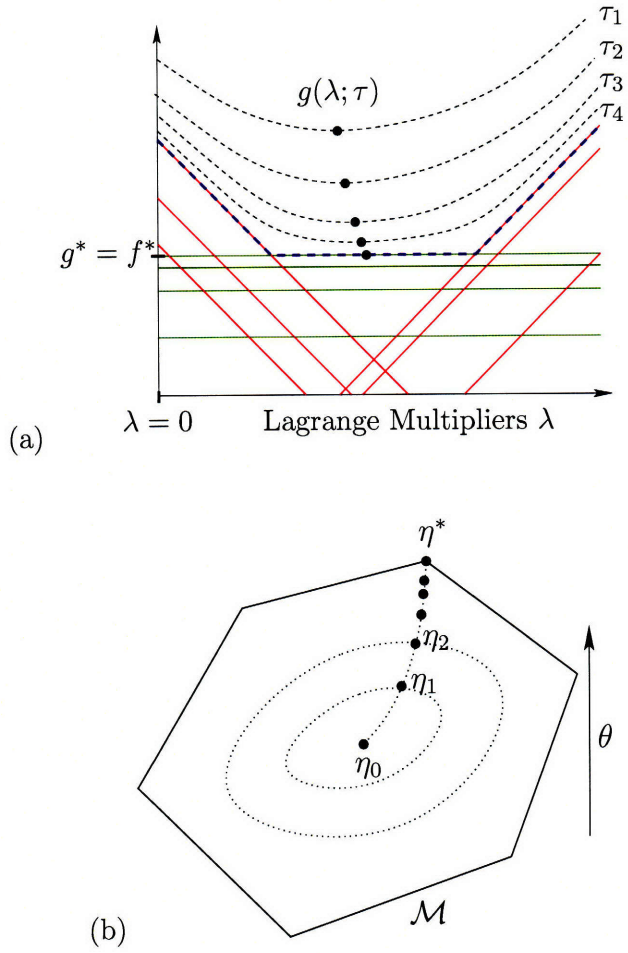

Figure 3.7. Illustration of our deterministic annealing method for solving the Lagrangian dual problem. (a) Our method solves a sequence of smoothed versions of the dual problem, each minimizing a smooth, convex upper-bound approximation to the non-differentiable dual function. In the LP interpretation of the dual problem, as finding the lowest-point of the epigraph of $g(\lambda)$, this is seen to be a kind of interior-point method, insofar as it generates a sequence of points within the epigraph that converges to a point on the boundary of the epigraph. (b) In the maximum-entropy interpretation of this method (see Section 3.3.2) this is seen to also be an interior point method for solving the LP relaxation of MAP estimation based on the pseudo-marginal polytope, where the entropy of the decomposed model serves as a barrier function of this polytope.

temperature is reduces. We do this by monitoring how many iterations it takes for the iterative local descent method to satisfy the moment-matching condition to within tolerance ϵ. If the number of iterations gets to large, then we increase ρ so as to more slowly decrease the temperature. The second, is to extrapolate the solution from the previous two temperatures to get a better initial guess for the optimal solution at the

next temperature. We do this be performing a line-search over

$$\lambda(t) = \lambda_{k-1} + t(\lambda_{k-1} - \lambda_{k-2}) \tag{3.36}$$

to determine the value of $t > 0$ that (approximately) minimizes $\hat{g}_{\beta_k}(\lambda(t))$, and then use this point to seed the local descent method. We continue this procedure until either (1) strong-duality is obtained, which is detected by checking if each block or subgraph of the decomposition has a unique MAP estimate, or (2) until the temperature τ_k is below some pre-specified minimum temperature $\tau_{\min}$. In the latter case, we takes this as an indication that there is probably a duality gap and we cannot recover the MAP estimates.

There are two advantages of this approach in comparison to trying to solve the dual problem directly. First, we may now use methods of smooth optimization to solve the dual problem, as opposed to using methods for non-smooth optimization such as the subgradient descent method or simplex method. In particular, coordinate-descent methods are guaranteed to converge to the global optimum of smooth, convex functions. This is in contract to non-smooth optimization for which it is known that the coordinate descent method may converge to a non-minimal point. In fact, this has been observed to be a difficulty of a number of algorithms developed for MAP estimation [134, 220]. In our approach to solve the smoothed dual problem (Gibbs-$\theta^\dagger$), developed in Section 3.3.3, we use a simple, efficient block-coordinate descent method to minimize the smoothed dual function. Second, we have observed that the smoothed dual problem is better conditioned at higher temperatures, that is, it is more easily solved using local descent methods (such as coordinate-descent) at higher temperatures. Thus, it is more efficient to obtain an initial solution at a high temperature and to use this to seed the local descent method at a lower temperature. Then, by gradually reducing the temperature while maintaining an optimal decomposition at each temperature, we efficiently generate a sequence of solutions converging to the optimal solution of the Lagrangian dual problem at zero temperature.

■ 3.3.2 Maximum Entropy Regularization

We now describe the maximum-entropy interpretation of our smoothed dual problem. We relate the Gibbsian smoothing method in the dual problem to a maximum-entropy regularization method in the LP-relaxation (LP-$\hat{\mathcal{M}}$). This regularized problem is:

$$\boxed{\begin{array}{ll} \text{(Gibbs-}\hat{\mathcal{M}}\text{)} & \begin{array}{ll} \text{maximize} & \theta^T\eta + \beta^{-1}\sum_{E\in\mathcal{G}} H(\eta_{[E]}) \\ \text{subject to} & \eta \in \hat{\mathcal{M}}(\mathcal{G}) \end{array} \end{array}}$$

This is essentially an entropy-regularized version of the LP-relaxation of MAP estimation based on the local-marginal polytope $\hat{\mathcal{M}}(\mathcal{G})$ based on $\mathcal{G}$, where the objective is modified by adding an entropy regularization term that favors more random distributions. Essentially, this serves as a barrier function on $\hat{\mathcal{M}}(\mathcal{G})$, as the gradient of entropy becomes infinite near the boundary of this set so as to prevent the solution from being

at the boundary of this set. This regularization term is actually a sum of block-wise entropies, and is therefore tractable to compute. We also note that this is similar to the problem of minimizing Gibbs free energy, with the entropy replaced by a sum of block entropies. We now demonstrate the following duality principle:

Proposition 3.3.2. *The smoothed dual problem (Gibbs-$\theta^\dagger$) and regularized problem (Gibbs-$\hat{\mathcal{M}}$) are dual problems and have equal values. Their optimal values $\theta^\dagger$ and η are related by $\eta_{[E]} = \Lambda_E(\beta\theta^E)$. Note that this implies consistency among the edge-wise potentials θ^E with respect to their moments.*

Proof. We show that the Lagrangian relaxation of (Gibbs-$\theta^\dagger$) has (Gibbs-$\hat{\mathcal{M}}$) as it dual problem. Introducing Lagrange multipliers for the constraints $\hat{D}^T\theta^\dagger = \theta$, we obtain the Lagrangian objective:

$$L(\theta^\dagger; \nu) = \hat{g}_\beta(\theta^\dagger) - \eta^T(\hat{D}^T\theta^\dagger - \theta) \tag{3.37}$$

The dual function is then

$$\begin{aligned}
\mathcal{F}(\eta) &= \min_{\theta^\dagger} \left\{ \hat{g}_\beta(\theta^\dagger) - \eta^T(\hat{D}^T\theta^\dagger - \theta) \right\} \\
&= \beta^{-1} \min_{\theta^\dagger} \left\{ \Phi^\dagger(\beta\theta^\dagger) - (\hat{D}\eta)^T(\beta\theta^\dagger) \right\} + \theta^T\eta \\
&= \beta^{-1} H^\dagger(\hat{D}\,\eta) + \theta^T\eta
\end{aligned}$$

In the last step, we have used the convex-duality between entropy and the log-partition function (see Section 2.3.2). This also requires that $\hat{D}\eta \in \mathcal{M}^\dagger(\mathcal{G})$, otherwise the dual function is $+\infty$. Thus, the dual problem is:

$$\begin{array}{ll} \text{maximize} & \theta^T\eta + \beta^{-1}H^\dagger(\hat{D}\,\eta) \\ \text{subject to} & \hat{D}\,\eta \in M^\dagger(\mathcal{G}) \end{array} \tag{3.38}$$

This may also be viewed as a linearly constrained optimization over $\eta^\dagger \in \mathcal{M}(\mathcal{G}^\dagger)$:

(Gibbs-$\mathcal{M}^\dagger$)	maximize	$(\tilde{\theta}^\dagger)^T\eta^\dagger + \beta^{-1}H^\dagger(\eta^\dagger)$
	subject to	$\eta^\dagger \in M(\mathcal{G}^\dagger)$
		$C\eta^\dagger = 0$

Here, $\tilde{\theta}$ may be taken to be any valid decomposition of θ. However, because of the block structure in $\mathcal{G}^\dagger$, the constraint $\hat{D}\eta \in \mathcal{M}^\dagger(\mathcal{G})$ is equivalent to $\eta \in \hat{\mathcal{M}}(\mathcal{G})$ and the entropy is equal to $H^\dagger(\hat{D}\eta) = \sum_{E\in\mathcal{G}} H_E(\eta_{[E]})$. Thus, this dual problem is equivalent to (Gibbs-$\hat{\mathcal{M}}$). Strong duality holds because (Gibbs-$\theta^\dagger$) is a convex optimization problem over an affine subspace. □

Interior-Point Interpretation of Temperature Annealing Method

In the temperature annealing method, the sequence of dual solutions λ_k, corresponding to dual optimal decompositions $\theta^\dagger_k$, each determine a point $\eta_k \in \hat{M}(\mathcal{G})$ by $(\eta_k)_{[E]} = \Lambda_E^{-1}(\theta^E(\theta^E_k))$. Using the above duality principle, each of these points solves the corresponding entropy-regularized LP for a specified value of β_k. Thus, our method is dually-related to an interior-point method for solving the LP relaxation of MAP estimation based on the marginal polytope $\hat{\mathcal{M}}(\mathcal{G})$, using the entropy $H^\dagger(D\eta)$ as a barrier function for the marginal polytope. However, our dual decomposition approach is *not* simply equivalent to any local descent method for solving each of these regularized LPs. This is because the mapping from $\theta^\dagger$ to η is only defined for optimal values of $\theta^\dagger$, for which the block-wise moments are all consistent on separate blocks. Rather, the iterative scaling method we develop to solve for the block decomposition is interpreted as an iterative projection algorithm for solving the *relaxed* representation of this LP, as in (Gibbs-$\mathcal{M}^\dagger$), based on $\mathcal{M}(\mathcal{G}^\dagger)$.

■ 3.3.3 Iterative Scaling Algorithm

Now that we have proposed a smooth variational approach to solution of the Lagrangian relaxation of the MAP estimation problem, we develop an iterative method to solve the smoothed dual problem (at a fixed temperature). The approach we develop is inspired by the iterative scaling method traditionally used to learn graphical models (the iterative proportional fitting procedure is a special case of the general iterative scaling algorithm). However, in the zero-temperature limit, these ideas turn out to have close ties to the max-sum diffusion algorithm and Kolmogorov's sequential variant of the TRMP algorithm [134, 135].

Algorithm Specification

In this section we specify the algorithm used to solve (Gibbs-$\theta^\dagger$). Interpretations of this algorithm are given in the following sub-sections. We begin with an MRF defined on $\mathcal{G}^\dagger$ with potentials functions given by $f_E(x_E) = (\theta^E)^T \phi_{[E]}(x_E)$ for each blocks $E \in \mathcal{G}$. We then specify a collection of *update sets* $S(k) \subset V$ for $k = 1, \ldots, m$, which represent intersections of two or more blocks. It is sufficient to define a single update for every pair of over-lapping blocks, with S given by the intersection of these two blocks. However, we also allow for updates on smaller sets which can occur as the intersection of more than two blocks (this may improve the rate of convergence of the algorithm). Thus, each update set lies in the intersection of two or more blocks $E \in \mathcal{G}$ of the original graphical model. Let $\mathcal{G}(S)$ denote the set of all blocks $E \in \mathcal{G}$ which contain S.

Then, the following iterative algorithm enforces consistency of the marginal distributions among the various blocks:

Iterative Scaling Algorithm For $t = 1, \ldots$ until convergence:
For $k = 1, \ldots, m$:

1. Let $S = S(k)$. For each $E \in \mathcal{G}(S)$ compute:

$$\hat{f}_E(x_S) = \beta^{-1} \log \sum_{x_{E \setminus S}} \exp\{\beta f_E(x_E, x_{E \setminus S})\}$$

2. Next, compute the average of these functions:

$$\bar{f}(x_S) = \frac{1}{|\mathcal{G}(S)|} \sum_{E \in \mathcal{G}(S)} \hat{f}_E(x_S)$$

3. For each $E \in \mathcal{G}(S)$, perform the updates:

$$f_E(x_E) \leftarrow f_E(x_E) + (\bar{f}(x_S) - \hat{f}_E(x_S)))$$

This may also be described in terms of the exponential family parameters:

1. For each $E \in \mathcal{G}(S)$ compute:

$$\hat{\theta}^{E,S} = \Pi_S(\theta^E)$$

where $\Pi_S(\theta^E) \triangleq \Lambda_S^{-1}(\Lambda(\theta^E)_{[S]})$ denotes the operation of marginalization in the θ-parameterization. It is defined such that $\hat{f}_E(x_S) \propto \exp\{(\Pi_S(\theta^E))^T \phi_{[S]}(x_S)\}$.

2. Next, compute the average of these parameters:

$$\bar{\theta}^S = \frac{1}{|\mathcal{G}(S)|} \sum_{E \in \mathcal{G}(S)} \hat{\theta}^{E,S}$$

3. For each $E \in \mathcal{G}(S)$, perform the parameter update

$$(\theta^E)_{[S]} \leftarrow (\theta^E)_{[S]} + (\bar{\theta}^S - \hat{\theta}^{E,S})$$

We continue this procedure until it convergences (at a fixed temperature), that is, until all copies of each update set have the same marginal distribution (to within some specified tolerance). Then, the resulting block decomposition is a solution to (Gibbs-$\theta^\dagger$) and the resulting set of (consistent) moments provide the solution to (Gibbs-$\hat{\mathcal{M}}$). We can use this procedure as a subroutine in the temperature annealing method (Section 3.3.1), to re-optimize the block decomposition at each step of that procedure. This then provides an interior point approach to solve (LR-$\theta^\dagger$). Likewise, the sequence of consistent moments generated by this procedure then correspond to an interior-point approach to solve of (LP-$\hat{\mathcal{M}}$).

There are two key properties of the algorithm. First, it is equivalent to performing reparameterizations of the original model, and thus stays within the subspace of valid block decompositions. That is, it preserves the potential on the original graphical model $\mathcal{G}$, obtained by summing edge potentials of the relaxed model on $\mathcal{G}^\dagger$ over all copies of an edge. For this reason, it is also equivalent to some update in the Lagrange multiplier representation of this space of valid decompositions. Second, each update step forces the marginal distributions on all blocks $E \in \mathcal{G}(S)$ to have the some marginal distribution on S.

Block Coordinate Descent Interpretation

First, we show that this procedure is equivalent to a block-coordinate descent procedure for minimization of the smoothed dual function $\hat{g}_\beta(\lambda)$. Because $\hat{g}_\beta(\lambda)$ is equal to the log-partition function of the MRF defined on $\mathcal{G}^\dagger$ (multiplied by the temperature) evaluated at $\theta^\dagger(\lambda) = \tilde{\theta}^\dagger + C^T\lambda$, we have by the moment-generating property of the log-partition function:

$$\begin{aligned}\frac{\partial \hat{g}_\beta(\lambda)}{\lambda_{\alpha,\beta}} &= \sum_\gamma \frac{\partial \hat{g}_\beta(\lambda)}{\partial \theta^\dagger_\gamma(\lambda)} \frac{\partial \theta^\dagger_\gamma(\lambda)}{\partial \lambda_{\alpha,\beta}} \\ &= \sum_\gamma \eta_\gamma(\theta^\dagger(\lambda)) \frac{\partial \theta^\dagger_\gamma(\lambda)}{\lambda_{\alpha,\beta}} \\ &= \eta_\alpha(\theta^\dagger(\lambda)) - \eta_\beta(\theta^\dagger(\lambda)) \end{aligned} \tag{3.39}$$

Here, we have used the chain rule and the fact that each Lagrange multiplier $\lambda_{\alpha,\beta}$ is added to $\tilde{\theta}^\dagger_\alpha$ and subtracted from $\tilde{\theta}^\dagger_\beta$. Thus, the condition for minimization of $\hat{g}_\beta(\lambda)$ over all $\lambda_{\alpha,\beta}$ between any two copies of an update set S, is that the sufficient statistics defined within each of these copies are consistent. This is equivalent to requiring that the marginal distributions on these subsets are equal. Thus, each update step of the iterative scaling algorithm is equivalent to one step of a block coordinate descent method to minimize $\hat{g}_\beta(\lambda)$ with respect to the subset of the Lagrange multipliers associated to equality constraints between multiple copies of the features defined on x_S in the decomposition. It also can be viewed as performing coordinate descent within the subspace of valid decompositions of θ. Block coordinate descent is guaranteed to converge to the global minimum of a smooth, convex function [24]. Thus the iterative scaling algorithm specified above is guaranteed to converge to the minimum of the smoothed dual function.

Iterative Projection Interpretation

We also remark that each step of the iterative scaling algorithm that we specified to minimize $\hat{g}_\beta(\theta^\dagger)$ in (Gibbs-$\theta^\dagger$) may also be viewed as an information projection step in the dual problem (Gibbs-$\mathcal{M}^\dagger$). That is, given the previous set of moments $\eta^\dagger$, the next set of moments $\nu^\dagger$ are obtained as the solution to the information projection problem:

$$\begin{aligned} &\text{minimize} && d_H(\nu^\dagger, \eta^\dagger) \\ &\text{subject to} && \nu^\dagger \in \mathcal{M}^\dagger_S \end{aligned}$$

where $\mathcal{M}^\dagger_S$ represents the subspace of moments within $\mathcal{M}(\mathcal{G}^\dagger)$ which are consistent among all blocks $E \in \mathcal{G}(S)$ that contain the update set S. Thus, by repeatedly iterating over all updates sets and performing these projection steps, the moments $\eta^\dagger$ converge to the information projection onto the intersection of all these sets $\cap_k \mathcal{M}^\dagger_{S(k)}$, which represents the subspace of consistent moments, that is, the set of all moments

$\eta^\dagger \in \mathcal{M}(\mathcal{G}^\dagger)$ that satisfy $C\eta^\dagger = 0$. A subtle point here is that the iterative scaling algorithm also has the property that each update stays within the subspace of valid decompositions. Because of this, the overall algorithm converges to the unique point in the intersection of these two subspaces, that is, it converges to a point that is both a valid decomposition (satisfying $\hat{D}^T\theta^\dagger = \theta$) and is consistent (satisfying $C\eta^\dagger = 0$). Then, $\theta^\dagger$ is the optimal solution of (Gibbs-$\theta^\dagger$) and $\eta^\dagger$ is the optimal solutions of (Gibbs-$\mathcal{M}^\dagger$).

■ 3.4 Heuristics for Handling Problems with a Duality Gap

In this section we present some techniques for dealing with problems where simple relaxations lead to a duality gap. This first method, which we refer to as low-temperature estimation, is aimed at providing approximate, near-optimal solutions of the MAP estimation problem. The second method is aimed at adaptively enhancing the block decomposition by including higher-order structure in the dual problem, that is, additional subgraphs or blocks.

■ 3.4.1 Low-Temperature Estimation for Approximate MAP Estimation

As we have discussed, when there is a duality gap in the Lagrangian relaxation method, at least some nodes of the graph must exhibit ties in the optimal block decomposition, that is, the sets $\hat{X}^E = \arg\max f_E(x^E)$ contain multiple solutions. For a dual optimal decomposition, these sets must be pairwise satisfiable, but are not jointly satisfiable in the case of a duality gap. If these sets each have a unique solution, then they must be jointly satisfiable and there is no duality gap. Moreover, this is the typical outcome when there is no duality gap. In relation to the optimal solution $\hat{\eta}$ of the LP relaxation (LP-$\hat{\mathcal{M}}$), a duality gap in the block decomposition over $\mathcal{G}$ corresponds to fractional probabilities in the marginal distributions of some or all nodes in $\hat{\eta}$. In fact, these also exhibit at least some ties if there is a gap, such that the local node-wise marginal MAP estimates are non-unique. In Section 3.5, we show a number of examples where *all* marginal node probabilities are tied in this way, such that one cannot glean any useful information as to the MAP estimate by inspecting these node marginals.

In our temperature annealing method, however, we do not actually solve the dual problem directly, but rather solve a sequence of smoothed low-temperature problems that become equivalent to the dual problem in the limit of zero temperature. This produces a sequence of decompositions $\theta^\dagger_k$ and corresponding moments $\hat{\eta}_k$. We can use the low-temperature node-wise marginals specified by $\hat{\eta}_k$ to break ties that occur in the zero-temperature solution $\hat{\eta}$. To provide some motivation for this idea, consider again the case of a duality gap in the dual problem as illustrated in Figure 3.5(a). The zero-temperature solution places all of its weight on just those two inconsistent configurations of $x^\dagger$ corresponding to the two sloped lines that determine the minimum of the dual function. Using the low-temperature method, we also put some weight on those near-optimal configurations, such as the flat line in the figure that corresponds to the optimal MAP estimate. Thus, we might hope to detect the MAP estimate using the

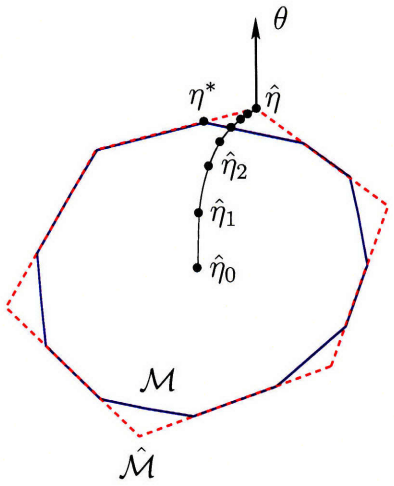

Figure 3.8. Illustration of the idea behind the low-temperature estimation method for approximating the MAP estimate when there is a duality gap. The sequence of solutions $\hat{\eta}_k$, of the entropy-regularized problem, converges to the LP solution $\hat{\eta}$ over $\hat{\mathcal{M}}(\mathcal{G})$ as the temperature approaches zero. At some point, this solution crosses the boundary of the marginal polytope $\mathcal{M}(\mathcal{G})$ and, hence, may also pass near the vertex η^* corresponding to the correct solution of the MAP estimation problem formulated as an LP over $\mathcal{M}(\mathcal{G})$.

low-temperature marginals. However, this picture is complicated by the fact that there may be many other inconsistent solutions between the minimum of the dual function and the MAP estimate. Also, if the temperature is made large enough so that the MAP estimate can compete with these other inconsistent solutions, then other near-optimal consistent configurations also influence the estimate.

Let us give another argument to help motivate this idea. We have shown in Section 3.3.2 that the dual interpretation of our smoothed dual problem (Gibbs-$\theta^\dagger$) is equivalent to an entropy-regularized version of (LP-$\hat{\mathcal{M}}$), the LP-relaxation of MAP estimation based on the pseudo-marginal polytope $\hat{\mathcal{M}}(\mathcal{G})$. As is well-known [211], the pseudo-marginal polytope gives an outer-bound approximation to the marginal polytope $\mathcal{M}(\mathcal{G})$. In the temperature annealing method, the set of solutions $\{\hat{\eta}_k\}$ define a set of points along a curve of solutions that begins at the maximum-entropy distribution $\hat{\eta}_0$ (because, at very high temperatures, the entropy term dominates over the linear term), which is located at the center of the marginal polytope (η_0 is just the center of mass of the vertices of $\mathcal{M}(\mathcal{G})$) to the final (inconsistent) solution $\hat{\eta}$ that is a vertex of the pseudo-marginal polytope. This is illustrated in Figure 3.8. Clearly, there is come critical temperature at which this curve crosses the boundary of the marginal polytope. It seems natural to speculate that this point may be close to the optimal solution η^* of the LP defined over $\mathcal{M}(\mathcal{G})$, which corresponds to the optimal MAP estimate (if it is unique). Then, using these low-temperature marginals to estimate each variable separately, we may be able to recover the MAP estimate or an estimate that is close to the MAP estimate. These

observations motivate the following heuristic method to obtain approximate solutions of the MAP estimation problem in cases where there is a duality gap.

Low-Temperature Estimation Method

For $k = 1, 2, 3, \ldots$:

1. Solve the smoothed dual problem (Gibbs-$\theta^\dagger$) at temperature β_k^{-1}, to obtain the decomposition $\theta_k^\dagger = (\theta_k^E, E \in \mathcal{G})$. This aspect of the algorithm is identical to our earlier specification for the temperature annealing method. We use the iterative scaling algorithm (Section 3.3.3).

2. Compute the corresponding moments $\hat{\eta}_k \in \hat{\mathcal{M}}(\mathcal{G})$, defined by $(\hat{\eta}_k)_{[E]} = \Lambda_E^{-1}(\beta_k \theta_k^E)$ (note that these edge-wise moment calculations are consistent for the optimal decomposition in the smoothed dual problem).

3. These moments determine the node-wise marginal distributions $\hat{P}_\beta(x_v)$ for all $v \in V$. Generate an estimate $\hat{x}_k$ to maximize each node's marginal probability, that is, let $(\hat{x}_k)_v = \arg\max \hat{P}_\beta(x_v)$ for all $v \in V$.

4. Try to improve this estimate using a greedy search algorithm, such as the iterated conditional modes (ICM) method [28]. Once this is done, evaluate $f(x)$ for this final improved estimate.

We continue this procedure until either strong duality is detected, so that all edges have a unique MAP estimate, or until the temperature becomes very small. In the latter case, we select the best estimate, that is, the one that gave the largest value of $f(x)$, among all of those that were generated by the above procedure. One could also randomize this procedure, by sampling from node marginals rather than simply taking the maximum. However, we have found good performance using the deterministic version of the method (see Section 3.5).

■ 3.4.2 Adaptive Methods to Enhance the Formulation

We also consider methods to improve the dual formulation by adaptively selecting additional blocks to include in the decomposition so as to reduce the value of the dual problem. This work is closely related to recent work of David Sontag [198] and earlier work of Barahona [13] in the context of Ising models.

The General Idea

We approach this from the point of view of strong duality being equivalent to joint satisfiability of the block-wise optimality conditions: $\hat{x}_E \in \hat{X}^E$ for all E where $\hat{X}^E = \arg\max f^E$ in an optimal block decomposition. This suggests the following general strategy for enhancing the block decomposition.

Start with some initial set of blocks specified by $\mathcal{G}^{(0)}$. Then, iteratively enhance the decomposition method as follows. For $t = 1, 2, 3, \ldots$:

1. Solve the dual problem using the current collection of blocks $\mathcal{G}^{(t)}$. Determine the edge-wise MAP sets $\hat{X}^E = \arg\max f^E$ in the optimal decomposition. If strong duality is detected, by virtue of each subgraph having a unique MAP estimate, then STOP. Then, there is no duality gap and we have solved the MAP estimation problem.

2. Otherwise, it is likely that there is a duality gap so that the collection of sets $\{\hat{X}^E\}$ is not jointly satisfiable. Suppose that we can identify a *minimal inconsistent subgraph*, that is, a subgraph $\mathcal{S} \subset \mathcal{G}^{(t)}$ such that the subset of constraints, $x_E \in \hat{X}^E$ for all $E \in \mathcal{G}$, is not satisfiable and any proper subgraph of $\mathcal{S}$ is satisfiable.

3. Given such an inconsistent subgraph, we then identify a chordal graph that covers all of its edges and then add the set of maximal cliques in the chordal graph as blocks in the next decomposition: $\mathcal{G}^{(t+1)} = \mathcal{G}^{(t)} \cup \mathcal{C}(\mathcal{S})$. Then, return to Step 1.

Including a chordal decomposition of the detected inconsistent subgraph ensures that this subgraph is satisfiable in the next iteration of the procedure and that the duality gap is reduced at each step. As long as there is a duality gap there must be another inconsistent subgraph such that, in principle, such a procedure would always eventually lead to strong duality.

This conveys the basic idea. However, there are fundamental difficulties with this approach. First, it it in general intractable to determine if the collection of edge-wise optimality constraints is satisfiable. Hence, finding a minimal subset of inconsistent constraints must also be intractable in general. However, in the case of pairwise relaxations of binary variable models, checking for strong duality can be reduced to a 2-SAT problem, for which there is an efficient solutions [7]. We use this method to prove strong duality of pairwise relaxations of ferromagnetic model in Appendix B. Although this method is only sufficient to check for satisfiability of all pairwise edges, this is a necessary condition for satisfiability of all blocks and therefore provides a heuristic for detecting inconsistent subgraphs among just the blocks of size two. This provides a tractable approach to identify one class of inconsistent subgraphs. However, in this method, it can happen that no inconsistency is found even though there is a duality gap. Another difficulty is that the inconsistent subgraph that we discover may not be thin. In that case, it is not tractable to further reduce the duality gap using this method. Of course, this is to be expected because the general problem of MAP estimation is indeed intractable.

Inconsistent Cycles of the Ising Model

Now, we develop a simpler version of this general idea in the context of the Ising model. Here, we only check for one type of inconsistent subgraph—*inconsistent cycles*. The advantage of this approach is that such subgraphs are both easy to detect and

to incorporate in the block decomposition method. In fact, because cycles are tree-width-two graphs, it is alway possible to decompose cycles into a set of blocks of size three. Thus, the following method can be implemented using only blocks of size three (or smaller, i.e. pairs of nodes).

This consistency check only involves constraints on pairwise edges of the graph. To obtain these, it is convenient to assume that every pair of nodes which is included in any block of $\mathcal{G}$ is itself a block of $\mathcal{G}$. We focus on two types of pairwise constraints for which inconsistencies are easily detected:[5] (strictly) ferromagnetic $\hat{X}^{ij} = \{++, --\}$ and (strictly) anti-ferromagnetic $\hat{X}^{ij} = \{+-, -+\}$. To check for joint satisfiability among the subset of ferromagnetic and anti-ferromagnetic pairwise constraints, we simply define a pairwise graph $\mathcal{G}_{\pm}$ with edges corresponding to the set of all ferromagnetic and anti-ferromagnetic constraints, and assign edges weights of $s_{ij} = +1$ for ferromagnetic constraints and $s_{ij} = -1$ for anti-ferromagnetic constraints. Then we check for cycles of this graph where the product of edge weights is -1. A simple linear-time algorithm can be used to check if there are any inconsistent cycles. Just take any spanning tree of the graph (or of each connected component of the graph) and assign the node variables consistently with respect to this spanning tree (which is always possible). Then check if any of the missing edges are not satisfied. If all edge constraints are satisfied, then this $\hat{x}$ is an optimal MAP estimate. Otherwise, any one of the violated edge constraints determines an inconsistent cycle using that edge and the path between its endpoints in the spanning tree. However, this does not necessarily produce the *smallest* inconsistent cycles of the graph.

To identify the set of shortest inconsistent cycles, we propose the following simple method using sparse matrix multiplication. Let S denote the *signed* adjacency matrix of the graph, in which those elements corresponding to edges are set to $+1$ or -1 according to the sign of the edge s_{ij} and the remaining elements are zero. Then, we compute

$$N_i^{(\ell)} = \frac{1}{2}(|S|^l - S^l)_{i,i} \tag{3.40}$$

for all nodes i and $\ell \in \{1, \ldots, n\}$. Here, $|S|$ is the usual (unsigned) adjacency matrix of the graph. We note that $(|S|^l)_{i,i}$ counts the number of closed walks of length ℓ that begin and end at node i. Let us say that a walk is consistent if the product of edge weights along the walk is $+1$, and is inconsistent if the product is -1. Thus, $(S^\ell)_{i,i}$ equals the number of consistent closed walks at node i, minus the number of inconsistent ones. Therefore, $N_i^{(\ell)}$ is the number of inconsistent closed walks of length ℓ at node i. Thus, by computing powers of $|S|$ and S, we can detect the shortest inconsistent cycles. To check for inconsistent cycles up to length ℓ requires $\mathcal{O}(\ell n^2)$ computation and $\mathcal{O}(n^2)$ memory, using sparse matrix multiplication to compute matrix powers.

To classify edges as ferromagnetic or anti-ferromagnetic, one would have to have an exact solution of the dual problem. To allow for numerical imprecision of the solution

[5] In the case of zero-field planar Ising models, it is sufficient to check just this subset of constraints. Moreover, we are usually able to solve other non-zero field and non-planar problems using just these constraints.

and that we actually solve a low-temperature approximation to the dual problem, we detect such edges by computing the edge-wise correlations:

$$\tilde{s}_{ij} \triangleq \eta_{ij} - \eta_i \eta_j$$

where $\eta_{ij} \triangleq \mathbb{E}\{x_i x_j\} = P_{ij}(++) + P_{ij}(--) - P_{ij}(+-) - P_{ij}(-+)$ and $\eta_i \triangleq \mathbb{E}\{x_i\} = P_i(+) - P_i(-)$ (computed using the Gibbs distribution). Note that $|\tilde{s}_{ij}| \leq 1$ for binary ± 1 variables. In the zero-temperature limit, these correlations $\tilde{s}_{ij}$ converge to either -1 or $+1$ for edges with (respectively) anti-ferromagnetic or ferromagnetic MAP sets in the zero-temperature solution. In the zero-field planar Ising model, all the other edges must have correlations $\tilde{s}_{ij}$ that converge to zero. This happens when the pairwise distribution is such that x_i and x_j are linearly independent.[6] In the general Ising model, other fractional values can also occur, but it is just the ones converging to -1 or $+1$ that we use to check for inconsistent cycles. Hence, we construct the matrix S by solving the smoothed dual problem (Gibbs-$\theta^\dagger$) at some low temperature and then define $\mathcal{G}_\pm$ and S based on the edges where $\tilde{s}_{ij}$ is close to either $+1$ or -1 (rounding these to ± 1 in S), and then checking for inconsistent cycles of this signed graph.

It must be emphasized that this procedure is not guaranteed to always lead to strong duality in general Ising models. However, in our experiments involving Ising models thus far, we have always been able to obtain the MAP estimate. There is reason to believe that this should always hold in the class of zero-field planar Ising models. That is consistent with results of Barahona *et al* [12, 13] and related recent work of Sontag *et al* [198]. However, these methods are formulated from a very different perspective. They consider cutting-plane methods to tighten the LP relaxation of MAP estimation and are based on a certain set of "cycle inequalities" that one may use to test if a point η is outside of the marginal polytope. Hence, these methods do not approach the problem from the point of a constraint satisfaction problem that characterizes strong duality in the block decomposition method. We think this is the interesting aspect of our approach, and the fact that it seems to provide a much simpler algorithm to decrease the duality gap in the dual problem. However, both approaches do involve checking for inconsistent cycles of the graph, although in different senses.

■ 3.5 Experimental Demonstrations

In this section we present an experimental study of the performance of Lagrangian relaxation methods on several model problems. We solve for the MAP estimate of random-field Ising models defined on 2-D lattices. We consider both ferromagnetic

[6] In the zero-temperature distributions, there are two ways this can happen. First, if the pairwise distribution becomes uniform, corresponding to the completely unresolved MAP set $\hat{X}^{ij} = \{++, --, +-, -+\}$. Second, if either node is resolved, such that $\eta_i = \pm 1$, which corresponds to MAP sets such as $\hat{X}^{ij} = \{++, +-\}$ or $\hat{X}^{ij} = \{+-\}$. Strictly speaking, the second possibility should not occur in zero-field models because $f(x) = f(-x)$. However, when we solve zero-field problems we actually set one of the field terms so as select one of the two optimal solutions.

models and disordered "spin glass" models (including a random mixture of both ferromagnetic and anti-ferromagnetic bonds so as to produce frustration effects). Also, both planar and non-planar graphs are investigated.

■ 3.5.1 Ferromagnetic Ising Model

First, we consider the ferromagnetic Ising model, which has binary variables $x_i \in \{-1, +1\}$ for all $i \in V$. We begin with a simple 12×12 grid model with nearest-neighbor bonds $\theta_{ij} = 0.2$ and random external field with independent, identically distributed $\theta_i \sim \mathcal{N}(0, 1)$, as seen in Figure 3.9. For a particular realization of the random field, we solve for the MAP estimate of this model using the pairwise decomposition method. Note that the dual value and the primal value of the estimates both converge to the correct MAP value (computed using junction trees in this small example). Also note that all the marginals converge to zero or one in this case, so that it is trivial to obtain the MAP estimate. Next, we consider a larger 50×50 example shown in Figure 3.10. This field is too large to be solved using the junction tree method, as the complexity of exact recursive inference grows as 2^w in an $w \times w$ grid.[7] Because there is a unique MAP estimate in the relaxed model, we can certify that this is indeed the optimal MAP estimate.

As expected, we always achieve strong duality and recover the correct MAP estimate in ferromagnetic models. Moreover, the iterative LR algorithm converges reasonably quickly in these examples. The computational complexity of each iteration scales linearly with the size of the field, but the number of iterations appears to grow slowly with the size of the field. Typically, the total number of iterations of the marginal matching algorithm ranges between several hundred to a thousand iterations. Using our MATLAB code for LR (which is far from optimal in regards to run-time efficiency), the 12×12 case converges in under 10 minutes whereas the 50×50 case takes 3-4 hours. Using a more efficient compiled C code, these examples would require substantially less run-time and it should become feasible to solve very large fields by this method.

■ 3.5.2 Disordered Ising Model

Next, we consider disordered Ising model where the edge parameters are randomized, allowing both ferromagnetic and anti-ferromagnetic bonds, so as to produce frustration effects. The bond parameters are independently and randomly chosen according to $\theta_{ij} = t_{ij} + w_{ij}$, where $t_{ij} = \pm 1$ (with equal probability) and $w_{ij} \sim \mathcal{N}(0, \sigma^2)$ is normally distributed with standard deviation $\sigma = 0.01$.[8] We now set the node parameters to a

[7] Using junction tree methods, we have been able to solve grids of sizes up to about 20×20 and would estimate that 40×40 is about the largest grid one could solve via junction trees using current technology.

[8] A small Gaussian perturbation was added to the bond strengths so that the optimal MAP estimate is (almost surely) unique in non-zero fields and is unique to within a global negation of all signs in zero-field. This may be viewed as a simple method to randomly select one of the MAP estimates of the integral model with ± 1 bonds.

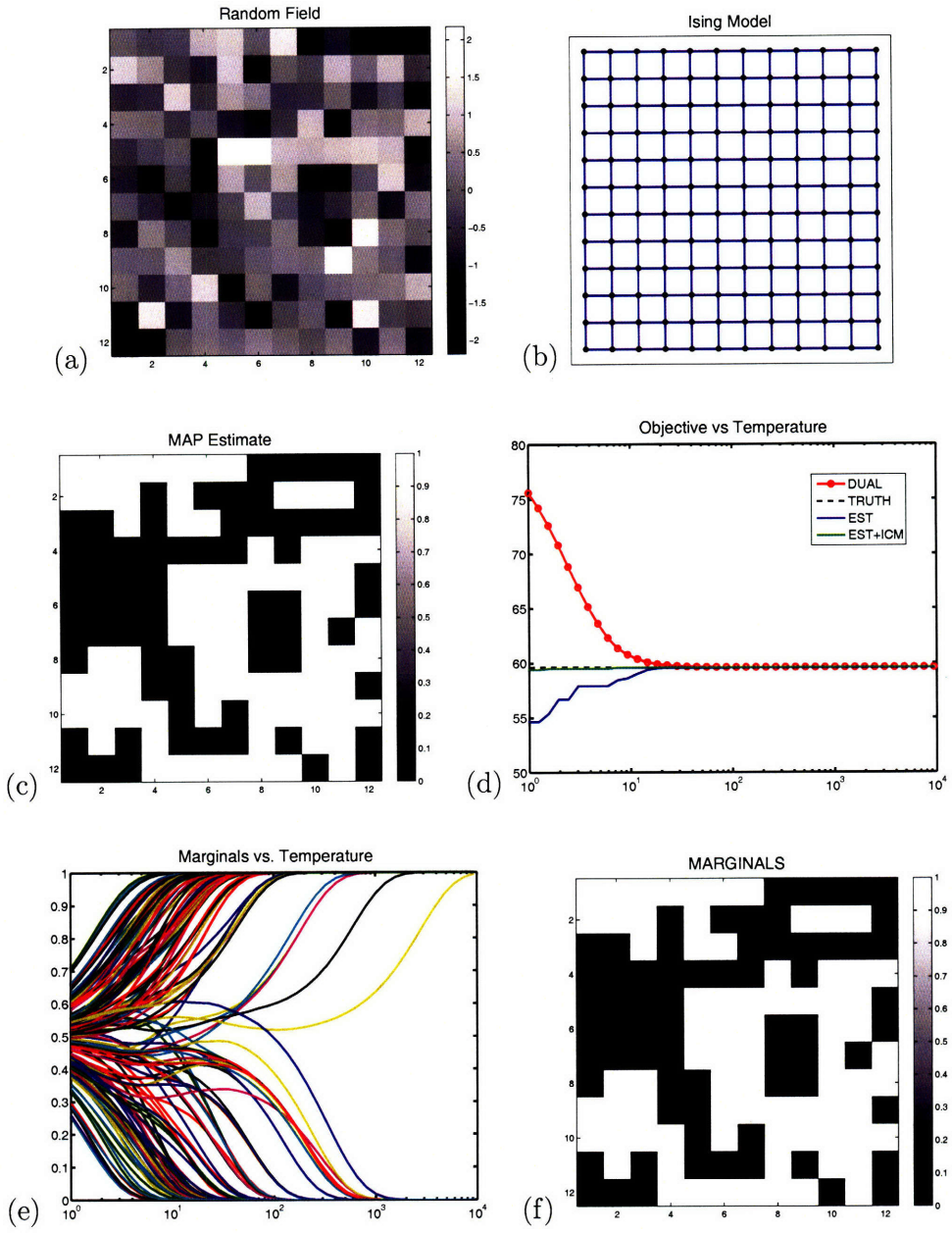

Figure 3.9. Results of applying pairwise LR to 12×12 ferromagnetic model. (a) the random-field parameters, (b) the 12×12 nearest-neighbor grid, (c) the correct MAP estimates computed using a junction tree method, (d) values of the dual function and low-temperature estimates as a function of temperature (compared to the MAP value denoted by the dashed horizontal line), (e) convergence of the marginal probabilities (or $x_i = +1$ at each node i) as a function of temperature, (f) an image of the final zero-temperature marginal probabilities, which is identical to the correct MAP estimate.

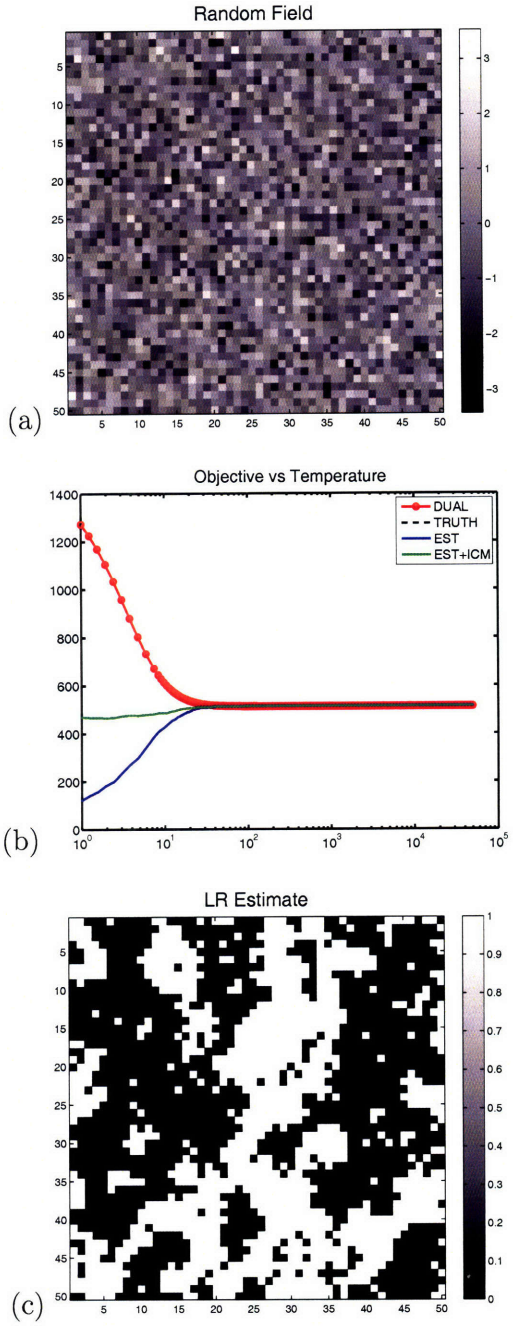

Figure 3.10. Results of applying pairwise LR to a 50×50 ferromagnetic model with a random field. (a) image of the random field, (b) values of the dual function and low-temperature estimates as a function of temperature (note that these become equal as the temperature approaches zero, certifying that there is no duality gap), (c) The correct MAP estimate obtained by LR.

constant external field $\theta_i = h$. For a particular realization of the random bonds, we apply LR to this model for several values of the external field strength. For zero and weak fields, we expect the MAP estimate to exhibit disorder, being a random mixture of small, disconnected + and − regions. As the field strength is increased, the MAP estimate transitions to a more ordered state with most nodes included in a global + and a few smaller − regions that are disconnected from one another.

Disordered Planar Model

We begin by considering the 12×12 square lattice with bonds configured as seen in Figure 3.11. Then, we solve the LR dual problem using the pairwise decomposition for field values $h = 0.0, 0.1, 0.5, 1.0, 2.0$.[9] For $h = 0$, we find that the pairwise decomposition exhibits a large duality gap as seen in Figure 3.11 and we do not recover the MAP estimate. However, we still obtain a near-optimal solution (that is, an estimate x for which $f(x)$ is close to the maximum f^*) using the low-temperature estimation heuristic. However, this estimate differs substantially from the actual MAP estimate. Next, we repeat this experiment for increasing values of the field. The outcome of these experiments is shown in Figure 3.12. As the field strength is increased, the duality gap is gradually reduced and we eventually obtain strong duality with field $h = 2.0$. Also, in the non-zero field examples with a duality gap, we again obtain estimates that have near-optimal values of f but are now closer to the actual MAP estimate, as measured by Hamming distance (the number of binary variables for which our estimate differs from the optimal MAP estimate).

Next, we try to reduce the duality gaps seen in this example by using the block decomposition method based on the faces of this planar graph (see Section 2.5.3), corresponding to the "squares" of the lattice. The results for $h \in \{0.0, 0.1, 0.5\}$ are shown in Figure 3.13. In the first two cases ($h = 0.0, 0.1$), we see that the duality gap is substantially reduced but is not completely eliminated. However, the similarity between the best estimate obtained from the low-temperature marginals is now more similar to the actual MAP estimate than in the corresponding pairwise decompositions. With field $h = 0.5$, we have eliminated the duality gap and obtained the exact MAP estimate. Thus, performing LR on the faces of the graph allows us to resolve the MAP estimate for lower values of the field strength than we were able to resolve using the pairwise decomposition.

Disordered Non-Planar Model

Now we consider a more challenging example using a non-planar graph. We use the "crossed-bond" Ising model, where each node of the square lattice is connected to its *eight* nearest neighbors. Again, we randomly set the bond strengths as before and now set a constant field of $h = 0.3$. First, we again try the pairwise decomposition. The

[9] In the zero-field case, we actually set the field at a single node to 1.0 in order to disambiguate the two MAP estimates of the model.

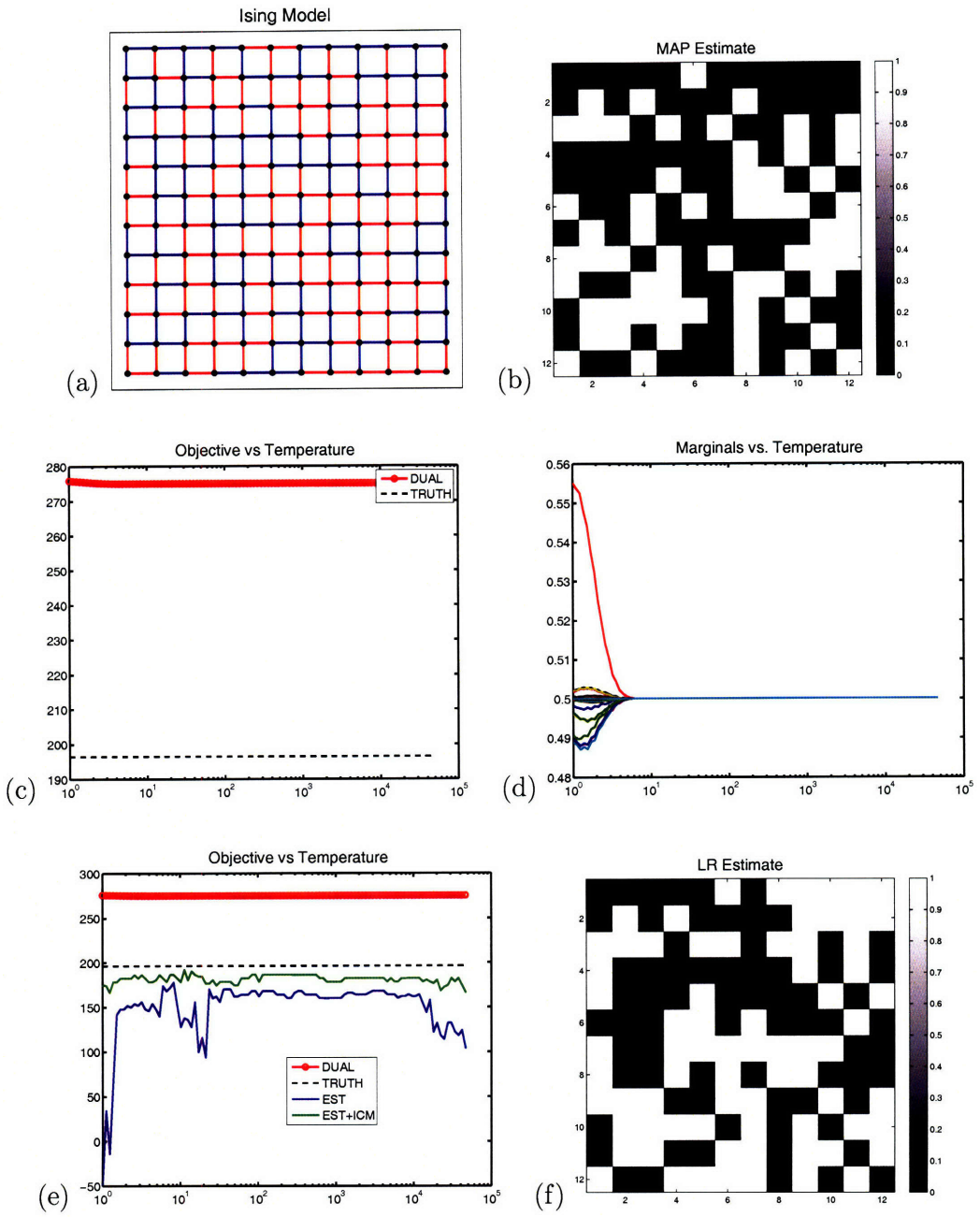

Figure 3.11. Results of applying pairwise LR to a planar frustrated Ising model in zero-field ($h = 0.0$). (a) the nearest-neighbor Ising model with a random mixture of ferromagnetic (blue) and anti-ferromagnetic (red) bonds, (b) the correct MAP estimate, (c) illustration of duality gap, (d) plot showing that all marginal probabilities converge to one-half as the temperature approaches zero, (e) illustration of low-temperature estimates and (f) the best estimate.

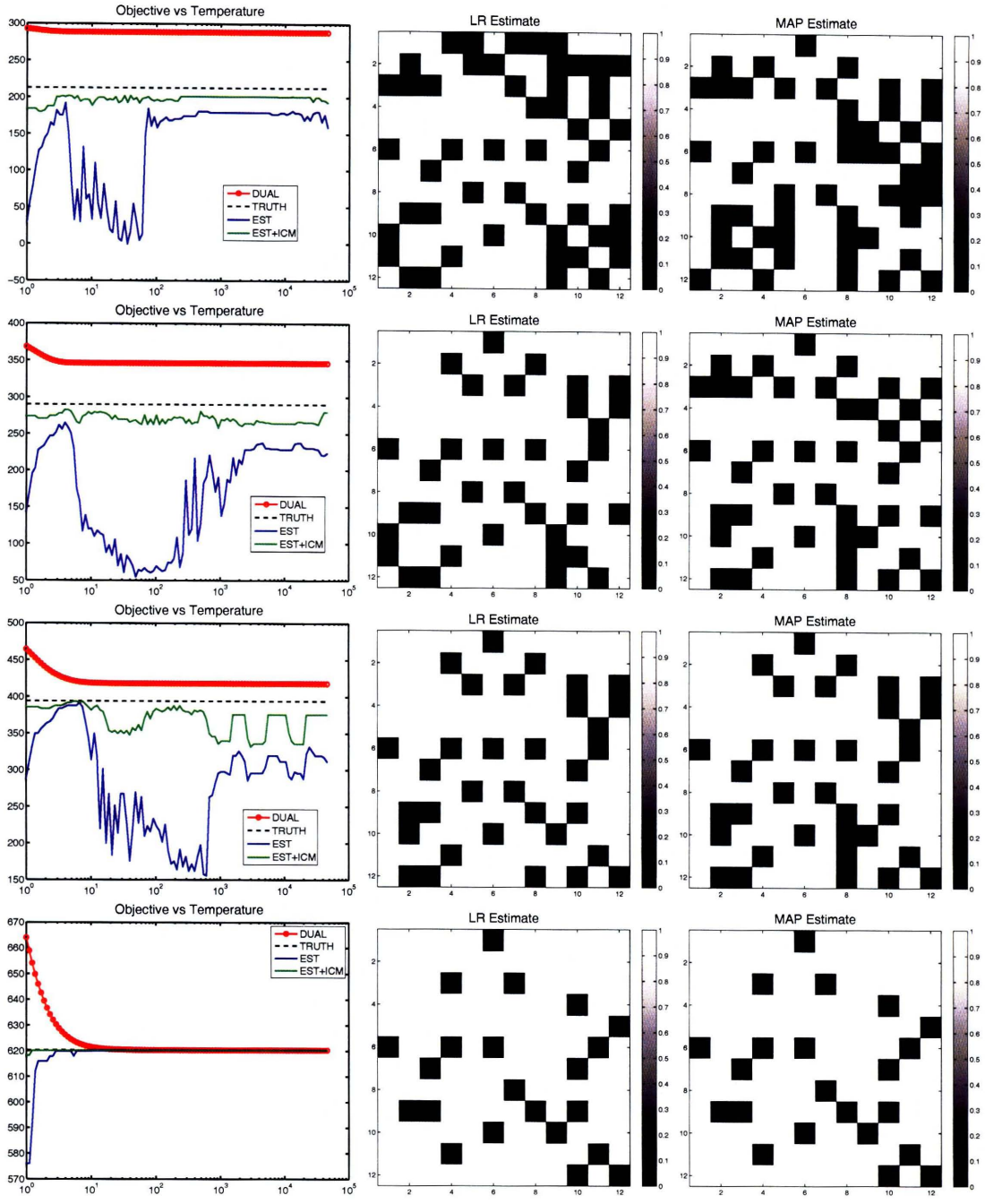

Figure 3.12. Further examples of pairwise LR in the planar frustrated Ising model with constant field $h = 0.1, 0.5, 1.0, 2.0$ (shown by row, top to bottom).

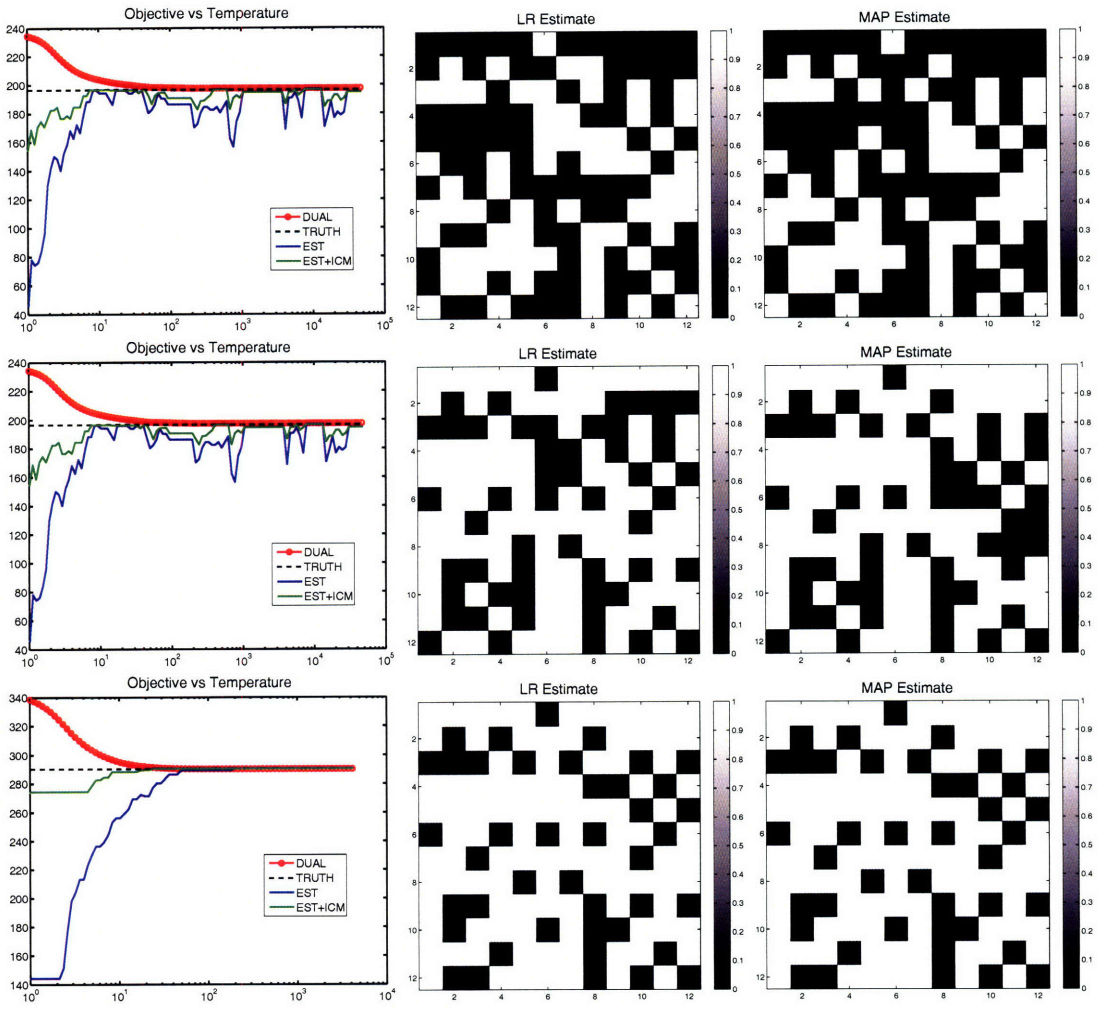

Figure 3.13. Examples of face-based decomposition in the planar frustrated Ising model with constant field $h = 0.0, 0.1, 0.5$ (shown by row, top to bottom). The duality gap is reduced in comparison to pairwise decompositions, and strong duality is obtained for $h = 0.5$ and above (note that the pairwise decomposition exhibits a duality gap at $h = 0.5$ and $h = 1.0$).

results are shown in Figure 3.14. In this case, the pairwise decomposition results in a very large gap (much larger than in the planar graph). It is somewhat surprising that the low-temperature estimates still led to estimates with near-optimal values and which also capture many salient features of the correct MAP estimate.

Next, we try the 4-node decomposition based on the squares of the grid (the maximal cliques of the crossed-bond graph). The results are shown in Figure 3.15. This drastically reduces the duality gap, coming very close to eliminating the gap. However, there is still a small gap, and we do not recover the MAP estimate. The low-temperature estimates again lead to an estimate with near-optimal value and is also very similar to

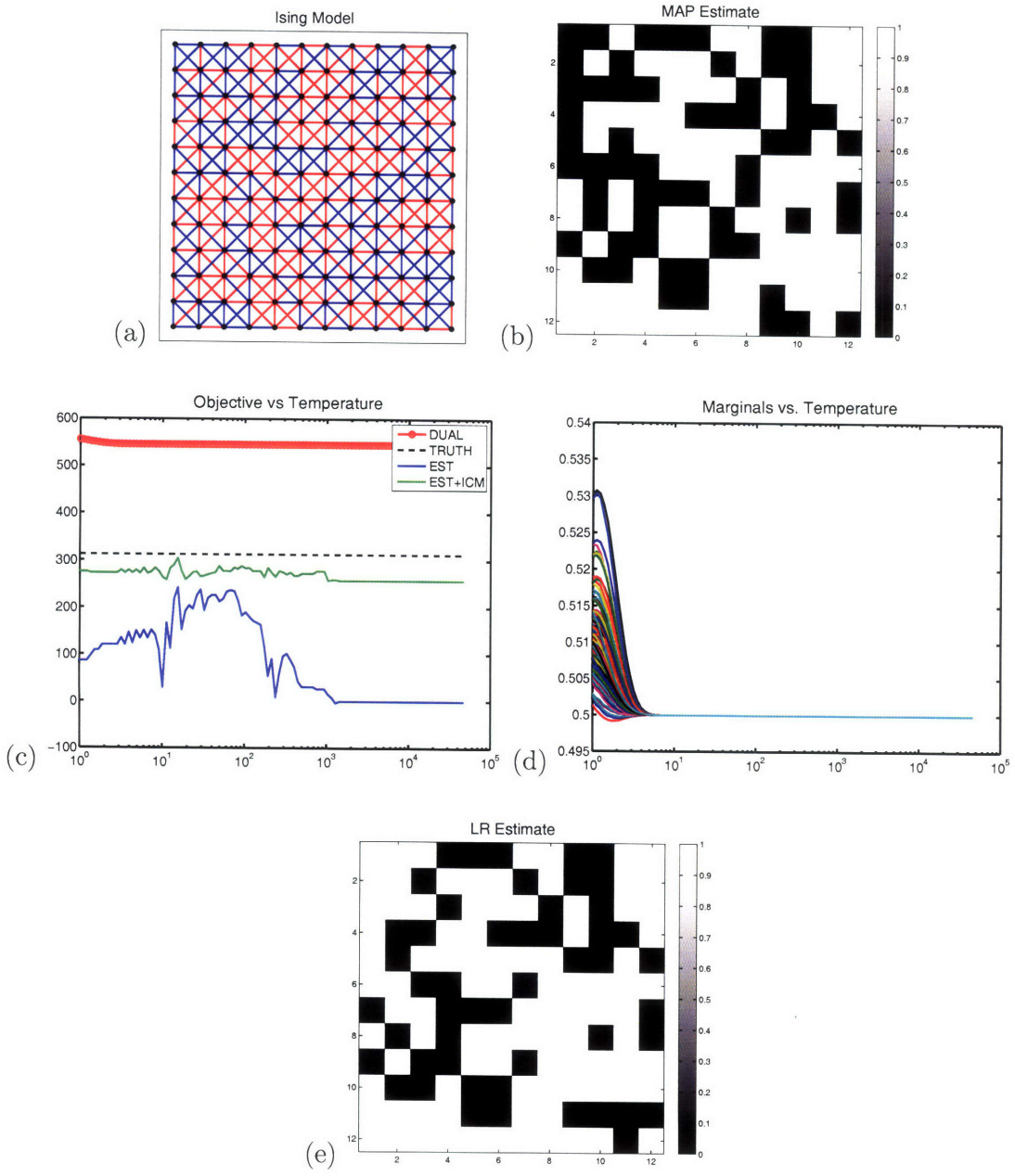

Figure 3.14. Results of applying pairwise LR to a non-planar frustrated model. (a) the crossed-bond lattice with a random mixture of ferromagnetic (blue) and anti-ferromagnetic (red) bonds, (b) the correct MAP estimate (computed using junction trees), (c) values of the dual function and low-temperature estimates as a function of temperature (relative to the MAP value shown by the dashed horizontal line), (d) convergence of node marginals to one-half as temperature approaches zero, (e) the best estimate obtained by the low-temperature estimation method.

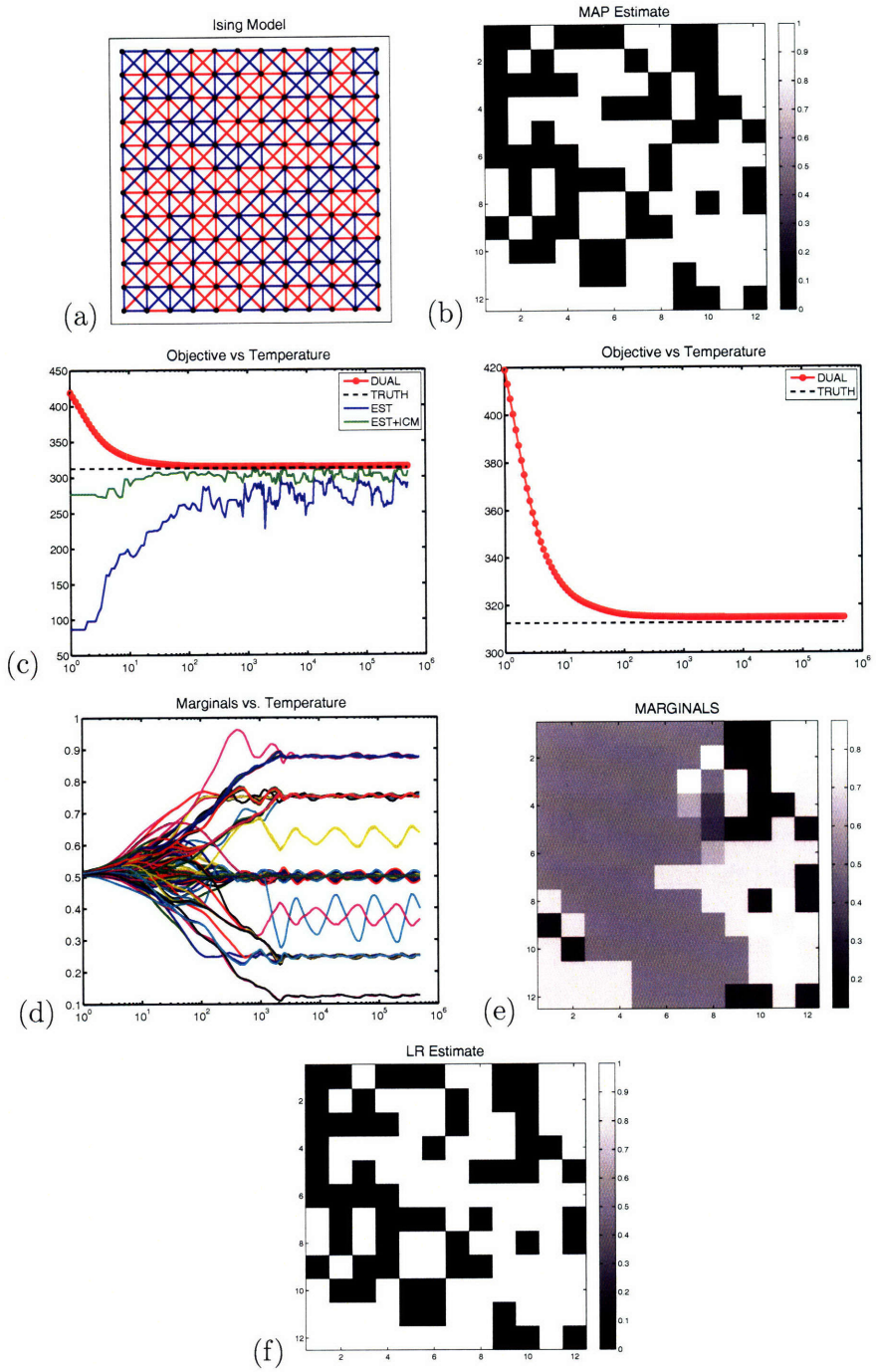

Figure 3.15. Results of applying LR using the 4-cliques of the non-planar frustrated model. (a) the bond configuration, (b) the correct MAP estimate, (c) dual and estimate values compared to MAP, (d) node marginals as a function of temperature (the oscillation is an artifact of the finite-precision optimization performed at each temperature), (e) image of the final zero-temperature marginals, (f) the best low-temperature estimate.

the correct MAP estimate. Another interesting feature of this experiment is that the zero-temperature marginals are not all equal to one-half as was the case in the previous examples with a duality gap. Many converge to one-half, but many converge to other fractional values. Interestingly, the signs of $\eta_i \triangleq \mathbb{E}\{x_i\} = P_i(+) - P_i(-) \in [-1, +1]$ of these partially resolved nodes (that is, the ones where η_i differs from zero) do *usually* correctly predict the value of the corresponding element of the MAP estimate. However, there are a few nodes where this fails to be the case. This is in contrast to the case of the pairwise or tree-based decomposition methods in binary variable models, for which it has been shown that nodes with marginal distributions that differ from one-half may be correctly classified (with respect to the MAP estimate) so as to obtain a partial MAP estimate [134, 135]. However, it may be true (using other block decompositions) that those fully resolved nodes, where $\eta_i = \pm 1$, do lead to a correct partial MAP estimate (our experiments seem to indicate this).

■ 3.5.3 Detecting and Correcting Inconsistent Cycles

To handle cases in which there is a duality gap using the block decomposition based on the squares of the lattice, we investigate the simple correction method presented in Section 3.4.2 for including additional blocks so as to reduce the duality gap. In all the experiments we have done thus far involving Ising models, this method has led to strong duality and recovered the MAP estimate. There is reason to believe that this is always the case for zero-field Ising models defined on planar graphs. However, both the non-planar case (in zero field) and the non-zero field case (on a planar graph) are known to be NP-hard [10], so there must be cases where it is intractable to fully eliminate the duality gap using this approach.

Planar Zero-Field Ising Model

First, we continue the zero-field planar example [shown in Figures 3.11(a) and (b)] where we found that the face-based decomposition led to a duality gap as seen in the top row of Figure 3.13. We now show that by adaptively adding inconsistent cycles arising in the block decomposition method (as defined in Section 3.4.2), we are able to eventually eliminate the duality gap using a small number of additional cycles of the original graph. This procedure is illustrated in Figure 3.16. In each step the duality gap is monotonically decreased, with strong duality achieved after including a total of five cycles (in three steps).

It is interesting to note that the minimal inconsistent cycles arising in each step of this procedure always appear as enlarged faces within the planar graph $\mathcal{G}_{\pm}$ (see Section 3.4.2) as seen on the left in Figures 3.16(a), (b) and (c). This occurs when edges separating faces of the original graph are "removed" by virtue of the corresponding pairwise correlations $\tilde{s}_{ij}$ being equal to zero in the zero-temperature solution at the next iteration, which corresponds to these edges either then having "unconstrained" MAP sets $\{++, +-, -+, --\}$ or one of the end-points being resolved. Also note that the signs s_{ij} of an edge, based on the dual optimal decomposition, can actually *change* between itera-

tions of the method and do not necessarily agree with the sign of $\sigma(f_{ij})$, which indicates whether the corresponding edge potential is ferromagnetic or anti-ferromagnetic. As we include additional inconsistent cycles as blocks in the next iteration, this then either causes the sign s_{ij} of some edge to be reversed (so that the cycle becomes consistent) or causes the edge to be removed so as to eliminate this cycle in $\mathcal{G}_{\pm}$.

Non-Planar Constant-Field Ising Model

Next, we consider a similar strategy for the general Ising model, with non-zero field and a non-planar graph. We apply our method to the non-planar "crossed-bond" example from the previous section. As seen in Figure 3.15, the decomposition based on the 4-cliques of the graph exhibits a duality gap in this example. In Figure 3.17(a), we show the graph of the pairwise correlations between adjacent variables computed from the zero-temperature pairwise marginals arising from the iterative LR algorithm. We observe that many of the correlations $\tilde{s}_{ij}$ have been forced to zero. Most of the remaining correlations converge to $+1$ or -1, but with some fractional values as well (oscillations in the plot are due to finite-precision criteria used to terminate the optimization at each temperature). Hence, we check this graph for inconsistent cycles, i.e., cycles where the product of correlations around the cycle is close to -1. In Figure 3.17(b), we show the result of this calculation in the example. Initially, we find 17 inconsistent 4-cycles in this graph. After correcting these cycles, three more inconsistent 4-cycles appear. Once these are corrected, we obtain strong duality and recover the correct MAP estimate. It is also interesting to note that after correcting for the first set of inconsistent cycles, the new zero-temperature marginals converge to either zero, one-half or one. Comparing the set of resolved nodes (with probabilities of zero or one) to the correct MAP estimate, we see that these resolved nodes correctly determine part of the MAP estimate configuration.

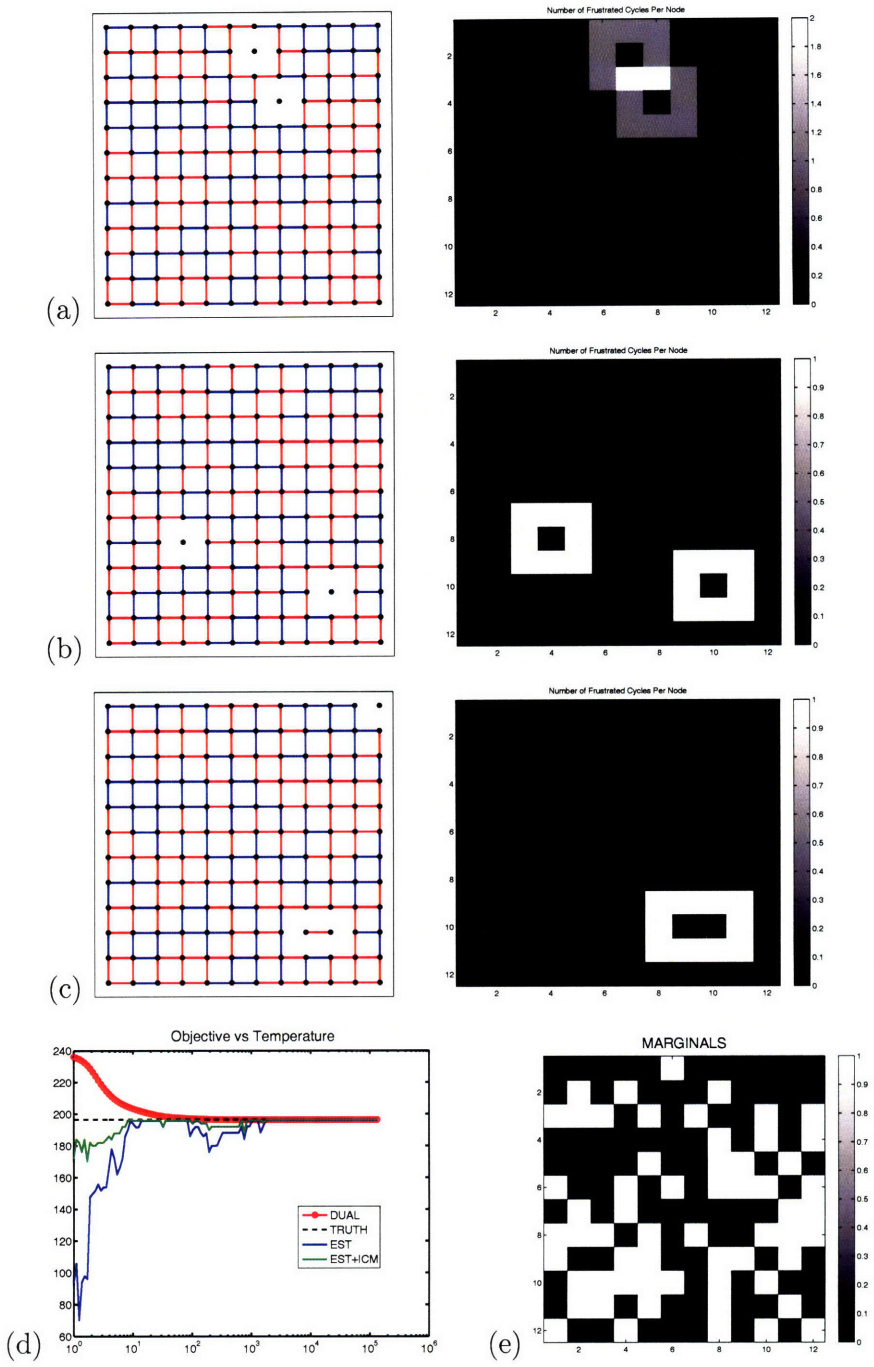

Figure 3.16. Illustration of cycle-correction procedure for the zero-field planar Ising model. (a) The signed graph $\mathcal{G}_{\pm}$ (left) and inconsistent cycles (right). Initially there are 2 inconsistent 8-cycles. (b) After correcting those inconsistent cycles, two more inconsistent 8-cycles appear. (c) After correcting those, one more inconsistent 10-cycle appears and there is one additional inconsistent cycle going around the perimeter of the graph (here, we only include the 10-cycle to correct LR). (d) Correcting this last 10-cycle, we finally eliminate the duality gap. (e) The optimal MAP estimate is obtained.

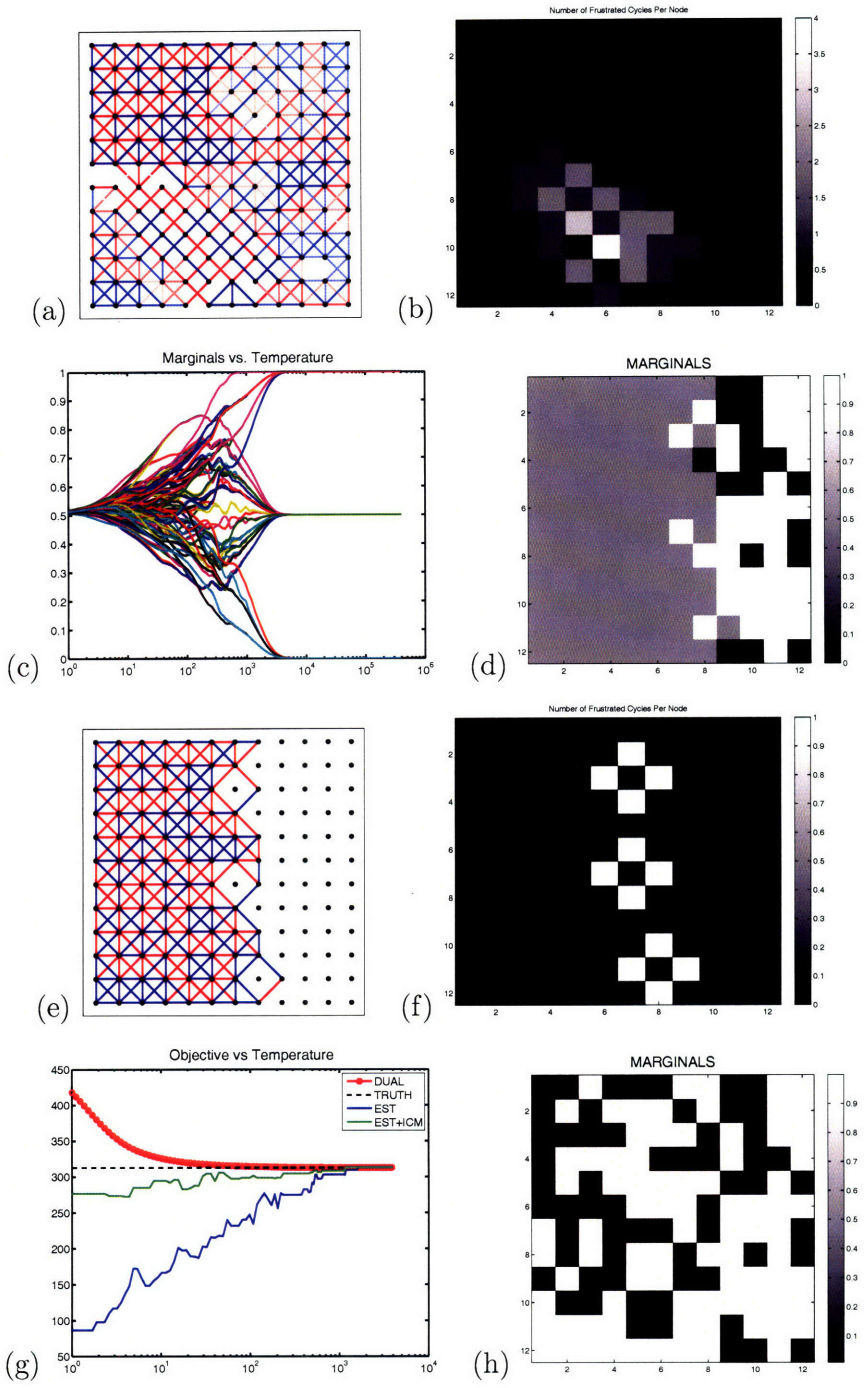

Figure 3.17. Illustration of detection and correction of inconsistent cycles in the non-planar frustrated model. (a) graph of pairwise correlations at zero-temperature, (b) detection of 17 inconsistent 4-cycles, (c) marginals as a function of temperature in corrected LR (including the detected inconsistent cycles), (d) image of the new zero-temperature marginals (partially resolved MAP estimate), (e) graph of new zero-temperature correlations, (f) three more inconsistent 4-cycles are detected, (g) plot showing that the duality gap has been eliminated, (h) the final set of marginals determine the correct MAP estimate.

Chapter 4

Lagrangian Relaxation for Gaussian MRFs

4.1 Introduction

In this chapter we extend the Lagrangian relaxation method of the previous chapter to Gaussian graphical models. In Gaussian graphical models, MAP estimation is a convex optimization problem, minimizing a convex quadratic objective function, which is also equivalent to solving a sparse linear system of equations. Direct methods to solve for the MAP estimate, such as inference over a junction tree or sparse Cholesky factorization, require $\mathcal{O}(nw^3)$ complexity, where w is the tree-width of the graph. In many applications, we must solve high tree-width problems such that direct methods are impractical and iterative methods are used instead. Using our Lagrangian decomposition and iterative scaling approach, we derive a new class of iterative methods to solve for the MAP estimate. In some simple estimation problems, we find that this method is competitive with traditional iterative methods and sometimes substantially out-performs these methods. We also propose a new class of *multiscale* relaxations and demonstrate this approach for the Gaussian model. This gives a new perspective on multiscale methods for solving large linear systems, providing an alternative to the well-known multigrid approach [205].

4.2 Convex-Decomposable Quadratic Optimization Problems

MAP estimation in Gaussian graphical models is equivalent to the problem of maximizing a concave quadratic objective function specified by the information form of the Gaussian distribution:

$$f(x) = -\frac{1}{2}x^T Jx + h^T x \tag{4.1}$$

Note that this is a concave objective function because of the requirement that the information matrix is positive definite. In this chapter we specifically consider the class of *convex-decomposable* objective functions, borrowing this terminology from [160,161],

where the information matrix admits a decomposition of the form

$$J = \sum_{E \in \mathcal{G}} [J^E]_{V \times V} \tag{4.2}$$

where $\mathcal{G}$ is a (generalized) graph and each edge-wise component $J^E \in \mathbb{R}^{|E| \times |E|}$ is a symmetric, positive-definite matrix. This is equivalent to decomposition of the overall energy function $f(x)$ into edge-wise concave potential functions:

$$f(x) = \sum_{E \in \mathcal{G}} f_E(x_E) \tag{4.3}$$

where

$$f_E(x_E) = -\frac{1}{2} x_E^T J^E x_E + (h^E)^T x_E \tag{4.4}$$

and

$$h = \sum_{E \in \mathcal{G}} [h^E]_V \text{ and} \tag{4.5}$$

$$J = \sum_{E \in \mathcal{G}} [J^E]_{V \times V}. \tag{4.6}$$

We note that the class of *pairwise normalizable* models [122, 124, 157] is a subclass of the class of convex decomposable models—pairwise normalizable models are convex decomposable models defined on a pairwise graph. This pairwise normalizable condition is equivalent to the *walk-summability* condition that has been shown to be a sufficient condition for convergence of several iterative methods for inference and estimation in Gaussian graphical models [47, 122, 157]. We show that the generalized condition of convex decomposability is sufficient for success of our Lagrangian relaxation methods, using appropriately structured versions of our method, therefore providing a broader class of models that are shown to be tractable to solve using simple, iterative, distributed algorithms.

Note that, although convex-decomposable models were considered in [160], the methods considered there (essentially Gaussian belief propagation) are only proven to converge in the special case of pairwise normalizable models. This concept of convex-decomposable models has also been studied in [154], under the name of *factor-graph normalizable* models. There, it was shown to be a sufficient condition for well-posedness and convergence of variance estimates in the factor-graph version of Gaussian belief propagation. However, convex-decomposability is still *not* a sufficient condition for convergence of the means in that algorithm (although it was conjectured that it may be sufficient for convergence of a damped version of the algorithm). There is also an open question as to whether or not convex-decomposability is a sufficient condition for convergence of other forms of Gaussian belief propagation, such as Gaussian versions of the generalized belief propagation algorithm [227]. To the best of our knowledge, the Lagrangian relaxation method presented here is the only method that exploits convex-decomposability and has been proven to succeed on this class of models.

■ 4.2.1 Thin-Membrane and Thin-Plate Models for Image Processing

Such models commonly arise in practice. For example, in image processing and remote sensing applications one often performs estimation based on a set of measurements y of an image or random field x. These measurements are typically corrupted by noise and may also be incomplete, in that we only have observations at a limited number of points scattered throughout the field. It is also possible that there is some "blurring" in the measurement process, so that each measurement represents an average over a local region within the field. All of these cases can be described by the measurement model $y_t = c_t^T x_{E_t} + w_t$. Here, t serves as an index over the set of measurements, $E_t \subset V$ specifies the subset of variables which influence measurement y_t, c_t gives the linear dependence of the measurement on those variables, and $w_t \sim N(0, r)$ is independent noise for each measurement.

Conditioned on all such measurements, we then seek to estimate the underlying field based on some *prior model* of the random field, which serves to regularize the estimate in some way, e.g., to impose continuity or smoothness in the image values. We describe two simple prior models that are commonly used in image processing and remote sensing applications. The *thin-membrane* model defines the energy function with respect to a pairwise graph $\mathcal{G}$ as:

$$f_{\text{prior}}(x) = -\frac{1}{2q} \sum_{\{u,v\} \in \mathcal{G}} (x_u - x_v)^2 \tag{4.7}$$

If we view the image values as defining a surface, the thin-membrane model favors level surfaces, where neighboring pixels should have similar values. The *thin-plate* model is another commonly used prior model. Let $N(v)$ denote the set of four nearest neighbors of vertex v in the two-dimensional grid. The thin-plate model is defined by the energy function:

$$f_{\text{prior}}(x) = -\frac{1}{2q} \sum_{v \in V} \left(x_v - \frac{1}{|N(v)|} \sum_{u \in N(v)} x_u \right)^2 \tag{4.8}$$

This energy function favors flat surfaces, that is, surfaces of low curvature. Both models are special cases of the general *conditional auto-regressive* (CAR) model, which is a model of the form:

$$f_{\text{prior}}(x) = -\frac{1}{2q} \sum_k (a_k^T x_{E_k})^2 \tag{4.9}$$

This corresponds to the Gaussian model $Ax \sim N(0, qI)$. In the thin-membrane model, each E_k is a pair of nearest neighbors in the grid and $a_k = (1, -1)^T$. In the thin-plate model, E_k is defined by a set of five nodes centered at each pixel and $c_k = \sigma_p^{-1}(1, -\frac{1}{4}, -\frac{1}{4}, -\frac{1}{4}, -\frac{1}{4})^T$, where the first element corresponds to the center-point of the five-node neighborhood. The general CAR model is convex decomposable, with $J_{\text{prior}} = \sum_k J^{E_k}$ where $\mathcal{G} = \{E_k\}$ and $J^{E_k} = q^{-1} a_k^T a_k$.

We then perform estimation based on a set of measurements y regularized by our prior model. The conditional distribution $P(x|y) \propto P(x) \prod_t P(y_t|x_{E_t})$ is also convex decomposable, with information matrix

$$J = J_{\text{prior}} + r^{-1} \sum_t [c_t c_t^T]_{V \times V} \tag{4.10}$$

and potential vector

$$h = r^{-1} \sum_t y_t [c_t]_V. \tag{4.11}$$

Thus, a wide class of image smoothing, interpolation and deconvolution problems fall within this class of convex-decomposable optimization problems. Using higher-order prior models, such as the thin-plate model, or non-local observations coupling more than two variables at a time (such as in deblurring or deconvolution problems) takes us beyond the simplest class of pairwise normalizable models.

■ 4.2.2 Applications in Non-Linear Estimation

The approaches developed in this chapter are applicable for a wider class of convex-decomposable optimization problems beyond Gaussian estimation. Suppose that we seek to maximize a concave objective function that decomposes into a sum of concave potential functions defined on subsets of variables: $f(x) = \sum_{E \in \mathcal{G}} f_E(x_E)$. We also assume that the objective function is smooth, being at least twice-differentiable so that it may be maximized using Newton's method. For example, in the half-quadratic approach to edge-preserving image restoration [111, 168], one poses estimation using a Gaussian measurement model (as in the preceding discussion) but specifies a non-linear prior model of the form:

$$f_{\text{prior}}(x) = -\sum_k \phi(a_k^T x_{E_k}) \tag{4.12}$$

Here, $\phi(t)$ represent a penalty function that is minimized at zero. The Gaussian model is obtained using the quadratic penalty $\phi(t) = t^2$. In edge-preserving methods, one uses other penalty functions, such as $\phi_p(t) = |t|^p$ with $p < 2$. This is still a convex objective for $1 \leq p < 2$. We may use the "soft-max" approximation to $\phi_p(t) = \max\{t^p, (-t)^p\}$ to obtain the smooth penalty function:

$$\phi_{p,\beta}(t) \triangleq \beta^{-1} \log(e^{\beta t^p} + e^{-\beta t^p}) \tag{4.13}$$

This provides a smooth upper-bound on $\phi_p(t)$ and converges to $\phi_p(t)$ as β becomes large.

Using Newton's method, one iteratively approximates the convex optimization problem as a quadratic optimization problem. Based on the current estimate $\hat{x}$, one com-

putes the gradient and negative Hessian of the objective function:

$$\begin{aligned} h &= \nabla f(\hat{x}) = \sum_{E \in \mathcal{G}} \nabla f_E(\hat{x}_E) \\ J &= -\nabla^2 f(\hat{x}) = -\sum_{E \in \mathcal{G}} \nabla^2 f_E(\hat{x}_E) \end{aligned} \tag{4.14}$$

Then, one solves $J\Delta x = h$ for Δx and the next estimate is set to $\hat{x} \leftarrow \hat{x} + \lambda \Delta x$, where the step-size $\lambda \in (0, 1]$ may be chosen by a back-tracking line search method so as to ensure convergence to the global maximum of $f(x)$. For convex decomposable objectives $f(x)$, the matrix J is convex-decomposable at each step of the procedure, since $J^E \triangleq -\nabla^2 f_E(x_E) \succ 0$ for concave potential functions $f_E(x_E)$. Thus, the methods developed in this chapter can be used in conjunction with Newton's method to solve a wider-class of non-Gaussian convex-decomposable estimation problems.

■ 4.3 Lagrangian Duality

Given a convex-decomposable model, we now consider the Lagrangian decomposition method. To simplify the discussion, we focus on block-decompositions in this chapter, although all the other types of decomposition strategies we have discussed in the previous chapter can also be analogously developed for Gaussian graphical models.

■ 4.3.1 Description of the Dual Problem

Let us begin by stating the dual problem that we solve as simply as possible. Suppose that we are given some decomposition $h = \sum_{E \in \mathcal{G}} [h^E]$ and $J = \sum_{E \in \mathcal{G}} [J^E]$. We obtain a relaxed version of the MAP problem by introducing auxiliary variables x^E on each edge (we use this superscript notation to emphasize that these are independent variables for each E, rather than a subvector or x), and then maximizing the following relaxation of the objective function:

$$f^\dagger(x^\dagger) \triangleq \sum_{E \in \mathcal{G}} f_E(x^E) = \sum_{E \in \mathcal{G}} \left\{ -\frac{1}{2} (x^E)^T J^E x^E + (h^E)^T x_E \right\} \tag{4.15}$$

We use the notation $x^\dagger = (x^E, E \in \mathcal{G})$ to denote the complete set of redundant auxiliary variables. Let the value of this relaxed MAP problem define the *dual function*:

$$g(h^\dagger, J^\dagger) \triangleq \max_{x^\dagger} f^\dagger(x^\dagger) = \sum_{E \in \mathcal{G}} \max_{x^E} \left\{ -\frac{1}{2} (x^E)^T J^E x^E + (h^E)^T x_E \right\} \tag{4.16}$$

Here, we write $h^\dagger = (h^E, E \in \mathcal{G})$ and $J^\dagger = (J^E, E \in \mathcal{G})$ to denote a decomposition of (h, J). Clearly, for each decomposition, this gives an upper-bound on the value of the original MAP problem.[1] Then, the *dual problem* is to minimize this upper-bound over

[1] Let x^* be the MAP estimate and define $x^{\dagger *}$ to be the consistent representation of x^* in the auxiliary variables. Then, $f^\dagger(x^{\dagger *}) = f(x^*)$, owing to $f^\dagger$ being a decomposition of f. Hence, $\max f^\dagger \geq f(x^*)$.

all valid decompositions:

$$\begin{array}{ll} \text{minimize} & g(h^\dagger, J^\dagger) \\ \text{subject to} & \sum_{E\in\mathcal{G}}[h^E]_V = h \\ & \sum_{E\in\mathcal{G}}[J^E]_{V\times V} = J \end{array} \tag{4.17}$$

Note that there is a hidden constraint that each J^E should be positive semi-definite. Otherwise, the maximum over x^E is unbounded above and the value of the dual function is $+\infty$. Thus, if the matrix J is not convex-decomposable, the value of the dual problem is $+\infty$. For positive-definite J^E the maximum over x^E is obtained by $\hat{x}^E = (J^E)^{-1}h^E$, which gives a maximum value of $\frac{1}{2}(h^E)^T(J^E)^{-1}h^E$. Thus, the dual problem can be equivalently stated as minimizing

$$g(h^\dagger, J^\dagger) = \frac{1}{2}\sum_{E\in\mathcal{G}}(h^E)^T(J^E)^{-1}h^E \tag{4.18}$$

subject to the constraint that $(h^\dagger, J^\dagger)$ is a valid *convex* decomposition of (h, J), such that $J^E \succ 0$ for all $E \in \mathcal{G}$. This dual problem is infeasible if J is not convex decomposable. We also remark that minimizing this dual function for a fixed decomposition of J reduces to a linearly-constrained convex quadratic optimization problem with respect to the problem variable $h^\dagger$.

Before proceeding further, we introduce some additional notational conventions that simplify our discussion. Let D denote the linear operator mapping x to a consistent decomposition $x^\dagger = Dx$. Also, let us overload our notation a bit by allowing $h^\dagger$ to denote a block-vector representation of $(h^E, E \in \mathcal{G})$ and by allowing $J^\dagger$ to denote a *block-diagonal matrix* representation of the J-decomposition, with each edge's J^E being assigned to a separate block along the diagonal of $J^\dagger$. In this notation, the condition that $(h^\dagger, J^\dagger)$ defines a valid decomposition of (h, J) is written simply as:

$$\begin{aligned} D^T h^\dagger &= h \\ D^T J^\dagger D &= J \end{aligned} \tag{4.19}$$

Also, the objective in the relaxed MAP problem becomes:

$$f^\dagger(x^\dagger) = -\frac{1}{2}(x^\dagger)^T J^\dagger x^\dagger + (h^\dagger)^T x^\dagger \tag{4.20}$$

We can easily check that $f^\dagger(Dx) = f(x)$ for all x using the conditions (4.19):

$$f^\dagger(Dx) = -\frac{1}{2}x^T(D^T J^\dagger D)x + (Dh^\dagger)^T x = -\frac{1}{2}x^T Jx + h^T x = f(x) \tag{4.21}$$

The optimal solution of the relaxed MAP problem is $\hat{x}^\dagger = (J^\dagger)^{-1}h^\dagger$. Thus, the dual problem can now be restated as minimizing

$$g(h^\dagger, J^\dagger) = \frac{1}{2}h^\dagger(J^\dagger)^{-1}h^\dagger \tag{4.22}$$

subject to the constraints that $D^T J^\dagger D = J$, $D^T h^\dagger = h$, $J^\dagger \succ 0$ and that $J^\dagger$ has the required block-diagonal structure.

■ 4.3.2 Derivation using Lagrange Multipliers

In this section, we take a brief detour to formally derive the above dual problem as a Lagrangian decomposition of the MAP estimation problem.

Given a decomposition $(h^\dagger, J^\dagger)$ of (h, J) over $\mathcal{G}$, the exact MAP can be equivalently stated as maximizing $f^\dagger(x^\dagger)$ subject to the following constraints on $x^\dagger$:

$$\begin{aligned} x_v^{E_1} &= x_v^{E_2} \\ x_u^{E_1} x_v^{E_1} &= x_u^{E_2} x_v^{E_2} \end{aligned}$$

for all $E_1, E_2 \in \mathcal{G}$ and $u, v \in E_1 \cap E_2$. We have added linear constraints between duplicated variables to force these to be assigned consistently in the MAP estimate. Note that we have also included second-order constraints, involving consistency of products of variables. This include the constraints $(x_v^{E_1})^2 = (x_v^{E_2})^2$ as a special case (for $u = v$). Although these second-order constraints are redundant (they are already implied by the first-order constraints), they do serve a purpose in the following relaxation.

Introducing Lagrange multipliers λ for each constraint, we obtain the Lagrangian:

$$\begin{aligned} L(x^\dagger; \lambda) &= f^\dagger(x^\dagger) \\ &\quad + \sum_{E_1, E_2, v \in E_1 \cap E_2} \lambda_v^{E_1, E_2} (x_v^{E_1} - x_v^{E_2}) + \\ &\quad + \sum_{E_1, E_2, u, v \in E_1 \cap E_2} \lambda_{u,v}^{E_1, E_2} (x_u^{E_1} x_v^{E_1} - x_u^{E_2} x_v^{E_2}) \end{aligned} \tag{4.23}$$

Note how the Lagrange multipliers coupling two edges simply serve to add or subtract from the information forms defined on either edge. Hence, we can rewrite this Lagrangian as a sum of edge-wise information forms:

$$L(x^\dagger; \lambda) = \sum_{E \in \mathcal{G}} f_E(x^E; \lambda) \tag{4.24}$$

where

$$\begin{aligned} f_E(x^E; \lambda) &= -\frac{1}{2}(x^E)^T J^E(\lambda) x^E + (h^E(\lambda))^T x^E \\ h_v^E(\lambda) &= h_v^E + \sum_{E' : v \in E'} (\lambda_v^{(E,E')} - \lambda_v^{(E',E)}) \\ J_{u,v}^E(\lambda) &= J_{u,v}^E + \sum_{E' : u,v \in E'} (\lambda_{u,v}^{(E,E')} - \lambda_{u,v}^{(E',E)}) \end{aligned}$$

The main point here is that optimization over the set of Lagrange multipliers is equivalent to optimization over all valid decompositions of (h, J) among the edges of $\mathcal{G}$.

The Lagrangian dual function $g(\lambda)$ is defined as the maximum-value of the Lagrangian $L(x^\dagger; \lambda)$ for a given λ. Note that the maximization over $x^\dagger$ is separable with

respect the edges of $\mathcal{G}$:

$$\begin{aligned} g(\lambda) &\triangleq \max_{x^\dagger} L(x^\dagger;\lambda) \\ &= \sum_{E\in\mathcal{G}} \max_{x^E} \left\{ -\frac{1}{2}(x^E)^T J^E(\lambda) x^E + (h^E(\lambda))^T x^E \right\} \end{aligned} \tag{4.25}$$

The Lagrangian dual problem is then to minimize this dual function over all λ. This is equivalent to the dual problem we stated earlier in terms of an optimization over all decompositions of (h, J). Note also that it is the second-order constraints that lead to optimization over the decomposition of J in the dual problem. If we omitted those constraints, we would obtain a dual problem which only optimizes over the decomposition of h (for a fixed decomposition of J), which reduces to a convex quadratic optimization problem in $h^\dagger$.

■ 4.3.3 Strong Duality of Gaussian Lagrangian Relaxation

In this section we demonstrate that if the information matrix J is convex-decomposable over $\mathcal{G}$, then the Lagrangian relaxation based on $\mathcal{G}$ results in strong duality and we recover the optimal MAP estimate. In fact, we show that strong duality holds even if we fix the decomposition of J, using any convex decomposition, and only optimize over the decomposition of h.

Proposition 4.3.1 (Strong Duality of Gaussian LR). *Let J be convex-decomposable over $\mathcal{G}$, that is, there exists a decomposition $J = \sum_{E\in\mathcal{G}}[J^E]$ with $J^E \succ 0$ for all $E \in \mathcal{G}$. Then,*

1. *For each convex decomposition of J there is a unique decomposition $h^\dagger = (h^E, E \in \mathcal{G})$ that minimizes the dual function.*
2. *This optimal decomposition of h produces a consistent set of edge-wise estimates in the relaxed MAP problem. This determines the optimal MAP estimate and shows that there is no duality gap.*
3. *The optimal decomposition is uniquely determined by the condition that it produces a consistent estimate. Thus, solving the dual problem is equivalent to finding a decomposition of h that yields a consistent estimate in the relaxed MAP problem.*

Proof. The optimal solution of this dual problem is shown to be $h^\dagger = J^\dagger D J^{-1} h$. First, we verify that this is a valid decomposition of h:

$$\begin{aligned} \sum_{E\in\mathcal{G}} [h^E] &= D^T h^\dagger \\ &= D^T J^\dagger D J^{-1} h \\ &= J J^{-1} h \\ &= h \end{aligned} \tag{4.26}$$

We have used the fact the $J^\dagger$ represents a valid decomposition of J, which is equivalent to $D^T J^\dagger D = J$. Next, we evaluate the dual function for this $h^\dagger$:

$$\begin{aligned} g(h^\dagger, J^\dagger) &= \frac{1}{2} h^T J^{-1} D^T J^\dagger (J^\dagger)^{-1} J^\dagger D J^{-1} h \\ &= \frac{1}{2} h^T J^{-1} D^T J^\dagger D J^{-1} h \\ &= \frac{1}{2} h^T J^{-1} J J^{-1} h \\ &= \frac{1}{2} h^T J^{-1} h \end{aligned} \tag{4.27}$$

This final value is equal to the maximum value of $f(x) = -\frac{1}{2} x^T J x + h^T x$. Because the dual function is greater than or equal to the maximum value of the primal objective $f(x)$, this shows both that $h^\dagger$ minimizes the dual function and that there is no duality gap, that is, the minimum value of the dual function is equal to the maximum value of $f(x)$. We also show that this choice of $h^\dagger$ leads to consistency in the relaxed MAP estimate $\hat{x}^\dagger = (J^\dagger)^{-1} h^\dagger$. For the optimal $h^\dagger$, we have:

$$\begin{aligned} \hat{x}^\dagger &= (J^\dagger)^{-1} h^\dagger \\ &= (J^\dagger)^{-1} J^\dagger D J^{-1} h \\ &= D J^{-1} h \\ &= D \hat{x} \end{aligned} \tag{4.28}$$

where $\hat{x} = J^{-1} h$ is the optimal MAP estimate. Finally, we demonstrate that there is only one valid decomposition that produces a consistent estimate in the relaxed MAP problem, and that is given by the optimal value of $h^\dagger$ given previously. If $h^\dagger$ produces a consistent solution $\tilde{x}$, then $h^\dagger = J^\dagger D \tilde{x}$ and $D h^\dagger = D J^\dagger D \tilde{x} = J \tilde{x}$. However, a valid decomposition must satisfy $D h^\dagger = h$, which implies $J \tilde{x} = h$ and $\tilde{x} = J^{-1} h = \hat{x}$. Thus, $h^\dagger = J^\dagger D J^{-1} h$ is the only valid decomposition of h that provides a consistent estimate in the relaxed MAP problem. □

■ 4.3.4 Regularized Decomposition of J

In view of the fact that strong duality holds using *any* convex decomposition of J, the reader may ask why we have included the optimization over decomposition of J as part of our definition of the dual problem. The reason for this is two-fold. First, it allows us to develop an iterative algorithm for solving the dual problem that is analogous to the iterative "marginal matching" algorithm of the previous chapter. Also, we find that optimizing the decomposition of J is important so that the problem of finding a corresponding decomposition of h is well-conditioned and therefore more easily solved.

We regularize the choice of how to decompose J over the edges of $\mathcal{G}$ using the

following log-determinant criterion:

$$\begin{array}{ll} \text{maximize} & \sum_{E\in\mathcal{G}} \log\det J^E \\ \text{subject to} & \sum_{E\in\mathcal{G}} [J^E]_V = J \\ & J^E \succ 0 \end{array}$$

Note that this is equivalent to maximizing $\log\det J^\dagger$ subject to the constraints that $D^T J^\dagger D = J$, $J^\dagger \succ 0$ and $J^\dagger$ has the required block-diagonal structure (so that the edge variables x^E are independent in the relaxed MAP problem). Essentially, this ensures that the decomposition is balanced such that no one edge potential is particularly close to being singular.

We combine this with the previously defined dual function to obtain the following *regularized dual problem*:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\sum_{E\in\mathcal{G}} \left\{ -\log\det J^E + (h^E)^T (J^E)^{-1} h^E \right\} \\ \text{subject to} & \sum_{E\in\mathcal{G}} [h^E]_V = h \\ & \sum_{E\in\mathcal{G}} [J^E]_{V\times V} = J \\ & J^E \succ 0 \end{array} \tag{4.29}$$

Recall that the Gaussian log-partition function is given by:

$$\begin{aligned} \Phi(h,J) &\triangleq \log \int \exp\{-\frac{1}{2}x^T J x + h^T x\} dx \\ &= \frac{1}{2}\left\{ -\log\det J + h^T J^{-1} h + n\log 2\pi \right\} \end{aligned} \tag{4.30}$$

This is a convex function of (h, J). Then, the regularized dual problem is equivalent to minimizing a sum of edge-wise log-partition functions in the relaxed model over all valid decompositions of (h, J):

$$\begin{array}{ll} \text{minimize} & \sum_{E\in\mathcal{G}} \Phi_E(h^E, J^E) \\ \text{subject to} & \sum_{E\in\mathcal{G}} [h^E]_V = h \\ & \sum_{E\in\mathcal{G}} [J^E]_{V\times V} = J \\ & J^E \succ 0 \end{array}$$

This is also equivalent to minimizing $\Phi(h^\dagger, J^\dagger)$ in the relaxed MRF subject to the constraints $D^T h^\dagger = h$, $D^T J^\dagger D = J$ and that $J^\dagger$ is block-diagonal. Note that this problem is the Gaussian analog of the log-sum-exp approximation we used previously to obtain a smoothed dual problem in the Lagrangian relaxation method for discrete MRFs. In the Gaussian case, however, it is not necessary to include a temperature parameter because the log-determinant regularization does not change the optimal estimate $\hat{x}$ that is obtained by solving the regularized dual problem.[2]

[2]In principle, we could include a temperature parameter here as well, using $\Phi_\tau(h,J) \triangleq \tau\Phi(\frac{h}{\tau}, \frac{J}{\tau})$ in the dual problem. This does not change the estimate $\hat{x} = (\frac{J}{\tau})^{-1}(\frac{h}{\tau}) = J^{-1}h$, but does scale the covariance $K(\tau) = \tau K$. Thus, gradually reducing the temperature to zero causes the Gaussian solution to become concentrated at $\hat{x}$ as the temperature goes to zero. Although this is not necessary to solve Gaussian problems, it could prove useful in certain Gaussian relaxations of discrete models discussed in the conclusions (see Chapter 6).

We also recall the following moment-generating property of the log-partition function:

$$\begin{aligned}\frac{\partial \Phi(h,J)}{\partial h_E} &= \hat{x}_E \\ \frac{\partial \Phi(h,J)}{\partial J_{E,E}} &= K_{E,E} + \hat{x}_E(\hat{x}_E)^T\end{aligned}$$

where $\hat{x} = J^{-1}h$ and $K = J^{-1}$. Using this property, it is not hard to show the following moment-matching condition for optimality of a decomposition of (h, J) over $\mathcal{G}$ in the regularized dual problem. This is analogous to the result for discrete graphical models that the optimal decomposition between intersecting edges should assign the same marginal distributions within each edge for any shared subset of variables. In Gaussian models, this is equivalent to asserting that the means and covariances of those shared variables should be equal within both edges.

Proposition 4.3.2. *If the regularized dual problem is feasible, that is, if there exist a convex decomposition of J, then there is a unique decomposition of h and J that minimizes the regularized dual function. A given decomposition is optimal if an only if it satisfies the following moment matching conditions. For all edges $A, B \in \mathcal{G}$ with non-empty intersection $S = A \cap B$ it holds that:*

$$((J^A)^{-1})_{S,S} = ((J^B)^{-1})_{S,S} \tag{4.31}$$

$$((J^A)^{-1}h^A)_S = ((J^B)^{-1}h^B)_S \tag{4.32}$$

Proof. To simplify the demonstration, we use the description of the dual problem in terms of Lagrange multipliers. The variations over how we decompose h and J between edges A and B can be summarized by a set of Lagrange multipliers λ and Λ defined on $S = A \cap B$ such that:

$$\begin{aligned}h_S^A(\lambda) &= h_S^A + \lambda \\ h_S^B(\lambda) &= h_S^B - \lambda \\ J_{S,S}^A(\Lambda) &= J_{S,S}^A + \Lambda \\ J_{S,S}^B(\Lambda) &= J_{S,S}^B - \Lambda\end{aligned}$$

Note that other elements of h^A, h^B, J^A and J^B (except for those indexed by S) do not change with (λ, Λ). The dependence of the dual function on these variables is captured by:

$$g(\lambda, \Lambda) = \Phi_A(h^A(\lambda), J^A(\Lambda)) + \Phi_B(h^B(\lambda), J^B(\Lambda)) \tag{4.33}$$

This is a convex function of (λ, Λ), so that the minimum is determined by:

$$\begin{aligned}\frac{\partial g}{\partial \lambda} &= 0 \\ \frac{\partial g}{\partial \Lambda} &= 0\end{aligned}$$

Using the moment-generating property and the chain rule, we have:

$$\begin{aligned}\frac{\partial g(\lambda,\Lambda)}{\partial \lambda} &= \frac{\partial \Phi_A(h^A(\lambda),J^A(\Lambda))}{\partial \lambda} + \frac{\partial \Phi_B(h^B(\lambda),J^B(\Lambda))}{\partial \lambda} \\ &= (J^A(\Lambda)^{-1}h^A(\lambda))_S - (J^B(\Lambda)^{-1}h^B(\lambda))_S\end{aligned}$$

and

$$\begin{aligned}\frac{\partial g(\lambda,\Lambda)}{\partial \Lambda} &= \frac{\partial \Phi_A(h^A(\lambda),J^A(\Lambda))}{\partial \Lambda} + \frac{\partial \Phi_B(h^B(\lambda),J^B(\Lambda))}{\partial \Lambda} \\ &= (J^A(\Lambda)^{-1})_{S,S} - (J^B(\Lambda)^{-1})_{S,S}.\end{aligned}$$

Thus, the optimal decomposition is determined by the conditions:

$$\begin{aligned}(J^A(\Lambda)^{-1}h^A(\lambda))_S &= (J^B(\Lambda)^{-1}h^B(\lambda))_S \\ (J^A(\Lambda)^{-1})_{S,S} &= (J^B(\Lambda)^{-1})_{S,S}\end{aligned}$$

Thus, if a given decomposition satisfies these moment matching conditions for all pairs of edges which share variables, then the gradient of the dual function with respect to the Lagrange multipliers is zero, which shows that this is the global minimum of the convex dual function. □

Comment on Variances

We also comment that, as discussed in [125], solution of the regularized dual problem leads to a set of *upper-bounds* on the correct variances of the model. However, we have found that these variance bounds tend to be quite loose in practice, and so we omit the proof here. It is an open-question as to whether or not the method can be modified to obtain tighter bounds on variances.

■ 4.4 Gaussian Iterative Scaling Algorithm

We now specify an iterative marginal-matching procedure to minimize the regularized Lagrangian dual function in Gaussian graphical models. This may be viewed either as a block-coordinate descent procedure with respect to Lagrange multipliers (which serve to parameterize valid decompositions of the model) or as an iterative information projection procedure that iteratively enforces marginal matching constraints between different components of the decomposition.

■ 4.4.1 Derivation of the Method

We begin by considering the problem of matching covariance matrices between two edges $A, B \in \mathcal{G}$ that share the variables $S = A \cap B$. Given an initial decomposition with components J^A and J^B on these two edges, we wish to solve for a matrix Λ such that if we add Λ to the S-submatrix of J^A and subtract it from the S-submatrix

of J^B, then the new decomposition satisfies this covariance matching condition. In mathematical notation, we write this as:

$$((J^A + [\Lambda]_{A\times A})^{-1})_{S,S} = ((J^B - [\Lambda]_{B\times B})^{-1})_{S,S} \tag{4.34}$$

Taking inverses of both sides of this equation and using the Schur complement formula for marginalization in the information matrix, this reduces to the condition:

$$(J^A_{S,S}+\Lambda) - J^A_{S,A\setminus S}(J^A_{A\setminus S,A\setminus S})^{-1}J^A_{A\setminus S,S} = (J^B_{S,S}-\Lambda) - J^B_{S,B\setminus S}(J^B_{B\setminus S,B\setminus S})^{-1}J^B_{B\setminus S,S} \tag{4.35}$$

Solving for Λ, we obtain:

$$\Lambda = \frac{1}{2}\left((K^B_{S,S})^{-1} - (K^A_{S,S})^{-1}\right) \tag{4.36}$$

where $K^A = (J^A)^{-1}$ and $K^B = (J^B)^{-1}$. Thus, the optimal decomposition for these two edges (with the potentials on all other edges held fixed) is obtained by the correction step:

$$J^A \leftarrow J^A + \frac{1}{2}\left((K^B_{S,S})^{-1} - (K^A_{S,S})^{-1}\right) \tag{4.37}$$

$$J^B \leftarrow J^B + \frac{1}{2}\left((K^A_{S,S})^{-1} - (K^B_{S,S})^{-1}\right) \tag{4.38}$$

Let K^S be defined by:

$$(K^S)^{-1} = \frac{1}{2}\left((K^A_{S,S})^{-1} + (K^B_{S,S})^{-1}\right), \tag{4.39}$$

Using this notation, we may rewrite the update as:

$$J^A \leftarrow J^A + \left[(K^S)^{-1} - (K^A_{S,S})^{-1}\right]_{A\times A} \tag{4.40}$$

$$J^B \leftarrow J^B + \left[(K^S)^{-1} - (K^B_{S,S})^{-1}\right]_{B\times B} \tag{4.41}$$

After performing this update, both edges then have the same covariance on nodes S. The new value of the covariance in either edge is then K^S. Thus, the optimal correction may be viewed as "shifting" either edge's marginal information matrix on nodes S from their previous values to the averaged value.

By a similar derivation, we can extend this method to also satisfy the condition that the means on nodes S should be equal. Given (h^A, J^A) and (h^B, J^B), let $\hat{x}^S$ be defined by:

$$(K^S)^{-1}\hat{x}^S = \frac{1}{2}\left((K^A_{S,S})^{-1}\hat{x}^A_S + (K^B_{S,S})^{-1}\hat{x}^B_S\right) \tag{4.42}$$

where $\hat{x}^A = (J^A)^{-1}h^A$ and $\hat{x}^B = (J^B)^{-1}h^B$ are the (relaxed) MAP estimates on edges A and B in the decomposed model. The decomposition of h is then updated according to:

$$h^A \leftarrow h^A + \left[(K^S)^{-1}\hat{x}^S - (K^A_{S,S})^{-1}\hat{x}^A_S\right]_A \tag{4.43}$$

$$h^B \leftarrow h^B + \left[(K^S)^{-1}\hat{x}^S - (K^B_{S,S})^{-1}\hat{x}^B_S\right]_A \tag{4.44}$$

This update of the h decomposition between edges A and B must be performed together with the update of the J decomposition. The new marginal distribution of nodes S are then equal in edge A and B.

This method can be generalized to enforce the moment matching condition between any number of edges that share a common subset of nodes. To demonstrate this method, let $S \subset V$ and let the set of all edges that cover S be defined as $\mathcal{G}(S) = \{E \in \mathcal{G} | S \subseteq E\}$. Then, the moment matching condition on nodes S may be enforced across all edges as follows. First, compute the marginal moments on S with respect to each of these edges and let this be denoted as $(\hat{x}_S^E, K_{S,S}^E)$. Then, let $(\hat{x}^S, K^S)$ be defined by:

$$(K^S)^{-1} = \frac{1}{|\mathcal{G}(S)|} \sum_{E \in \mathcal{G}(S)} (K_{S,S}^E)^{-1} \tag{4.45}$$

$$(K^S)^{-1}\hat{x}^S = \frac{1}{|\mathcal{G}(S)|} \sum_{E \in \mathcal{G}(S)} (K_{S,S}^E)^{-1}\hat{x}_S^E \tag{4.46}$$

Note, this corresponds to averaging the marginal information forms across all the edges that cover S. Then, each edge's information form is updated as shown below:

$$J^E \leftarrow J^E + \left[(K^S)^{-1} - (K_{S,S}^E)^{-1}\right] \tag{4.47}$$

$$h^E \leftarrow h^E + \left[(K^S)^{-1}\hat{x}^S - (K_{S,S}^E)^{-1}\hat{x}_S^E\right] \tag{4.48}$$

Thus, each correction step may be viewed as an *information diffusion* step in which overlapping edges share their "beliefs" concerning the distribution of common variables, and then achieve consensus among themselves by averaging these beliefs in the information-form representation.

We demonstrate correctness of this method in the following proposition.

Proposition 4.4.1. *The preceding correction to the decomposition of (h, J) with respect to S satisfies the following conditions:*

1. *The information form $h = \sum_E [h^E]$ and $J = \sum_E [J^E]$ is unchanged by this correction step.*

2. *If the initial decomposition $J = \sum_E [J^E]$ is convex, that is, if $J^E \succeq 0$ for all $E \in \mathcal{G}$, then the decomposition after the update is also convex.*

3. *In the corrected decomposition, the marginal moments on S are equal to $(\hat{x}^S, K^S)$ in all edges $E \in \mathcal{G}(S)$.*

Proof. The proof is primarily based on the Schur complement formula for marginalization in the information form. We use $(\tilde{h}^E, \tilde{J}^E)$ to denote the model after the correction step (and similarly for the moment parameters).

Proof of (1): To show that the correction leads to a valid decomposition, we check that the total addition to the submatrix $J_{S,S}$ is zero:

$$\begin{aligned}\Delta J_{S,S} &= \sum_{E\in\mathcal{G}(S)} (\tilde{J}^E_{S,S} - J^E_{S,S}) \\ &= \sum_{E\in\mathcal{G}(S)} \left[(K^S)^{-1} - (K^E_S)^{-1}\right] \\ &= |\mathcal{G}(S)|(K^S)^{-1} - \sum_{E\in\mathcal{G}(S)} (K^E_{S,S})^{-1} \\ &= 0\end{aligned}$$

The last step follows because K^S is defined by $(K^S)^{-1} = |\mathcal{G}(S)|^{-1}\sum_{E\in\mathcal{G}(S)}(K^E_{S,S})^{-1}$. Similarly, the total addition to the subvector h_S is:

$$\begin{aligned}\Delta h_S &= \sum_{E\in\mathcal{G}(S)} (\tilde{h}^E_S - h^E_S) \\ &= \sum_{E\in\mathcal{G}(S)} \left[(K^S)^{-1}\hat{x}^S - (K^E_{S,S})^{-1}\hat{x}^E_S\right] \\ &= |\mathcal{G}(S)|(K^S)^{-1}\hat{x}^S - \sum_{E\in\mathcal{G}(S)} (K^E_{S,S})^{-1}\hat{x}^E_S \\ &= 0\end{aligned}$$

Thus, it holds that $D^T\tilde{J}^\dagger D = D^T J^\dagger D = J$ and $D^T\tilde{h}^\dagger = D^T h^\dagger = h$.

Proof of (2): We recall that $J^E \succ 0$ is equivalent to the condition that both $J^E_{E\setminus S,E\setminus S} \succ 0$ and $(K^E_{S,S})^{-1} \succ 0$, that is, that both the Schur complement on S and the submatrix of $E \setminus S$ are positive definite. If $J^E \succ 0$ for all $E \in \mathcal{G}$, then we show that $\tilde{J}^E \succ 0$. This is because $\tilde{J}^E_{E\setminus S,E\setminus S} = J^E_{E\setminus S,E\setminus S} \succ 0$ and $(\tilde{K}^E_{S,S})^{-1} = (K^S)^{-1} = |\mathcal{G}(S)|^{-1}\sum_{E'\in\mathcal{G}(S)}(K^{E'}_{S,S})^{-1} \succ 0$ by (3) (proven below).

Proof of (3): If we recompute the Schur complement in each edge using the corrected decomposition, we now obtain:

$$\begin{aligned}(\tilde{K}^E_{S,S})^{-1} &= \tilde{J}^E_{S,S} - \tilde{J}^E_{S,E\setminus S}(\tilde{J}^E_{E\setminus S,E\setminus S})^{-1}\tilde{J}_{E\setminus S,S} \\ &= (J^E_{S,S} + (K^S)^{-1} - (K^E_{S,S})^{-1}) - J^E_{S,E\setminus S}(J^E_{E\setminus S,E\setminus S})^{-1}J_{E\setminus S,S} \\ &= (K^S)^{-1} - \cancel{(K^E_{S,S})^{-1}} + \cancel{(J^E_{S,S} - J^E_{S,E\setminus S}(J^E_{E\setminus S,E\setminus S})^{-1}J_{E\setminus S,S})} \\ &= (K^S)^{-1}\end{aligned} \tag{4.49}$$

Thus, $\tilde{K}^E_{S,S} = K^S$. Similarly,

$$\begin{aligned}
(\tilde{K}^E_{S,S})^{-1}\tilde{x}^E_S &= \tilde{h}^E_{S,S} - \tilde{J}^E_{S,E\setminus S}(\tilde{J}^E_{E\setminus S,E\setminus S})^{-1}\tilde{h}_{E\setminus S,S} \\
&= (h^E_{S,S} + (K^S)^{-1}\hat{x}^S - (K^E_{S,S})^{-1}\hat{x}^E_S) - J^E_{S,E\setminus S}(J^E_{E\setminus S,E\setminus S})^{-1}h_{E\setminus S,S} \\
&= (K^S)^{-1}\hat{x}^S - \cancel{(K^E_{S,S})^{-1}\hat{x}^E_S} + \cancel{(h^E_{S,S} - J^E_{S,E\setminus S}(J^E_{E\setminus S,E\setminus S})^{-1}h_{E\setminus S,S})} \\
&= (K^S)^{-1}\hat{x}^S
\end{aligned} \tag{4.50}$$

Thus, $\tilde{x}^E_S = \tilde{K}^E_{S,S}(K^S)^{-1}\hat{x}^S = K^S(K^S)^{-1}\hat{x}^S = \hat{x}^S$. □

■ 4.4.2 Algorithm Specification

We now present a simple version of the iterative scaling method to minimize the regularized dual function for a Gaussian graphical model using a block decomposition based on $\mathcal{G}$. The input to this procedure is the initial decomposition $(h^\dagger, J^\dagger)$, where $J^\dagger$ represents a convex decomposition, and a specified collection of *update sets* $S(k)$ for $k = 1, \ldots, m$. We then iteratively optimize this decomposition using the following procedure.

For $t = 1, \ldots$ until convergence:
For $k = 1, \ldots, m$:

1. Let $S = S(k)$ below. For each $E \in \mathcal{G}(S)$ compute:

$$\begin{aligned}
\hat{h}^{S,E} &= h^E_S - J^E_{S,E\setminus S}(J^E_{E\setminus S,E\setminus S})^{-1}h^E_{E\setminus S} \\
\hat{J}^{S,E} &= J^E_{S,S} - J^E_{S,E\setminus S}(J^E_{E\setminus S,E\setminus S})^{-1}J^E_{E\setminus S,S}
\end{aligned}$$

   This is the information form of the marginal moments:

$$\begin{aligned}
\hat{x}^E_S &= ((J^E)^{-1}h^E)_S \\
K^E_{S,S} &= ((J^E)^{-1})_{S,S}.
\end{aligned}$$

2. Compute the average of these information forms:

$$\begin{aligned}
\bar{h}^S &= \frac{1}{|\mathcal{G}(S)|}\sum_{E\in\mathcal{G}(S)} \hat{h}^{S,E} \\
\bar{J}^S &= \frac{1}{|\mathcal{G}(S)|}\sum_{E\in\mathcal{G}(S)} \hat{J}^{S,E}
\end{aligned} \tag{4.51}$$

   Let $K^S = (\bar{J}^S)^{-1}$ and $\hat{x}^S = (\bar{J}^S)^{-1}\bar{h}^S$.

3. For each $E \in \mathcal{G}(S)$, update the edge potential (h^E, J^E) according to:

$$\begin{aligned} h_S^E &\leftarrow h_S^E + (\bar{h}^S - \hat{h}^{S,E}) \\ J_{S,S}^E &\leftarrow J_{S,S}^E + (\bar{J}^S - \hat{J}^{S,E})) \end{aligned}$$

The marginal moments on S now equal $(\hat{x}^S, K^S)$ in all edges $E \in \mathcal{G}(S)$.

The procedure is terminated once it is found that the marginal-matching conditions are approximately satisfied on all update sets. The procedure is equivalent to performing block coordinate-descent on the Lagrange multiplier representation of the dual function. Provided we include all maximal intersections of edges of $\mathcal{G}$ as update sets S in this procedure, this leads to the global minimum of the dual function. Upon convergence, the MAP estimate $\hat{x}$ is determined by the (consistent) local estimates $\hat{x}^E = (J^E)^{-1}h^E$ for all $E \in \mathcal{G}$.

■ 4.5 Multiscale Relaxations

In this section we propose a new multiscale approach to MAP estimation in Markov random fields. Although we develop this approach here for Gaussian models, the basic idea also applies for discrete MRFs. While our approach is certainly inspired by other multiscale methods, such as the multigrid approach to solving linear systems [205] or the renormalization group method in physics [129] and image processing [91], our formulation is quite different from these other multiscale methods. These methods essentially involve trying to construct coarse-scale approximations to the MRF, and using MAP estimates computed from these coarse-scale approximate models to provide an initial guess for the solution at a finer scale. Our method instead involves constructing a multiscale decomposition of the model, in which potentials are decomposed across scale as well as within each scale, and then minimizing the value of the relaxed MAP estimate over all multi-scale decomposition of the model. The purpose of this relaxation is similar to that of the multigrid and renormalization group methods. Iterative methods generally involve simple rules that propagate information locally within the graph. Using a multiscale representation of the model allows information to propagate through coarse scales, which improves the rate of convergence to global equilibrium. Also, in non-linear problems, such multiscale representations can help to avoid local minima. In our convex LR approach, we try to use the multiscale methods to reduce the duality gap. For Gaussian graphical models, this means that we obtain a broader class of multiscale convex-decomposable graphical models for which we can recover the optimal MAP estimate.

■ 4.5.1 The Multiscale Formulation

We now describe how to formulate a class of multiscale decompositions of a Gaussian MRF. We illustrate our general approach with a simple example seen in Figure 4.1

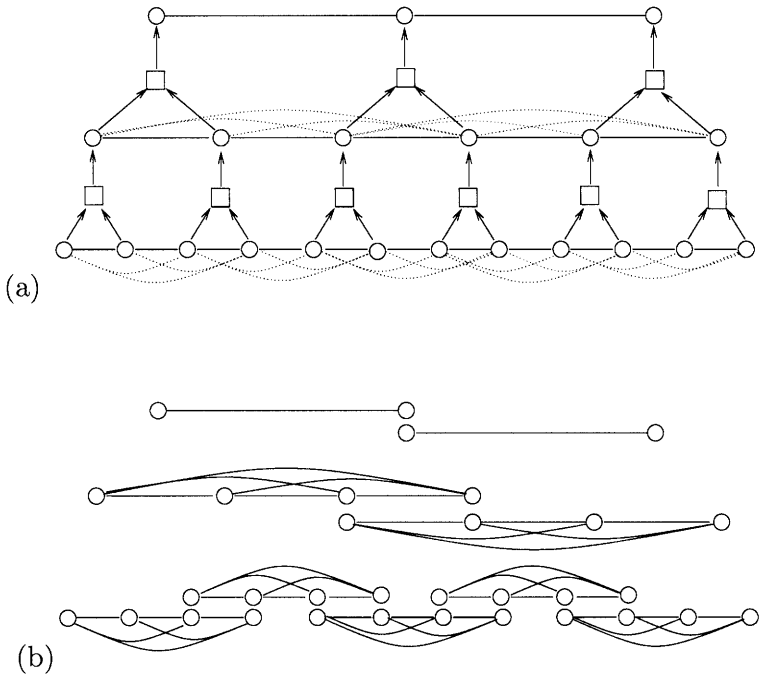

Figure 4.1. Illustration of multiscale LR method. (a) First, we define an equivalent multiscale model subject to cross-scale constraints. Relaxing these constraints leads to a set of single-scale models. (b) Next, each single scale is relaxed to a set of tractable subgraphs.

corresponding to a multiscale relaxation of a Markov chain. The decomposition method consists of two distinct steps.

First, as illustrated in Figure 4.1(a), we introduce coarse-scale representations of the fine-scale variables by defining the variables at each coarser scale to be a deterministic function of variables at the previous finer scale. Let $x^{(0)} = x$ denote the fine-scale variables corresponding the MRF that we wish to solve. Then, the coarse-scale variables are defined by:

$$x^{(s+1)} = A(s)x^{(s)} \tag{4.52}$$

for $s = 1, \ldots, N_s$. We define the matrices $A(s)$ to have a block-diagonal structure such that each coarse-scale variable at, say node v, represents a separate subset $E(v)$ of the fine-scale variables. However, our approach could also allow some overlap in the support of each coarse scale variable. This defines a set of cross-scale constraints between variables at adjacent scales:

$$x_v^{(s+1)} = A(s)_{v,E(v)} x_{E(v)}^{(s)} \tag{4.53}$$

In the figure, we represent these cross-scale constraints by square nodes. To allow interactions between coarse-scale variables (in the relaxation method described here) we must also introduce extra edges (the dotted ones in Figure 4.1(a)) between blocks

of nodes that have a (solid) edge between their summary nodes at the next coarser scale. The meaning of this will be made clear in a moment. One strategy for designing the coarsening matrix A is to define the coarse-scale variables to be sums or averages of the fine-scale variables. It is also possible to allow each coarse-scale variable to be vector-valued so as to allow each node variable to capture more information about the block of fine-scale variables that it summarizes. For instance, one could define A such that the course-scale variables represent a set of wavelet coefficients in a multiscale description of x similar to [65]. For Gaussian models it is necessary that the mapping between scales is linear. However, more generally, other non-linear maps could also be used.

Next, to obtain a tractable dual problem, we must decompose the MAP problem at each scale as depicted in Figure 4.1(b). This corresponds to the block decomposition method we have described previously for single-scale MRFs. In the example, we decompose the graph at each scale into its maximal cliques as shown in Figure 4.1(b). This defines a graph $\mathcal{G}^\dagger$ representing the final multiscale decomposition. At each scale, the variables $x^{(s)}$ are mapped to set of redundant auxiliary variables $(x^\dagger)^{(s)} = D(s)x^{(s)}$. Then, the complete set of auxiliary variables is $x^\dagger = ((x^\dagger)^{(s)}, s = 1, \ldots, N_s)$. We write the overall linear map from x to $x^\dagger$ as $x^\dagger = Dx$. Note, however, this now subsumes both the coarse-to-fine map described by the matrices $A(s)$ and the decomposition within each scale described by the matrices $D(s)$.

We now define the concept of *multiscale decomposition* of the fine-scale model specified by (h, J). Let $(h^\dagger, J^\dagger)$ be an information form defined on the variables $x^\dagger$ and with $J^\dagger$ restricted to have the block-diagonal structure indicated by the graph $\mathcal{G}^\dagger$ as seen in Figure 4.1(b). We say that this represents a *multiscale decomposition* of (h, J) if it holds that $D^T h^\dagger = h$ and $D^T J^\dagger D = J$. This ensures that $f^\dagger(Dx) = f(x)$ for all x. We also require that this decomposition is convex, which is equivalent to the condition $J^\dagger \succ 0$, so that the maximum of $f^\dagger$ is finite. Now, the story is otherwise the same as in the single-scale method. For each convex multiscale decomposition, we obtain an upper-bound on the value of the MAP problem by maximizing $f^\dagger(x^\dagger)$. Then, the dual problem is to minimize this upper-bound over all convex multiscale decompositions.

The variation over multiscale decompositions can also be equivalently described in terms of Lagrange multipliers associated with the constraints that define consistency of $x^\dagger$. However, these constraints now also include the cross-scale constraints, which allow us to trade-off potentials on coarse-scale variables with potentials on fine-scale variables. For example, relaxing the constraint $x_F = Ax_E$ creates an additional term in the relaxed objective of the form

$$\lambda^T(x_F - Ax_E) = \lambda^T x_F - (A^T\lambda)^T x_E, \tag{4.54}$$

which is equivalent to adding λ to $h_F^\dagger$ and subtracting $A^T\lambda$ from $h_E^\dagger$. Similarly, relaxing the second-order constraint $x_F x_F^T = Ax_E x_E^T A^T$ results in a term of the form

$$\mathrm{Tr}(\Lambda(x_F x_F^T - Ax_E x_E^T A^T)) = x_F^T \Lambda x_F - x_E^T A^T \Lambda A x_E, \tag{4.55}$$

which is equivalent to adding Λ to $J^{\dagger}_{F,F}$ and subtracting $A^T \Lambda A$ from $J^{\dagger}_{E,E}$. Using these kind of constructs, it is possible to create potentials between coarse-scale variables, provided we also include canceling terms in the fine-scale variables. This is the reason we add the dotted edges in the graph $\mathcal{G}^{\dagger}$, to accommodate these cancelling terms so as to allow the multiscale decomposition method to make use of interactions between the coarse-scale variables.

■ 4.5.2 Gaussian Iterative Scaling with General Linear Constraints

To minimize the multiscale Lagrangian dual function, we need a more general form of the iterative scaling algorithm. Previously, we have only allowed for linear constraints among blocks of the decomposition that enforce equality among copies of the same node (or subset of nodes) across the set of all blocks which contain this node (or subset of nodes). We now allow for a more general class of linear constraints among a set of blocks. Let $\mathcal{G}^{(k)} \subset \mathcal{G}$ denote some subset of blocks in the decomposition which are inter-related by linear constraints (we use the index $k = 1, \ldots, m$ to allow us to enumerate different sets of constraints in the following algorithm). Now, we allow for *arbitrary* linear constraints among these blocks, which may be specified by a set of matrices $A^{(k)}_E$, one for each $E \in \mathcal{G}^{(k)}$, and pairwise constraints between these blocks:

$$A^{(k)}_E x^E = A^{(k)}_F x^F \quad \text{for all} \quad E, F \in \mathcal{G}^{(k)} \tag{4.56}$$

In the previous version of the iterative scaling algorithm, we only allowed for linear constraints in which each $A^{(k)}_E$ selects a subset of variables in x^E corresponding to variables S in the intersection of a set of blocks. Now, we allow the procedure to enforce equality constraints among the blocks using any linear function of each block. This is general enough to also accommodate the inter-scale linear constraints introduced in the multiscale decomposition method. Relaxing these constraints then leads to a set of generalized moment-matching conditions for optimality in our regularized dual problem:

$$\begin{aligned} A^{(k)}_E \hat{x}^E &= A^{(k)}_F \hat{x}^F \text{ and} \\ A^{(k)}_E K^E (A^{(k)}_E)^T &= A^{(k)}_F K^F (A^{(k)}_F)^T \end{aligned}$$

for all k and $E, F \in \mathcal{G}^{(k)}$. That is, rather than matching means and covariances on subsets of nodes, we now match means and covariances of linear functions of subsets of nodes. The iterative scaling method we developed previously can be generalized to enforce these constraints. We omit the derivation because it is similar to the single-scale case.

Algorithm Given an initial decomposition $(h^{\dagger}, J^{\dagger})$ with respect to the set of blocks specified by $\mathcal{G}$ and linear constraints among these blocks specified by $\mathcal{G}^{(k)}$ (for $k = 1, \ldots, m$) and matrices $\{A^{(k)}_E, E \in \mathcal{G}^{(k)}\}$, we iteratively optimize the decomposition as follows.

For $t = 1, \ldots$ until convergence:
  For $k = 1, \ldots, m$:

1. For each $E \in \mathcal{G}^{(k)}$ calculate the moments:

$$\begin{aligned} \hat{x}^E &= A_E^{(k)}(J^E)^{-1}h^E \\ K^E &= A_E^{(k)}(J^E)^{-1}(A_E^{(k)})^T \end{aligned}$$

Also compute the information form of these statistics:

$$\begin{aligned} \hat{h}^E &= (K^E)^{-1}\hat{x}^E \\ \hat{J}^E &= (K^E)^{-1} \end{aligned}$$

2. Take the average of these information forms:

$$\begin{aligned} \bar{h}^{(k)} &= \frac{1}{|\mathcal{G}^{(k)}|}\sum_{E\in\mathcal{G}^{(k)}} \hat{h}^E \\ \bar{J}^{(k)} &= \frac{1}{|\mathcal{G}^{(k)}|}\sum_{E\in\mathcal{G}^{(k)}} \hat{J}^E \end{aligned}$$

3. For each $E \in \mathcal{G}^{(k)}$, update the edge potentials according to:

$$\begin{aligned} h^E &\leftarrow h^E + (A_E^{(k)})^T(\bar{h}^{(k)} - \hat{h}^E) \\ J^E &\leftarrow J^E + (A_E^{(k)})^T(\bar{J}^{(k)} - \hat{J}^E)A_E^{(k)} \end{aligned}$$

This algorithm is used to minimize the (regularized) dual function of the multiscale relaxation in Gaussian models. It may also be viewed as an iterative information projection algorithm enforcing the moment-matching conditions in the relaxed multiscale representation of the graphical model.

■ 4.6 Experimental Demonstrations

In this section, we provide a brief study of the performance of LR methods for solving Gaussian MAP estimation problems.

■ 4.6.1 LR in the Thin-Plate Model

We begin with a simple example using two block relaxations of a thin-plate model defined on a 64×64 grid. Recall (from Section 4.2) that the thin-plate model defines potential functions centered at each node x_i involving its four nearest neighbors $N(i)$:

$$f(x_i, x_{N(i)}) = -\tfrac{1}{2}(x_i - \tfrac{1}{4}\sum_{j\in N(i)} x_j)^2$$

We also define node-wise potentials $f(x_i) = -\frac{1}{2}\gamma x_i^2 + h_i x_i$ with $\gamma = .01$ and random $h_i \sim N(0,1)$ chosen independently at each node. This model is not pairwise-normalizable,

and we have found that both the pairwise and factor-graph versions of Gaussian belief propagation do not converge in this example. We apply two versions of the block LR method for this example. First, we use the set of 5-node blocks associated with the thin-plate potentials. This insures that the model is convex-decomposable with respect to this decomposition. We also try using larger 8×8 blocks, where blocks are shifted by increments of 4 pixels vertically and horizontally such that adjacent blocks overlap on either a 2×4, 4×2 or 2×2 block. The results of applying the Gaussian iterative scaling procedure for these two decompositions are shown in Figure 4.2. We use the maximum discrepancy in marginal moments (both means and covariances) between overlapping blocks to measure convergence in the Gaussian iterative scaling method. The method using 8×8 blocks converges more quickly both in terms of iterations (shown) and run-time. However, using even larger blocks, the cubic growth in computational complexity of the block-wise computations will eventually out-weight the advantage of using larger blocks. In this example, block sizes of about 8×8 led to the fastest convergence.

■ 4.6.2 Comparison to Belief Propagation and Gauss-Seidel

Thin-Membrane Model. We apply LR for two Gaussian models defined on a 50×50 grid with correlation lengths comparable to the size of the field. First, we use the *thin membrane* model, which encourages neighboring nodes to be similar by having potentials $f_{ij} = \frac{1}{2}(x_i - x_j)^2$ for each edge $\{i, j\} \in \mathcal{G}$. We split the 2-D model into vertical strips of narrow width K, which have overlap L (we vary K and set $L = 2$). We impose marginal agreement conditions in $K \times L$ blocks in these overlaps. The updates are done consecutively, from top to bottom blocks, from the left to the right strip. This corresponds to a subgraph decomposition similar to the one seen in Figure 3.3(f) (Chapter 3), where K specifies the width of the each strip and also controls the amount of "fill" edges we add along the boundary of each strip. A full update of all the blocks constitutes one iteration. We compare LR to loopy belief propagation (LBP). The LBP variances are underestimated by 21.5 percent (averaged over all nodes), while LR variances for $K = 8$ are overestimated by 16.1 percent. In Figure 4.3 (top) we show convergence of LR for several values of K, and compare it to LBP. The convergence of variances is similar to LBP, while for the means LR converges considerably faster. In addition, the means in LR converge faster than using block Gauss-Seidel on the same set of overlapping $K \times 50$ vertical strips.

Thin-Plate Model. Next, we use the *thin plate model*, which enforces that each node v is close to the average of its nearest neighbors $N(v)$ in the grid, and penalizes curvature. Gaussian belief propagation does not converge for this example. LR gives rather loose variance bounds for this more difficult case: for $K = 12$, it overestimates the variances by 75.4 percent. More importantly, it accelerates convergence of the means. In Figure 4.3 (bottom) we show convergence plots for means and variances, for several values of K. As K increases, the agreement is achieved faster, and for $K = 12$ agreement is achieved in under 13 iterations for both means and variances. We note that LR with $K = 4$ converges much faster for the means than block Gauss-Seidel.

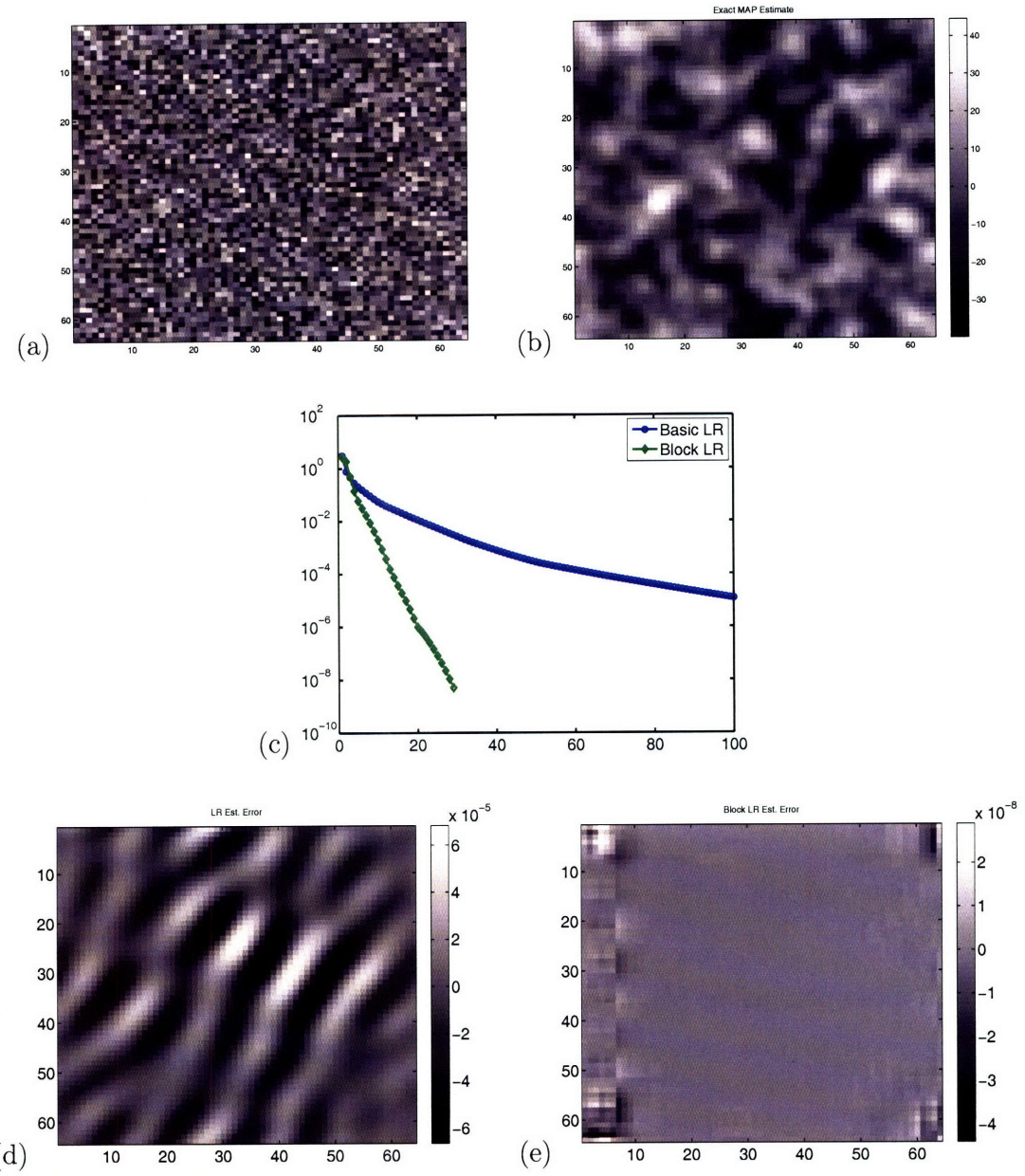

Figure 4.2. Results of applying LR to the thin-plate model. (a) The random field h. (b) The MAP estimate based on the (regularized) thin-plate model. (c) Plot showing convergence of the basic (edge-wise) and block (8×8 blocks) versions of LR, showing the maximum discrepancy in marginal moments (on a log-scale) versus the number of iterations. (d) and (e) show the errors in the estimates produced by these two methods, respectively using 5-node blocks and 8×8 blocks, after 100 iterations. The errors in the estimate $\hat{x}$ produced by the 8×8 block method are 3-4 orders of magnitude smaller than in the 5-node block method.

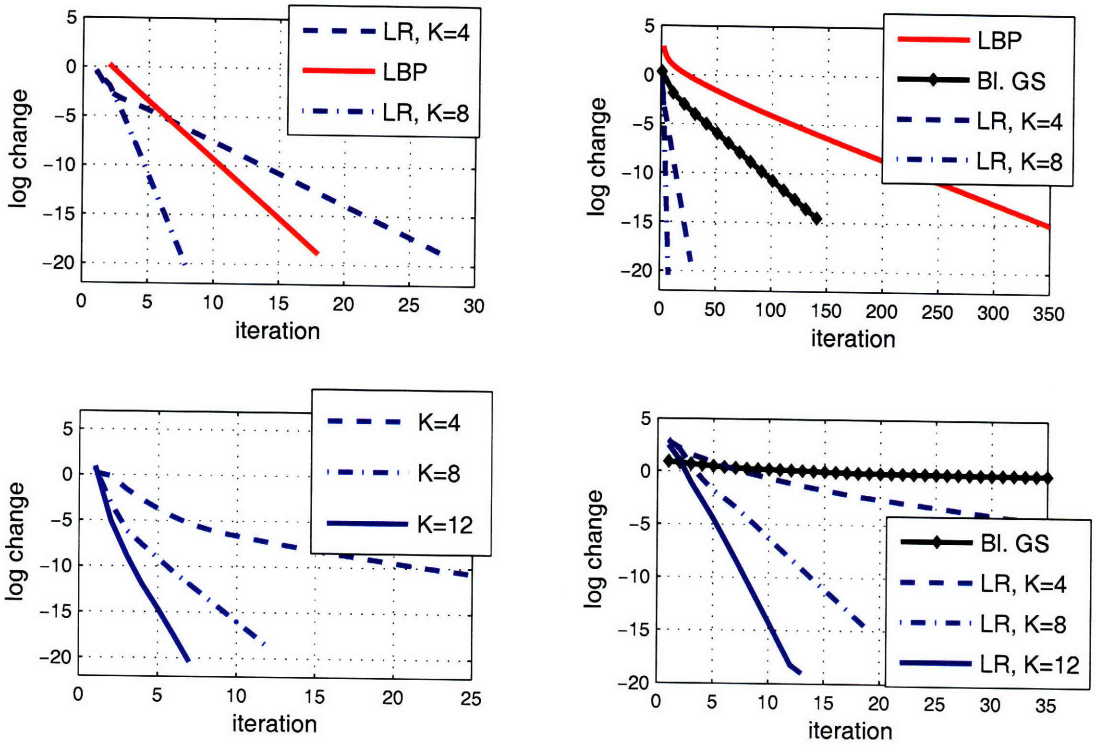

Figure 4.3. Convergence plots for variances (left) and means (right), in the thin-membrane model (top) and thin-plate model (bottom).

We also note that these subgraph decomposition methods converge faster than block decompositions using $K \times K$ blocks (such as in the thin-plate example at the beginning of the section).

■ 4.6.3 Examples using Multiscale Relaxations

1-D Example. Lastly, we perform two experiments using multiscale relaxations. First, we consider a 1-D thin-membrane model with 1024 nodes. It is defined to have a long correlation length comparable to the length of the field. Again using random $h_i \sim N(0,1)$ chosen independently at each node, we solve for the MAP estimates using three methods: a standard block Gauss-Seidel iteration using overlapping blocks of size 4; the (single-scale) Gaussian LR method with the same choice of blocks; and the multiscale LR method as seen in Figure 4.1 (merging two nodes at a time in the coarsening procedure and using blocks of size four to decompose the graph within each scale). We define the coarse-scale nodes to represent the average of the fine-scale nodes. The convergence of all three methods is shown in Figure 4.4. We see that the single-scale LR approach is moderately faster than block Gauss-Seidel, but introducing coarser-scales into the method leads to a significant speed-up in the rate of convergence.

2-D Example. Next, we try a 2-D multiscale example based on a 128×128 thin-

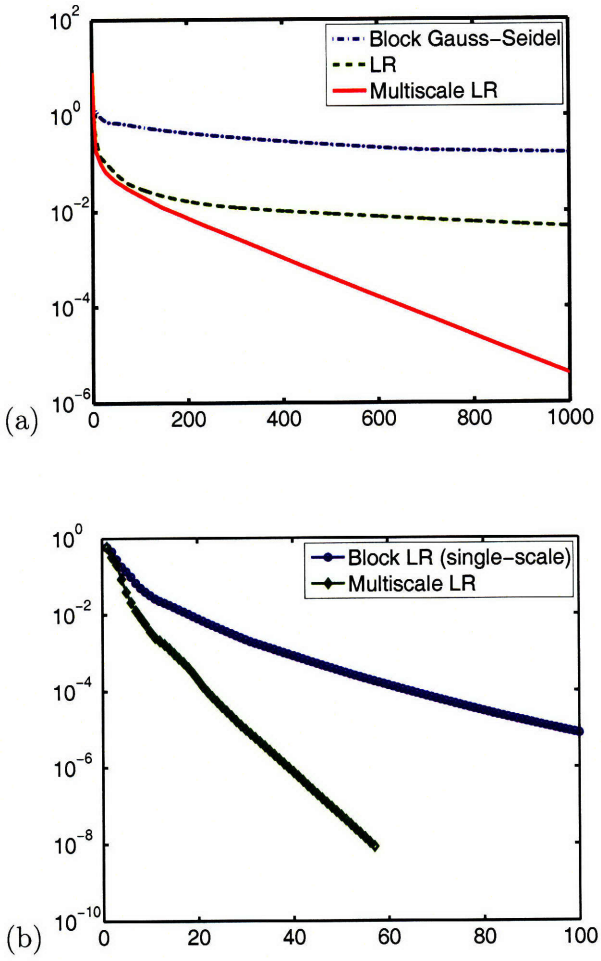

Figure 4.4. (a) Comparison of block LR (single-scale) versus multi-scale block LR and block Gauss-Seidel in the 1-D Gaussian model. (b) Comparison of (single-scale) block LR and multiscale block LR in a 128 × 128 thin-plate model. The convergence of LR is again measured by discrepancy in marginal moments, whereas we use the maximum residual error $\|h - J\hat{x}\|_\infty$ to measure convergence of Gauss-Seidel.

membrane model. Here, we compare the rate of convergence using the single-scale and multi-scale versions of LR. In the single-scale method we decompose the graph into 4×4 blocks shifted in increments of 2, such that blocks overlap on 2×4, 4×2 and 2×2 blocks. We use the same block decomposition at each scale in the multiscale approach, where each coarse-scale variable is defined to be the average of the four nodes within the 2×2 block beneath that node at the next finer scale. Again, we see a significant speed-up using the multi-scale method. We expect that the improvement of the multiscale method relative to the single-scale approach will become more pronounced in larger fields and with longer correlation lengths.

Chapter 5

Maximum Entropy Relaxation for Learning Graphical Models

5.1 Introduction

Inspired by the duality between maximum likelihood parameter estimation over exponential families and maximum entropy modeling, we propose a relaxed version of the maximum entropy principle for learning graphical models. Our main motivation for developing this method, which we refer to as *maximum entropy relaxation* (MER), is that it provides a convex optimization approach to learning *both* the graph structure and the potential specification of the model. However, it also provides a *robust* formulation for learning, one that results in good generalization performance when learning from limited sample data.

Motivation

Before presenting the details of our MER problem formulation, we first provide some intuition as to the motivation for our approach. In the maximum entropy approach to graphical modeling, the potential functions may be regarded as *Lagrange multipliers* associated with constraints on marginal distributions of the model (e.g., that they equal the empirically observed marginal distributions based on sample data). However, in this standard approach, one must specify *which* subsets of variables to constrain and it is this choice of the constraint set that determines the graphical structure of the maximum-entropy model. Intuitively, if one selects the "right" graph (one which captures the main interactions of the distribution being modeled) then we should expect that the remaining marginal distributions (on other subsets of nodes that are not cliques of this graph) should still come close to the correct marginal distributions. In other words, if we find that the maximum likelihood model over this graph still has some subsets of variables with marginal distribution that substantially deviate from what is seen in the sample data, then we have probably not found the right graph and should include additional constraints on those subsets of node variables. On the other hand, given that the sample distributions are only *approximations* to the correct marginals of the underlying distribution being observed, requiring that marginal distributions should *exactly* agree with sample distributions is perhaps too strict. One might instead

consider using relaxed constraints that allow some uncertainty in the specification of marginal distributions.

Based on these considerations, it then seems natural that, rather than imposing exact marginal matching only for certain selected subsets of nodes (and no constraints on other subsets of nodes), we should instead impose *approximate* marginal matching over *all* subsets of node (or just all subsets up to a certain size) and then rely on the maximum entropy principle to decide which among these constraints to enforce with potential functions in the graphical model. For instance, considering just pairs of nodes, we may require that *all* pairs of nodes should have pairwise marginal distributions that are close to their sample distributions. We take "close" to mean that the relative entropy (the natural information-theoretic measure of divergence between probability distributions) between the pairwise marginals of our model and the sample marginals is below some specified tolerance. Then, we pose learning as solving for the least informative probability distribution, as measured by entropy, subject to these approximate marginal-matching constraints. Very importantly, recalling the role that potentials play as Lagrange multipliers in the maximum entropy method, we can see that in this relaxed formulation it is those subsets of *active* marginal constraints (that are satisfied with equality) that result in non-zero potentials in the maximum entropy distribution and thereby determine the graphical structure of the learned model. Thus, the selection of which edges to include is decided *adaptively* through solving a convex optimization problem. In this manner, we rely on the maximum entropy principle to decide which edges to include in the graph so as to obtain a good fit to all pairwise marginal distributions of the sample data. For an appropriate choice of the tolerances on marginal divergence (e.g., chosen to reflect the expected divergence between the unknown true marginal distributions and the sample distributions) we expect that enforcing these approximate marginal-matching conditions over the edges of the correct graphical structure will typically cause most of the other constraints to be satisfied so that the learned graph should provide a good estimate to the actual structure of the probability distribution that generated the sample data.

■ 5.2 Mathematical Formulation

Now that we have described the basic idea behind our approach, we present the mathematical formulation of MER. We specify MER with respect to the moment parameterization of an exponential family graphical model, as discussed in Chapter 2, such as the Boltzmann machine, Ising model, general discrete Markov random field (using an overparameterized representation) or a Gaussian graphical model. Recall that these models are probability distributions of the form $P(x) \propto \exp\{\theta^T \phi(x)\}$ where $x = (x_1, \ldots, x_n)$ are the node variables and the graph structure is determined by the feature vector $\phi(x)$, which consists of edge-wise features that are functions of subsets of variables corresponding to (generalized) edges of the graph. The parameter vector θ then corresponds to the potential specification of the model and the moment vector $\eta = \mathbb{E}_\theta\{\phi\}$

corresponds to the marginal specifications.

In the following statement of the MER problem, we formulate the problem over a generalized graph $\tilde{\mathcal{G}}$. This graph $\tilde{\mathcal{G}}$ may be very dense. In fact, $\tilde{\mathcal{G}}$ may even be a fully connected graph, for example, the graph including all subsets of s or fewer nodes as (generalized) edges. The reason for this specification is that we rely upon the model thinning property of MER (to be discussed further in Section 5.2.2) to select a sparse subgraph of $\tilde{\mathcal{G}}$ that provides a good fit to the sample data. However, to solve this "full" MER problem on a dense graph, we actually solve a sequence of subproblems based on subgraphs of the full graph. Each subproblem of that procedure is also formulated as solving an MER problem, but where $\tilde{\mathcal{G}}$ is then a tractable (thin) graph (a subgraph of the intractable full graph). This procedure is be discussed in more detail in Sections 5.3.1 and 5.3.2.

■ 5.2.1 MER Problem Statement

Given a (generalized) graph $\tilde{\mathcal{G}}$ and specification of an exponential family of graphical models based on $\tilde{\mathcal{G}}$, we solve the following convex optimization problem over the moment parameters of this model subject to approximate marginal-matching constraints on all edges of $\tilde{\mathcal{G}}$:

$$\begin{array}{lll} & \text{maximize} & H(\eta) \\ \text{(MER)} & \text{subject to} & \eta \in \mathcal{M} \\ & & d_H(\eta_{[E]}, \tilde{\eta}_{[E]}) \leq \delta_E \text{ for all } E \in \tilde{\mathcal{G}} \end{array} \tag{5.1}$$

The problem variables are the moment parameters η of the model and we seek to maximize the global entropy $H(\eta)$. In the minimal parameterization of the exponential family (e.g., the Ising, Boltzmann and Gaussian models) the marginal distribution $P(x_E)$ is determined by the subset of moment parameters $\eta_{[E]} = (\eta_\alpha, \alpha \in [E])$ where $[E] \subset \mathcal{I}$ denotes the subset of features $\phi = (\phi_\alpha, \alpha \in \mathcal{I})$ which depend only on variables within edge E. For instance, in the (generalized) Boltzmann and Ising models, $\eta_{[E]}$ actually represents the set of moments $(\eta_A, A \subseteq E)$ where $\eta_A = \mathbb{E}\{\prod_{v \in A} x_A\}$. In Gaussian models, it is sufficient to consider just the zero-mean case of MER.[1] Then, $\eta_{[E]}$ represents the moments: $\eta_v = \mathbb{E}\{x_v^2\}$ for all $v \in E$ and $\eta_{uv} = \mathbb{E}\{x_u x_v\}$ for all $\{u, v\} \subset E$. In the over-parameterized representation of a general discrete model, $\eta_{[E]}$ corresponds directly to the marginal $P(x_E)$ (specified for all values of x_E). We write $\tilde{\eta}$ to denote the sample moments (the sample marginals in the over-parameterized model). Thus, the constraints that the marginal distributions of the model should approximately agree with the sample distributions of the data is encoded as $d_H(\eta_{[E]}, \tilde{\eta}_{[E]}) \leq \delta_E$ where d_H denotes the relative entropy as a function of moment parameters (the Bregman distance based on the marginal entropy $H(\eta_{[E]})$) and $\delta_E > 0$ specifies our tolerance to marginal divergence on this edge. We impose these constraints over all edges of

[1] For Gaussian models, the MER solution has the same means $\hat{x}$ as the sample data. Hence, we can instead consider the problem of learning a zero-mean Gaussian model for the variable $x' = x - \tilde{x}$, where $\tilde{x}$ is the sample-mean, and solve MER in the family of zero-mean Gaussians.

$\tilde{\mathcal{G}}$. Note that we also have the constraint that $\eta \in \mathcal{M}$, that is, that η represents a valid (realizable) set of moment of our model.[2] This is essentially implied by using the entropy $H(\eta)$ as our objective function since it is defined only for points within $\mathcal{M}$. Thus, MER is defined by our choice of exponential family, the graph $\tilde{\mathcal{G}}$, the tolerances δ and the sample moments $\tilde{\eta}$.

Comments about MER

We now summarize some important properties of the MER optimization problem. First, it is a convex optimization problem, maximizing the concave function $H(\eta)$ (or, equivalently, minimizing the convex function $-H(\eta)$) over the convex set of feasible moment vectors. To show that the feasible set of MER is convex, we recall that $\mathcal{M}$ is convex and each of the constraint functions $d_H(\eta_{[E]}, \tilde{\eta}_{[E]})$ is a convex function of the first argument $\eta_{[E]}$ (for each fixed value of $\tilde{\eta}_{[E]}$). Thus, the set of all $\eta \in \mathcal{M}$ which satisfy the constraint $d_H(\eta_{[E]}, \tilde{\eta}_{[E]}) \leq \delta_E$ is convex for each $E \in \tilde{\mathcal{G}}$ and the feasible set of MER is an intersection of convex sets and is therefore convex. Assuming $\tilde{\eta} \in \mathcal{M}$, that is, that the sample moments are actually realized by some (finite) $\tilde{\theta} \in \mathbb{R}^{|\mathcal{I}|}$, then the MER problem is feasible for all $\delta \geq 0$. If $\delta > 0$, MER is *strictly* feasible, that is, there exists an $\eta \in \mathcal{M}$ for which $d_H(\eta_{[E]}, \tilde{\eta}_{[E]})$ is *strictly* less than δ_E for all $E \in \tilde{\mathcal{G}}$. In discrete models, the set $\mathcal{M}$ itself is bounded so that the feasible set of MER is also bounded. In the Gaussian model, $\mathcal{M}$ is unbounded, but the feasible set of MER is bounded if every node $v \in V$ is included by at least one constrained edge $E \in \tilde{\mathcal{G}}$ (with $\delta_E < \infty$). Boundedness of the MER feasible set, together with the fact that boundary points of $\mathcal{M}$ cannot be local maxima of entropy,[3] ensures that there exists $\hat{\eta} \in \mathcal{M}$ which achieves the maximum value of the MER problem (that is, there exists a solution within $\mathcal{M}$). Moreover, because $H(\eta)$ is strictly concave over $\mathcal{M}$, this MER solution $\hat{\eta}$ is unique.

Assumptions on Sample Moments

In most cases, it is safe to assume that the sample moments $\tilde{\eta}$ are non-degenerate, that is, $\tilde{\eta}$ is an interior point of $\mathcal{M}$. This means that there exists some (finite) $\tilde{\theta}$ such that $\Lambda(\tilde{\theta}) = \tilde{\eta}$. We will assume this to be the case in our analysis and methods developed in later sections. However, in this section, we also consider how our approach can be modified to cope with either boundary points $\tilde{\eta} \in \partial\mathcal{M}$ or collections of inconsistent moments $\{\tilde{\eta}^{[E]}, E \in \tilde{\mathcal{G}}\}$ such as may occur if we only have incomplete samples of x.

[2] In the case of the over-parameterized representation of the general discrete model, one should keep in mind that the constraint $\eta \in \mathcal{M}$ implies consistency among marginal distributions, such that overlapping subsets must have marginal distributions that give the same marginal distribution on their intersection.

[3] This can be seen easily for the discrete models, where $\mathcal{M} = \{\Lambda(\theta), \theta \in \mathbb{R}^{|\mathcal{I}|}\}$ is a bounded, open subset of $\mathcal{R}^{|\mathcal{I}|}$ whose boundary points correspond to infinitely large θ. Recalling that $\nabla H(\eta) = -\Lambda^{-1}(\eta)$, we see that the gradient of entropy becomes infinitely large near the boundary of $\mathcal{M}$ and must point away from the boundary. Similarly, boundary points of $\mathcal{M}$ in the Gaussian model correspond to singular covariance matrices, and hence $H(\eta)$ goes to $-\infty$ as η approaches the boundary.

First, we note that if the sample moments $\tilde{\eta}$ are computed from a set of *complete* samples of x, such that all variables are observed in each sample, it must hold that they are consistent (that is, $\tilde{\eta} \in \bar{\mathcal{M}} \triangleq \mathcal{M} \cup \partial\mathcal{M}$). This holds because $\bar{\mathcal{M}}$ is the convex hull of the set $\{\phi(x), x \in \mathbb{X}^n\}$. Then, we need only deal with the problem of boundary points.

Dealing with Boundary Points of $\mathcal{M}$. It is possible that the sample moments $\tilde{\eta}$ obtained from samples $x^{(1)}, \ldots, x^{(n_s)} \sim P(x)$ leads to a point on the boundary of the set $\mathcal{M}$. This will occur if the distribution $P(x)$ being sampled is itself a degenerate distribution such as a discrete probability distribution in which some values of x have zero probability. It can also happen that $P(x)$ is not degenerate, but $\tilde{\eta}$ is still a boundary point. This happens in discrete models if any sample marginal $\tilde{P}(x_E)$, on some edge $E \in \tilde{\mathcal{G}}$, has zero probability for some values of x_E that did not occur in any of the samples. However, if the marginals $P(x_E)$ of $P(x)$ over edges $E \in \tilde{\mathcal{G}}$ are not degenerate, then the probability that $\tilde{\eta}$ is a boundary point goes to zero exponentially fast with sample size. Thus, it is usually safe to assume that $\tilde{\eta} \in \mathcal{M}$.

To guard against the exceptional case that $\tilde{\eta}$ is a boundary point (if $P(x)$ has degenerate marginals or if the sample size is small), we suggest the following simple modification of our approach. Rather than using the actual sample moments $\tilde{\eta}$ in MER, we may instead use a slightly perturbed set of moments:

$$\tilde{\eta}(\varepsilon) = (1 - \varepsilon)\tilde{\eta} + \varepsilon\eta_0$$

where $\eta_0 \in \mathcal{M}$ is some default model (e.g. the maximum-entropy distribution $\eta_0 = \Lambda(0)$ in discrete graphical models) and $\varepsilon \in (0, 1)$ is small (e.g., $\varepsilon = .001$). This ensures that $\tilde{\eta}(\varepsilon) \in \mathcal{M}$ so that we can solve the perturbed MER problem using the methods of this chapter so as to obtain the solution $\hat{\eta}(\varepsilon) \in \mathcal{M}$ of the perturbed MER problem, which provides an approximate solution of the MER problem based on $\tilde{\eta}$. Our reason for suggesting this method is that if $\tilde{\eta}$ is degenerate, it does present technical difficulties in our algorithms and would also complicate our analysis. This is because the MER relative-entropy constraints would force the MER solution to also be a boundary point of $\mathcal{M}$. Using the method above, it is simple enough to obtain an approximate solution $\hat{\eta}(\varepsilon) \in \mathcal{M}$ which approaches the desired MER solution $\hat{\eta} \in \partial\mathcal{M}$ as $\varepsilon \to 0$.

Dealing with Inconsistent Moments. Suppose now that the sample moments are instead computed from a set of incomplete samples of x. For instance, suppose that for sample s we only observe the variables $x_{V(s)}$ for some $V(s) \subset V$, but with every edge $E \in \tilde{\mathcal{G}}$ being observed in some sample. Then, there is no simple method to summarize all of these observations via a single consistent moment vector. To handle this problem, an extension of the version of MER presented in this thesis was developed [46]. In that approach, each MER edge constraint is replaced by the constraint: $d_H(\eta_{[E]}, \tilde{\eta}^{[E]}) \leq \delta_E$ where $\tilde{\eta}^{[E]}$ is specified *independently* for each edge $E \in \tilde{\mathcal{G}}$. This $\tilde{\eta}^{[E]}$ is defined by the sample average of the feature $\phi_{[E]}(x_E)$ over all samples such that $E \subset V(s)$. Then, if we make the edge tolerances δ_E large enough, the MER problem is feasible and we can solve it using similar algorithms as developed in this chapter. We refer the reader to [46] for further details of that approach.

Relation to Maximum-Likelihood and Information Projections

It is also worth noting that if we take $\delta_E = 0$ for all $E \in \tilde{\mathcal{G}}$, then the MER solution is $\hat{\eta} = \tilde{\eta}$, which is also the maximum likelihood model over the exponential family defined on $\tilde{\mathcal{G}}$. Similarly, we may obtain the information projection to some subgraph $\mathcal{G} \subset \tilde{\mathcal{G}}$ by defining $\delta_E = 0$ for $E \in \mathcal{G}$ and $\delta_E = +\infty$ for $E \notin \mathcal{G}$, which is equivalent to imposing exact marginal-matching constraints on all edges of $\mathcal{G}$ and completely relaxing (removing) the marginal-matching conditions on all other edges of $\tilde{\mathcal{G}}$. Then, the MER solution $\hat{\eta}$ is equal to the information projection (m-projection) of $\tilde{\eta}$ to the sub-family of Markov models on $\mathcal{G}$. Thus, MER with non-zero values for the δ parameters may be regarded as a relaxation of the usual concept of information projection to a graphical model, with the parameters δ controlling the level of relaxation.

■ 5.2.2 Model Thinning Property

We have already discussed the role that potential functions play as Lagrange multipliers in the standard maximum-entropy approach to graphical modeling. In MER, we have replaced hard marginal-matching constraints by a set of relaxed inequality constraints on marginal divergences. Based on the notion of complementary slackness, this suggests the following result concerning the Markov structure of the final MER solution:

Proposition 5.2.1 (MER Model Thinning). *Let $\hat{\eta}$ denote the MER solution and let $\hat{\mathcal{G}}$ denote the set of* active *edges, where $d_H(\eta_{[E]}, \tilde{\eta}_{[E]}) = \delta_E$. Then, the corresponding potential parameters $\hat{\theta} \triangleq \Lambda^{-1}(\hat{\eta})$ are sparse with respect to $\hat{\mathcal{G}}$, such that $\hat{\theta}_\alpha = 0$ unless $\alpha \in [E]$ for some $E \in \hat{\mathcal{G}}$. Thus, the MER solution is Markov with respect to the subgraph $\hat{\mathcal{G}} \subset \tilde{\mathcal{G}}$ (which, of course, also implies that it is Markov on $\tilde{\mathcal{G}}$).*

Proof. There is a Lagrange multiplier λ_E associated with each edge's approximate marginal-matching constraint. The Karush-Kuhn-Tucker (KKT) necessary conditions for $\hat{\eta}$ to be the optimal solution of the MER problem is that there exists a set of Lagrange multipliers λ such that: (i) $\lambda_E \geq 0$ for all $E \in \tilde{\mathcal{G}}$, (ii) $d_H(\hat{\eta}_{[E]}, \tilde{\eta}_{[E]}) \leq \delta_E$ for all $E \in \tilde{\mathcal{G}}$, (iii) $\sum_{E \in \tilde{\mathcal{G}}} \lambda_E (d_H(\hat{\eta}_{[E]}, \tilde{\eta}_{[E]}) - \delta_E) = 0$ and

$$\text{(iv)} \quad \nabla \left\{ H(\hat{\eta}) - \sum_{E \in \tilde{\mathcal{G}}} \lambda_E \, d_H(\hat{\eta}_{[E]}, \tilde{\eta}_{[E]}) \right\} = 0. \tag{5.2}$$

where the gradient ∇ is taken with respect to $\hat{\eta}$. Recalling the gradient relations $\nabla H(\hat{\eta}) = -\Lambda^{-1}(\hat{\eta}) \triangleq \hat{\theta}$ and $\nabla d_H(\hat{\eta}, \tilde{\eta}) = \Lambda^{-1}(\tilde{\eta}) - \Lambda^{-1}(\hat{\eta})$, condition (iv) becomes:

$$\hat{\theta} = \sum_{E \in \tilde{\mathcal{G}}} \lambda_E \left[\Lambda_E^{-1}(\tilde{\eta}_{[E]}) - \Lambda_E^{-1}(\hat{\eta}_{[E]}) \right]_{\mathcal{I}} \tag{5.3}$$

Note that each marginal constraint only depends on a subset of moment parameters, so that the gradient of each marginal divergence is restricted to this subset of moment

parameters (it is zero for the other parameters). Also, we note that the KKT conditions (i)-(iii) imply that $\lambda_E = 0$ for all inactive edge constraints, where $d_H(\hat{\eta}_{[E]}, \tilde{\eta}_{[E]}) < \delta_E$. Thus,

$$\hat{\theta} = \sum_{E \in \hat{\mathcal{G}}} \lambda_E \left[\Lambda_E^{-1}(\tilde{\eta}_{[E]}) - \Lambda_E^{-1}(\hat{\eta}_{[E]}) \right]_{\mathcal{I}} \tag{5.4}$$

From this, we see that $\hat{\theta}_\alpha = 0$ unless $\alpha \in [E]$ for some edge of $\hat{\mathcal{G}}$. Finally, sparsity of $\hat{\theta}$ with respect to $\mathcal{G}$ implies

$$P(x) \propto \exp \left\{ \sum_{\alpha \in \mathcal{I}: \alpha \in [E], E \in \hat{\mathcal{G}}} \hat{\theta}_\alpha \phi_\alpha(x_\alpha) \right\},$$

which can be factored in terms of potentials defined only on the cliques of $\hat{\mathcal{G}}$ and is therefore Markov with respect to $\hat{\mathcal{G}}$. □

This is the basic property of MER that we rely on to obtain a sparse graphical model (even though the graph $\tilde{\mathcal{G}}$ may have been more dense or even fully-connected). Although we formulate MER in terms of the moment parameters η over the exponential family defined on $\tilde{\mathcal{G}}$, ultimately it is this graph subgraph $\hat{\mathcal{G}}$ and the corresponding non-zero parameters $\hat{\theta}$ defined over this graph that we seek to determine when solving MER.

■ 5.2.3 Selecting the Relaxation Parameters

Given the sample moments $\tilde{\eta}$, based on M samples $\tilde{x}^{(1)}, \ldots, \tilde{x}^{(M)} \sim P$, we now consider how to set the relaxation parameters δ so as to estimate Markov structure of the probability distribution P. Let $\eta = \mathbb{E}_P\{\phi\}$ denote the moments with respect to P and let $\tilde{\eta} = \frac{1}{M} \sum_{s=1}^{M} \phi(\tilde{x}^{(s)})$ denote the sample moments.

Estimating the Divergence

To begin with, we consider an estimate of the marginal divergence $d_H(\eta_{[E]}, \tilde{\eta}_{[E]})$. Following the analysis of Akaike [2], one can obtain an approximation for the *expected value* of the divergence between the actual moments η and the moments $\tilde{\eta}$ based on random samples. Assuming a minimal representation of the exponential family (with linearly independent sufficient statistics), the expected divergence is approximated as

$$\mathbb{E}_P\{d_H(\eta_{[E]}, \tilde{\eta}_{[E]})\} \approx \frac{\dim(E)}{M}, \tag{5.5}$$

where $\dim(E)$ denotes the number of model parameters (sufficient statistics) associated with edge E, determined by the number of sufficient statistics ϕ that depend only on variables x_E. This result is derived using the central limit theorem, which essentially asserts that $\tilde{\eta}_{[E]}$ is approximately Gaussian distributed with mean $\eta_{[E]}$ and covariance $G_E(\eta_{[E]})/M$, where $G(\eta_{[E]})$ is the Fisher information matrix, and the second-order

Taylor-series approximation of the divergence, which also depends on the Fisher information matrix $G(\eta_{[E]})$. It shows that the modeling error increases with the model order, which Akaike uses to derive his model selection criterion. For example, in the general binary-variable model (using either the Ising or Boltzmann representations and including interactions on all subsets of edge E) this "local" model complexity measure is equal to the number of non-empty subsets of E:

$$\dim(E) = 2^{|E|} - 1 \tag{5.6}$$

Here, $|E|$ represent the cardinality of the set E, that is, the number of nodes within edge E. In the zero-mean Gaussian model, which has features x_v^2 for each variable and $x_u x_v$ for each edge $\{u, v\}$, the model complexity on a constraint edge $E \in \mathcal{G}$ is:

$$\dim(E) = |E| + \binom{|E|}{2} \tag{5.7}$$

This analysis suggest setting the δ parameters as

$$\delta_E = \gamma \frac{\dim(E)}{M}, \tag{5.8}$$

so that the level of relaxation in MER reflects the typical accuracy of the the moments $\tilde{\eta}$. The parameter $\gamma > 1$ is included to allow the sparsity of the MER solution to be adjusted. Increasing values of γ correspond to higher levels of relaxation, which tend to produce sparser graphical models in the MER method.

Large-Deviations Analysis

Alternatively, we may instead consider the probability that the marginal constraint is violated and set δ so as to ensure that this probability becomes small as either the sample size or as the size of an edge becomes large. The idea here is that we want to make sure that we set the δ's large enough so that the true moments are contained by the MER feasible set with high probability. In discrete variable models, we may use Sanov's theorem [59] to obtain the following upper-bound of the probability of error:

$$P(d_H(\eta_{[E]}, \tilde{\eta}_{[E]}) \geq \delta_E) \leq (1 + M)^{\mathbb{X}^{|E|}} \exp\{-M\delta_E\} \tag{5.9}$$

To control the probability of error, we may specify an arbitrary function $\varepsilon(M, k)$ of the sample size M and the size of an edge $k = |E|$, and then choose δ_E as

$$\delta_E = \frac{1}{M}\left\{|\mathbb{X}|^{|E|} \log(1 + M) - \log \varepsilon(M, |E|)\right\} \tag{5.10}$$

so as to ensure that the probability of error on edge E is less than $\varepsilon(M, |E|)$. Later, in the experiments section, we show some examples where we have used:

$$\varepsilon(M, k) = \varepsilon_0 \, M^{-\zeta} \, 2^{-k} \binom{n}{k}^{-1} \tag{5.11}$$

The parameters $\varepsilon_0 \in (0, 1]$ and $\zeta > 0$ allow us to adjust adjust the sparsity of the MER solution. Typical values for these parameters are $\varepsilon_0 = \zeta = 1$. Smaller values of ε_0 tend to produce sparser graphical models with fewer edges. Larger values of ζ cause the δ's to decrease more quickly with sample size. We normalize by $2^k\binom{n}{k}$ to ensure that smaller edges dominate the MER solution. This is also motivated by the observation that the number of edges of size k grows as $\binom{n}{k}$ such that division by 2^k ensures that the total probability of error (for all k) stays bounded (using the union bound and $\sum_{k=1}^{n} 2^{-k} < 1$). Thus, the total probability of error is bounded by $\varepsilon_0 M^{-\zeta}$.

■ 5.3 Algorithms for Solving MER

■ 5.3.1 MER Boot-Strap Method

If we are to directly solve the MER problem on the graph $\tilde{\mathcal{G}}$, we must be able to compute the entropy function $H(\eta)$ over the exponential family of graphical models defined on this graph and also be able to check if a given η is in fact realizable by this model. If the graph $\tilde{\mathcal{G}}$ is not a thin (low treewidth) graph, then these calculations are intractable.[4] Thus, it would appear that it is intractable to solve MER on a very dense graph. However, as discussed in the previous sections, our goal in formulating the MER problem over $\tilde{\mathcal{G}}$ is actually to find a *sparse* subgraph $\hat{\mathcal{G}} \subset \tilde{\mathcal{G}}$ that still provides a good fit to the sample moments $\tilde{\eta}$. This suggests that, if the solution to MER is indeed very sparse, then it may still be possible to find the optimal MER solution with respect to the intractable graph $\tilde{\mathcal{G}}$. In this section, we describe our "bootstrap" approach to find the graph $\hat{\mathcal{G}}$ without ever having to perform intractable inference calculations in the full exponential family model defined by $\tilde{\mathcal{G}}$. The basic idea is to build up the graph incrementally, adding a few edges at a time, until the constraints on all the other edges of $\tilde{\mathcal{G}}$ (that have not yet been included) are found to be already satisfied. The key point one must appreciate here is that, once this is done, the solution obtained also determines the solution of the MER problem on the full graph $\tilde{\mathcal{G}}$. To make this simple but important point clear, we formalize it in the following proposition:

Proposition 5.3.1 (Embedded MER). *Let $\mathcal{G}$ be a subgraph of $\tilde{\mathcal{G}}$ and let $\hat{\eta}_{\mathcal{G}}$ denote the solution of the MER problem based on $\mathcal{G}$:*

$$(\textit{MER-}\mathcal{G}) \qquad \begin{array}{ll} \textit{maximize} & H(\eta_{\mathcal{G}}) \\ \textit{subject to} & \eta_{\mathcal{G}} \in \mathcal{M}(\mathcal{G}) \\ & d_H(\eta_{[E]}, \tilde{\eta}_{[E]}) \leq \delta_E \textit{ for all } E \in \mathcal{G} \end{array} \tag{5.12}$$

Note, this differs from the MER problem based on $\tilde{\mathcal{G}}$ in two ways: (1) we use the reduced moment parameterization $\eta_{\mathcal{G}}$ corresponding to the subset of moments η that are defined within $\mathcal{G}$, (2) we now impose approximate marginal-matching constraints only over the (generalized) edges present in $\mathcal{G}$, that is, omitting the constraints on edges of $\tilde{\mathcal{G}}$ that are not included in $\mathcal{G}$.

[4] In fact, if the graph $\tilde{\mathcal{G}}$ is not chordal, $H(\eta)$ may not even have an explicit closed form formula.

Given $\hat{\eta}_{\mathcal{G}}$, let $\hat{\eta} \in \mathcal{M}$ be defined as follows: (i) define $\hat{\theta}_{\mathcal{G}} = \Lambda_{\mathcal{G}}^{-1}(\hat{\eta}_{\mathcal{G}})$, (ii) zero-pad $\hat{\theta}_{\mathcal{G}}$ to obtain $\hat{\theta} \in \mathcal{M}$ and (iii) define $\hat{\eta} = \Lambda(\hat{\theta})$. Then,

1. *$\hat{\eta}$ is the* maximum-entropy completion *of $\hat{\eta}_{\mathcal{G}}$, that is, $\hat{\eta}$ solves the problem:*

$$\begin{array}{ll} \textit{maximize} & H(\eta) \\ \textit{subject to} & \eta \in \mathcal{M} \\ & \eta_\alpha = \hat{\eta}_\alpha, \quad \textit{for all} \quad \alpha \in \mathcal{I}(\mathcal{G}) \end{array} \tag{5.13}$$

   where $\mathcal{I}(\mathcal{G})$ denotes the subset of moments corresponding to edges of $\mathcal{G}$.

2. *If $\hat{\eta}$ satisfies the remaining MER constraints, that is, if for all edges $E \in \tilde{\mathcal{G}} \setminus \mathcal{G}$ it holds that $d_H(\hat{\eta}_{[E]}, \tilde{\eta}_{[E]}) \leq \delta_E$, then the maximum-entropy completion $\hat{\eta}$ is also the solution of the MER problem defined on $\tilde{\mathcal{G}}$.*

3. *Now let $\hat{\eta}$ denote the MER solution with respect to $\tilde{\mathcal{G}}$. If it results in a sparse graph $\hat{\mathcal{G}}$, then this solution can be obtained by solving MER-$\mathcal{G}$, as described above, over any $\mathcal{G}$ such that $\hat{\mathcal{G}} \subset \mathcal{G} \subset \tilde{\mathcal{G}}$. Any edges $E \in \mathcal{G} \setminus \hat{\mathcal{G}}$ must be inactive in the MER solution based on $\mathcal{G}$.*

With this principle in mind, we now propose a boot-strap method to find a graph $\mathcal{G}$ which includes the graph $\hat{\mathcal{G}}$ as a subgraph. The basic idea is to start with a simple graph and then gradually grow the graph by adding edges corresponding to violated constraints. Rather than add all such violated constraints at each step, we instead include the m edges with the largest constraint violations. We begin with the disconnected graph $\mathcal{G}^{(0)}$, comprised of singleton edges $E = \{v\}$ for all vertices $v \in V$, and then iterate the following procedure until either the MER solution is found or until the graph becomes intractable:

MER "Bootstrap" Algorithm For $k = 1, 2, \ldots,$

1. Solve the reduced MER problem over the graph $\mathcal{G}^{(k)}$, yielding the solution $\hat{\eta}^{(k)}$ (defined over $\mathcal{G}^{(k)}$).

2. Using the solution $\hat{\eta}^{(k)}$, check to see if any of the approximate marginal-matching constraints are violated. Note, this involves computing other moments corresponding to edges of $\tilde{\mathcal{G}}$ not present in $\mathcal{G}^{(k)}$ (we discuss this further below). If all constraints are satisfied, then we are done. Then, the solution $\hat{\eta}^{(k)}$ solves the full MER problem over $\tilde{\mathcal{G}}$.

3. If there are any constraint violations, then pick the m *most violated* edge constraints, that is, those edges $E \in \tilde{\mathcal{G}} \setminus \mathcal{G}^{(k)}$ where $d_H(\hat{\eta}^{(k)}_{[E]}, \tilde{\eta}_{[E]}) - \delta_E$ is largest and greater than zero. Define $\mathcal{G}^{(k+1)}$ to be the union of $\mathcal{G}^{(k)}$ with these m edges.

4. Check if $\mathcal{G}^{(k+1)}$ is tractable. This is performed using some heuristic method for finding efficient junction trees of a graph and checking if the resulting junction tree is thin. If $\mathcal{G}^{(k+1)}$ is thin, then set $k \leftarrow k+1$ and go back to Step 1. Otherwise, terminate the procedure (this suggests that the MER solution is probably not thin, and therefore intractable to obtain using this method).

There are two aspect of this procedure that need to be clarified. First, even assuming that each of the graphs $\mathcal{G}^{(k)}$ is thin, we still need to explain how we intend to solve the reduced MER problem over each thin graph. We do this using the algorihtm developed in the following section. The basic idea is that, if $\mathcal{G}^{(k)}$ is thin, then we can actually solve MER on a thin, chordal supergraph of $\mathcal{G}^{(k)}$, for which it is tractable to compute the entropy.

Evaluating the MER Constraints

The second question we need to address is how to tractably extend the moments $\hat{\eta}^{(k)}$, defined over $\hat{\mathcal{G}}^{(k)}$, to the full set of moments $\hat{\eta}$ defined over $\tilde{\mathcal{G}}$, which is necessary so that we can evaluate the MER constraints. This requires computing the marginal distribution on each edge $E \in \tilde{\mathcal{G}}$ given the sparse graphical model specified by $\hat{\theta}_k = \Lambda^{-1}(\hat{\eta}^{(k)})$ defined on $\mathcal{G}^{(k)}$. If we restrict $\tilde{\mathcal{G}}$ only to contain edges up to size s, then this step can in principle be accomplished by a tractable procedure. For each subset A of $s-1$ nodes, we build an augmented version of the graph $\mathcal{G}^{(k)}$ in which edges are added so that each node of the graph is linked to every node of A. This augmented graph is still thin (the treewidth is increased by at most $s-1$), so that we can perform inference over this graph to compute the marginal distributions of edges $E = A \cup \{v\}$ for all $v \in V$. Repeating this procedure for all $\binom{n}{s-1}$ subsets, we thereby obtain all marginal distributions on edges E comprised of s or fewer nodes. The number of subsets A, for which we must perform this inference calculation, is bounded by $\binom{n}{s-1} \leq \frac{n^{s-1}}{(s-1)!}$, with each inference calculation requiring $\mathcal{O}(n|\mathbb{X}|^{w+s})$ calculations. The total complexity is then $\mathcal{O}(n^s|\mathbb{X}|^{w+s-1})$ where w is the treewidth of $\mathcal{G}^{(k)}$. Once we have all the moments $\hat{\eta}$ over $\tilde{\mathcal{G}}$, actually evaluating the constraints has complexity $\mathcal{O}((n|\mathbb{X}|)^s)$. Thus, the complexity of the overall procedure is essentially $\mathcal{O}(n^s|\mathbb{X}|^{w+s-1})$ where w is the treewidth of the final MER solution.

Tractability of MER

Thus, if we keep the maximum edge size s small, it is tractable to solve the MER problem as long as the MER solution results in a thin graphical model.[5] Of course,

[5]We are being a bit optimistic, as we assume that the width of the graphs $\mathcal{G}^{(k)}$ does not substantially exceed that of $\hat{\mathcal{G}}$. Conceivably, this assumption could be false, if we were to find that many of the "most violated" edges in the bootstrap method become inactive in the final MER solution. However, in all of our experiments performed thus far, this has not been the case. As long as the number of edges m added at each step is not made too large, the final graph $\mathcal{G}^{(k)}$ has been almost identical to $\hat{\mathcal{G}}$, with only a few extra inactive edges in the final solution.

if the solution $\hat{\mathcal{G}}$ is not thin, then the graph $\hat{\mathcal{G}}^{(k)}$ becomes intractable and we cannot tractably solve the MER problem. In that case, there are two possible options one might consider. First, by increasing the level of relaxation of the MER problem one can increase the sparsity of the solution to eventually obtain a thin model. However, this approach will most likely lead to a poor model of the probability distribution that we are trying to learn or approximate. Another approach would be instead to consider *approximate* versions of MER in which the intractable entropy function $H(\eta)$ is replaced by a tractable proxy for entropy. However, we focus on solving the exact MER problem in this thesis. Some possible extensions of MER, using approximate inference to learn non-thin graphical models, are discussed in the conclusions (Chapter 6).

■ 5.3.2 Solving MER on Thin Chordal Graphs

In this section we specify an algorithm for solving MER on thin chordal graphs. The methods that we develop rely on the fact that the graph is chordal in an essential way. However, by Proposition 5.3.1, we may obtain the solution of the MER problem defined on any thin (non-chordal) graph $\mathcal{G}$ by reformulating the problem on a thin chordal supergraph of $\mathcal{G}$. Hence, the methods developed in this section can be used to solve MER on any thin graph. Moreover, using the boot-strap method of the previous section, this also allows us to solve MER on essentially arbitrary (non-thin) graphs, provided the solution is still a thin graphical model.

Tractable Computations on Chordal Graphs

The main fact we use to solve MER on thin chordal graphs is that the entropy $H(\eta)$ is tractable to compute using the junction-tree decomposition (derived in Section 2.7.1):

$$H(\eta) = \sum_C H_C(\eta_{[C]}) - \sum_S H_S(\eta_{[S]}) \tag{5.14}$$

To evaluate $H(\eta)$ using this formula we need only compute local marginal entropies on the maximal cliques and corresponding separators in the junction tree. The total complexity of this calculation is $\mathcal{O}(n|\mathbb{X}|^w)$ in discrete graphical models and $\mathcal{O}(nw^3)$ in the Gaussian model, where w is the size of the largest clique. This also implies it is tractable to compute the gradient of entropy as:

$$\nabla H(\eta) = \sum_C [\nabla H_C(\eta_{[C]})]_{\mathcal{I}} - \sum_S [\nabla H_S(\eta_{[S]})]_{\mathcal{I}} \tag{5.15}$$

We use $\mathcal{I}$ to denote the index set of the model features $\phi(x) = (\phi_\alpha(x_\alpha), \alpha \in \mathcal{I})$ and let $[\theta_{[C]}]_{\mathcal{I}}$ denotes zero-padding of a vector $\theta_{[C]} \in \mathbb{R}^{[C]}$, where $[C] \subset \mathcal{I}$, to a vector in $\mathbb{R}^{\mathcal{I}}$. Recalling that $-\nabla H(\eta) = \Lambda^{-1}(\eta)$, it is also tractable to compute this mapping from η to θ by the formula:

$$\Lambda^{-1}(\eta) = \sum_C [\Lambda_C^{-1}(\eta_{[C]})]_{\mathcal{I}} - \sum_S [\Lambda_S^{-1}(\eta_{[S]})]_{\mathcal{I}} \tag{5.16}$$

This is the exact formula for computing the projection to a chordal graph discussed in Section 2.7.1. Of course, the forward map $\Lambda(\theta) = \eta$ corresponds to inference in a thin graph, which is also tractable using the two-pass inference algorithm on the junction tree, as discussed in Sections 2.4.1 and 2.6.2. These are the essential tools one needs to solve MER on a thin chordal graph.

Fisher Information Calculations

In order to solve MER using second-order optimization methods, that is, methods such as Newton's method that exploit curvature information, we also need to be able to handle computations involving the Fisher information matrix $G(\eta)$ or its inverse. Recall, from Section 2.3.3, that $G(\eta) = -\nabla^2 H(\eta) = \partial\Lambda^{-1}(\eta)$. That is, the Fisher information matrix in η is equivalent to the negative Hessian of the entropy function $H(\eta)$, or the Jacobian of the mapping Λ^{-1} from η to θ. Also, the inverse of this Fisher information matrix is equivalent to the Fisher information in θ, that is, $G_\eta^{-1}(\eta) = G_\theta(\theta) = \nabla^2\Phi(\theta) = \partial\Lambda(\theta)$ where $\theta = \Lambda^{-1}(\eta)$ (here, G_η and G_θ denote Fisher information in either the η or θ parameterizations).

Taking derivatives with respect to (5.16), we obtain the following junction tree decomposition of the Fisher information in η:

$$G(\eta) = \sum_C [G_C(\eta_{[C]})]_{\mathcal{I}\times\mathcal{I}} - \sum_S [G_S(\eta_{[S]})]_{\mathcal{I}\times\mathcal{I}} \tag{5.17}$$

Here, $G_E(\eta_{[E]}) \triangleq -\nabla^2 H_C(\eta_{[E]}) = \partial\Lambda_E^{-1}(\eta_{[E]})$ denotes the marginal Fisher information computed on edge E and we write $[G_E]_{\mathcal{I}\times\mathcal{I}}$ to denote zero-padding of the matrix $G_E \in \mathbb{R}^{[E]\times[E]}$ to a matrix in $\mathbb{R}^{\mathcal{I}\times\mathcal{I}}$. This decomposition shows that $G(\eta)$ is a *sparse* matrix, reflecting the underlying Markov structure of the chordal graphical model. This allows one to implement second-order optimization methods for entropy maximization problems tractably in thin graphs. Such methods general require both multiplication by $G(\eta)$ and by its inverse, or, equivalently, solving a sparse linear system of the form $G(\eta)d\eta = d\theta$ for $d\eta$ given $d\theta$. Because the fill-pattern of the matrix $G(\eta)$ is based on a junction tree of the chordal graph $\mathcal{G}$, one can see that it defines another chordal graph with essentially the same junction tree structure as $\mathcal{G}$. However, this latter graph has more nodes (one for each feature $\phi_\alpha(x)$) and a larger treewidth, either $w' \approx |\mathbb{X}|^w$ in discrete models or $w' \approx w^2$ in Gaussian models, because each clique C of $\mathcal{G}$ maps to clique with $|[C]|$ nodes in the adjacency graph of $\mathcal{G}(\eta)$. Hence, solving a linear system based on $\mathcal{G}(\eta)$ (using direct methods analogous to Gaussian inference on a junction tree) results in complexity $\mathcal{O}(n|\mathbb{X}|^{3w})$ in discrete models or $\mathcal{O}(nw^6)$ in the Gaussian models.

While these direct computations using the sparse matrix $G(\eta)$ are linear in n, the dependence on the treewidth w of the graph is less efficient than exact evaluation of either $\Lambda(\theta)$ or $\Lambda^{-1}(\eta)$. Recall that $G(\eta) = \partial\Lambda^{-1}(\eta)$ and $G^{-1}(\eta) = \partial\Lambda(\theta)$. This suggests a more efficient method to implement the *operations* of multiplication of a vector by either $G(\eta)$ or $G^{-1}(\eta)$. For example, to compute $d\theta = G(\eta)d\eta$, we linearize

the projection formula (5.16) for computing $\Lambda^{-1}(\eta)$, to obtain:

$$d\theta = \sum_C [G_C(\eta_{[C]})d\eta_{[C]}]_{\mathcal{I}} - \sum_S [G_S(\eta_{[S]})d\eta_{[S]}]_{\mathcal{I}} \tag{5.18}$$

The complexity of the calculation $G_E(\eta_{[E]})d\eta_{[E]} = \partial\Lambda_E^{-1}(\eta_{[E]})d\eta_{[E]}$ on edge E is essentially the same as for computing Λ_E^{-1}, e.g., requiring $\mathcal{O}(|\mathbb{X}|^{|E|})$ calculations in discrete models or $\mathcal{O}(|E|^3)$ calculations in the Gaussian model. Thus, the operation of multiplication by $G(\theta)$ has overall complexity $\mathcal{O}(n|\mathbb{X}|^w)$ in discrete models and $\mathcal{O}(nw^3)$ in the Gaussian model. We give further details on the computation of Fisher information matrices in the Boltzmann machine and Gaussian models in Appendices C and D.

In a similar manner, one may linearize each message-passing step of the two-pass inference algorithm for computing $\eta = \Lambda(\theta)$ over a junction tree of the graph. This then gives an efficient method to compute $d\eta = G^{-1}(\eta)d\theta$. These methods allow one to develop second-order optimization methods for maximum-entropy problems which have essentially the same complexity (per iteration) as exact inference, thereby providing efficient optimization methods on the class of thin chordal graphs. We provide complete details of such a calculation in the Gaussian model in Appendix D.

Primal-Dual Interior Point Algorithm

We now summarize the primal-dual interior point method that we actually use to solve the MER problem. The primal-dual interior point method [37] is the preferred method for convex optimization problems with inequality constraints. We summarize a basic version of the algorithm presented in [37]. The basic idea is to use Newton's method to solve for the saddle point of the MER Lagrangian function, obtained by introducing Lagrange multipliers λ_E to relax the approximate marginal-matching constraints over the edges of $\mathcal{G}$. More precisely, one solves the modified Karush-Kuhn-Tucker (KKT) conditions associated with this saddle point, but with the modified complementary-slackness condition $\sum_{E\in\mathcal{G}} \lambda_E(d_H(\eta_{[E]}, \tilde{\eta}_{[E]}) - \delta_E) = \epsilon_k$, where ϵ_k is made smaller (approaching zero) as the procedure progresses. In MER, this generates a sequence of interior points of the MER feasible set that converge to the MER solution $\hat{\eta}$ (typically on the boundary of the feasible set) as ϵ approaches zero. Here, we only summarize the main computational steps of the procedure and refer the reader to [37] for further details and a complete derivation of the algorithm. An important point, however, is that in our implementation of the algorithm we take care to exploit the problem structure, using the efficient Fisher information calculations on chordal graphs described in the previous section.

The method computes both the moment parameters η in the MER problem defined over $\mathcal{G}$ and a set of Lagrange multipliers λ associated with the MER approximate marginal-matching constraints. Initialized by $\eta^{(0)} = \tilde{\eta}$ and $\lambda_E^{(0)} = 1$ for all $E \in \mathcal{G}$, the method generates a sequence of points $(\eta^{(k)}, \lambda^{(k)})$ and parameters ϵ_k, for $k = 1, 2, \ldots$, as follows. Given the previous solution (η, λ), we must first solve the following linear

system of equations for $\Delta\eta$:

$$G_{\text{pd}}\Delta\eta = r_{\text{pd}} \tag{5.19}$$

where:

$$\begin{aligned} G_{\text{pd}} &\triangleq G(\eta) + \sum_{E\in\mathcal{G}} \lambda_E \left[G_E(\eta_{[E]}) + \frac{\nabla d_H(\eta_{[E]}, \tilde{\eta}_{[E]}) \nabla d_H(\eta_{[E]}, \tilde{\eta}_{[E]})^T}{\delta_E - d_H(\eta_{[E]}, \tilde{\eta}_{[E]})} \right]_{\mathcal{I}\times\mathcal{I}} \\ r_{\text{pd}} &\triangleq \Lambda^{-1}(\eta) + \epsilon^{-1} \sum_{E\in\mathcal{G}} \frac{\lambda_E}{\delta_E - d_H(\eta_{[E]}, \tilde{\eta}_{[E]})} [\nabla d_H(\eta_{[E]}, \tilde{\eta}_{[E]})]_{\mathcal{I}} \end{aligned}$$

and $\nabla d_H(\eta_{[E]}, \tilde{\eta}_{[E]}) = \Lambda_E^{-1}(\eta) - \Lambda_E^{-1}(\tilde{\eta}_{[E]})$. We consider two methods to compute the solution of 5.19. First, we note that the fill-pattern of G_{pd} is the same as of $G(\eta)$ itself. Each constraint on edge E contributes a term to the submatrix indexed by $[E]$, which is already covered by some clique of the graph. Hence, using direct methods we can exactly solve this system of equations with complexity that is cubic in the treewidth w' of the adjacency graph of $\mathcal{G}(\eta)$. However, as we have discussed, this w' is equal to the number of features defined within the largest clique of $\mathcal{G}$, which is larger than the treewidth w of $\mathcal{G}$ (e.g., in the discrete model $w' \approx |\mathbb{X}|^w$). Alternatively, it can be more effective to use the efficient algorithm for multiplication by either $G(\eta)$ and $G^{-1}(\eta)$ on thin chordal graphs to implement an iterative method, such as the conjugate gradient method [96], using $G^{-1}(\eta)$ as a preconditioner for the iterative method. If the constraints in the MER problem are not made too tight, which leads to smaller values of the Lagrange multipliers λ, then $G^{-1}(\eta)$ provides a good preconditioner and the iterative method converges very quickly. Because each iteration of this iterative method for solving (5.19) only requires $\mathcal{O}(|\mathbb{X}|^w)$ calculations (in the discrete model), this can be much more efficient than using the direct solution method that requires $\mathcal{O}(|X|^{3w})$ computation.

Once we have solved the linear system (5.19) for $\Delta\eta$, we also compute

$$\Delta\lambda_E = -\lambda_E + (\delta_E - d_H(\eta_{[E]}, \tilde{\eta}_{[E]}))^{-1} \left(\lambda_E \nabla d_H(\eta_{[E]}, \tilde{\eta}_{[E]})^T \Delta\eta_{[E]} - \epsilon^{-1} \right). \tag{5.20}$$

The new values of (η, λ) are then determined by:

$$\eta^{(k+1)} = \eta^{(k)} + s\Delta\eta \tag{5.21}$$

$$\lambda^{(k+1)} = \lambda^{(k)} + s\Delta\lambda \tag{5.22}$$

where $s \in (0, 1)$ is a step-size parameter that is set by an inexact "back-tracking" line search procedure. This line search procedure ensures that we stay within the MER feasible set and that the residual error in the modified KKT conditions is reduced at each step (see [37] for details). Lastly, the parameter ϵ is set to the new value

$$\epsilon_{k+1} = \frac{\alpha}{|\mathcal{G}|} \sum_{E\in\mathcal{G}} \lambda_E^{(k+1)} (\delta_E - d_H(\eta_{[E]}^{(k+1)}, \tilde{\eta}_{[E]})) \tag{5.23}$$

where $\alpha \in (0,1)$ is a parameter of the primal-dual algorithm (typically set to $\alpha \approx 0.1$). This procedure is continued until we obtain a good solution to the modified KKT conditions for a sufficiently small value of ϵ (see [37] for the precise stopping conditions).

This provides a super-linearly convergent algorithm for solution of the MER problem defined on chordal graphs, one in which all necessary calculations have the same complexity as recursive inference over the chordal graph.

■ 5.4 MER Dual Interpretation and Methods

In this section we consider a dual form of MER that is explicitly formulated in terms of the potentials of the graphical model. We see that this dual form corresponds to a regularized version of maximum-likelihood parameter estimation, where an additional information regularization term is included in the objective that favors sparse graphical models.

■ 5.4.1 Dual Decomposition of MER

To derive this dual formulation of MER, we use a Lagrangian decomposition method that allows us to decouple the constraints so as to obtain a closed-form expression for the Lagrangian dual function [37].

The MER Lagrangian

This involves first introducing auxiliary moment parameters $\eta^{[E]} \in \mathbb{R}^{[E]}$ for each edge $E \in \mathcal{G}$. Unlike the expression $\eta_{[E]}$, which refers to a *subset* of the moment parameters $\eta \in \mathbb{R}^{\mathcal{I}}$, each moment vector $\eta^{[E]}$ should be regarded as an *independent* set of parameters defined for each edge E. We use this superscript convention throughout the section. Also, we denote the complete set of all of these auxiliary moment parameters as $\eta^\dagger = (\eta^{[E]}, E \in \mathcal{G})$. We substitute these parameters $\eta^{[E]}$ for $\eta_{[E]}$ within each edge constraint of MER (this serves to decouple the constraints when we take the dual). However, to enforce consistency among these redundant moment parameters, we also include equality constraints $\eta^{[E]} = \eta_{[E]}$ for each edge. This gives us the following equivalent formulation of MER in terms of the redundant problem variables $(\eta, \eta^\dagger)$:

$$\begin{array}{lll} & \text{maximize} & H(\eta) \\ (\text{MER}^\dagger) & \text{subject to} & d_H(\eta^{[E]}, \tilde{\eta}_{[E]}) \leq \delta_E \text{ and} \\ & & \eta^{[E]} = \eta_{[E]} \text{ for all } E \in \tilde{\mathcal{G}} \end{array} \tag{5.24}$$

Now we take the Lagrangian dual of this representation of MER. Introducing Lagrange multipliers λ_E for each constraint $d_H(\eta^{[E]}, \tilde{\eta}_{[E]}) \leq \delta_E$ and a vector of Lagrange multipliers $\theta^{[E]}$ associated with the constraints $\eta^{[E]} = \eta_{[E]}$, we obtain the Lagrangian function:

$$L(\eta, \eta^\dagger; \theta^\dagger, \lambda) = H(\eta) + \sum_{E \in \mathcal{G}} \left\{ \lambda_E(\delta_E - d_H(\eta^{[E]}, \tilde{\eta}_{[E]})) + (\theta^{[E]})^T(\eta_{[E]} - \eta^{[E]}) \right\} \tag{5.25}$$

Note, the Lagrange multipliers $\theta^{[E]}$ defined on different edges are independent of one another and we write $\theta^\dagger = (\theta^{[E]}, E \in \mathcal{G})$ to denote this complete set of Lagrange multipliers. We will see that these Lagrange multipliers $\theta^{[E]}$ serve as an over-parameterized potential representation of the MER solution.

Computing the MER Dual Function

Now, the Lagrangian dual function is defined by taking the maximum of the Lagrangian over $(\eta, \eta^\dagger)$ for each choice of the Lagrange multipliers $(\theta^\dagger, \lambda)$:

$$\begin{aligned}\mathcal{F}(\theta^\dagger, \lambda) &\triangleq \max_{\eta,\eta^\dagger} L(\eta, \eta^\dagger; \theta^\dagger, \lambda) \\ &= \max_{\eta} \left\{ H(\eta) + \left(\sum_E [\theta^{[E]}]_{\mathcal{I}} \right)^T \eta \right\} \\ &\quad + \sum_E \left(\lambda_E \delta_E + \max_{\eta^{[E]}} \left\{ -\lambda_E \, d_H(\eta^{[E]}, \tilde{\eta}_{[E]}) - (\theta^{[E]})^T \eta^{[E]} \right\} \right) \end{aligned} \tag{5.26}$$

Introducing the auxiliary parameters $\eta^{[E]}$ on each edge has allowed the maximization to be decomposed into separate maximizations with respect to η and each $\eta^{[E]}$. Each separate maximization can now be evaluated using the conjugate duality relation between entropy and the log-partition function. First, the maximum over η evaluates to

$$\max_{\eta} \{ H(\eta) + \eta^T \theta \} = \Phi(\theta)$$

with $\theta \triangleq \sum_E [\theta^{[E]}]_{\mathcal{I}}$. Thus, we see that the Lagrange multipliers $\theta^\dagger$ corresponds to an over-parameterized decomposition of the overall potential vector θ into edge-wise potential vectors $\theta^{[E]}$.

Each maximum over $\eta^{[E]}$ can be computed similarly. Let us expand the marginal relative entropy as

$$d_H(\eta^{[E]}, \tilde{\eta}_{[E]}) \equiv d(\eta^{[E]}, \tilde{\theta}^{[E]}) = -H_E(\eta^{[E]}) + \Phi_E(\tilde{\theta}^{[E]}) - (\eta^{[E]})^T \tilde{\theta}^{[E]}$$

where $\tilde{\theta}^{[E]} \triangleq \Lambda_E^{-1}(\tilde{\eta}_{[E]})$. Then, the maximum over $\eta^{[E]}$ in (5.26) becomes:

$$\max_{\eta^{[E]}} \left\{ \lambda_E H_E(\eta^{[E]}) + (\eta^{[E]})^T (\lambda_E \tilde{\theta}^{[E]} - \theta^{[E]}) \right\} - \lambda_E \Phi_E(\tilde{\theta}^{[E]})$$

If $\lambda_E = 0$, then the value of the maximum is $+\infty$ if $\theta^{[E]} \neq 0$ or is zero if $\theta^{[E]} = 0$. If $\lambda_E > 0$, then it is equal to

$$\lambda_E \left(\max_{\eta^{[E]}} \left\{ H_E(\eta^{[E]}) + (\eta^{[E]})^T (\tilde{\theta}^{[E]} - \lambda_E^{-1} \theta^{[E]}) \right\} - \Phi_E(\tilde{\theta}^{[E]}) \right) = \lambda_E \left(\Phi_E(\tilde{\theta}^{[E]} - \lambda_E^{-1} \theta^{[E]}) - \Phi_E(\tilde{\theta}^{[E]}) \right)$$

Let $\hat{\mathcal{G}}(\lambda) \subset \mathcal{G}$ denote the set of edges where $\lambda_E > 0$.

Putting this all together, we have that the dual function is given by

$$\mathcal{F}(\theta^\dagger, \lambda) = \Phi(\theta) + \sum_{E \in \hat{\mathcal{G}}(\lambda)} \lambda_E \left[\Phi_E(\tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]}) - \Phi_E(\tilde{\theta}^{[E]}) + \delta_E \right] \tag{5.27}$$

if $\theta^{[E]} = 0$ for all $E \notin \hat{\mathcal{G}}(\lambda)$, and $\mathcal{F}(\theta^\dagger, \lambda) = +\infty$ if $\theta^{[E]}$ is non-zero for any edge $E \notin \hat{\mathcal{G}}(\lambda)$.

The MER Dual Problem

The dual function (5.27) provides an upper-bound on the value of the MER problem for all $\theta^\dagger$ and $\lambda \geq 0$. Thus, the dual problem is then to minimize the dual function to obtain the tightest possible upper-bound:

$$\text{(MER-D)} \qquad \begin{array}{ll} \text{minimize} & \mathcal{F}(\theta^\dagger, \lambda) \\ \text{subject to} & \lambda \geq 0 \end{array} \tag{5.28}$$

Note, the optimization is over both $\theta^\dagger$ and $\lambda \geq 0$. This is a convex optimization problem, minimizing the convex function $\mathcal{F}(\theta^\dagger, \lambda)$ over a convex set, and its solution is essentially equivalent to solving MER:

Proposition 5.4.1 (MER Strong Duality). *If $\tilde{\eta} \in \mathcal{M}$ and $\delta > 0$, then the value (MER-D) is equal to that of (MER) and the solution of these two problems are related by $\hat{\eta} = \Lambda(\hat{\theta})$ where $\hat{\theta} = \sum_{E \in \mathcal{G}} [\hat{\theta}^{[E]}]_{\mathcal{I}}$.*

Strong duality holds because the MER problem is convex and strictly feasible (see [24] for proof that these are sufficient conditions for strong duality). Consider the reduced dual problem, defined as minimizing the following reduced dual function:

$$\begin{aligned} \mathcal{F}(\theta^\dagger) &\triangleq \min_{\lambda \geq 0} \mathcal{F}(\theta^\dagger, \lambda) && (5.29) \\ &= \Phi(\theta) + \sum_{E \in \tilde{\mathcal{G}}} \Psi_E(\theta^{[E]}) && (5.30) \end{aligned}$$

where

$$\Psi_E(\theta^{[E]}) \triangleq \min_{\lambda_E > 0} \left\{ \lambda_E \left[\Phi_E(\tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]}) - \Phi_E(\tilde{\theta}^{[E]}) + \delta_E \right] \right\} \tag{5.31}$$

for $\theta^{[E]} \neq 0$ and $\Psi_E(0) = 0$. For non-zero $\theta^{[E]}$, this function can be computed by performing a line-search with respect to $\lambda_E > 0$ to solve for the zero of the derivative of the quantity $\{\dots\} \triangleq h(\lambda_E)$ being minimized in (5.31). The derivative is

$$\begin{aligned} \frac{dh(\lambda_E)}{d\lambda_E} &= \left[\Phi_E(\tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]}) - \Phi_E(\tilde{\theta}^{[E]}) + \delta_E \right] - (\lambda_E^{-1}\theta^{[E]})^T \Lambda_E^{-1} \left(\tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]} \right) \\ &= \delta_E - d_\Phi(\tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]}, \tilde{\theta}^{[E]}) && (5.32) \end{aligned}$$

In the second line we have used the Bregman distance function based on Φ_E, which is equal to KL-divergence. The optimal value of λ_E can be determined as follows. We

search over the half-line $\{\theta^{[E]}(t) \triangleq \tilde{\theta}^{[E]} - t\theta^{[E]}, t > 0\}$, starting from $\tilde{\theta}^{[E]}$ and searching in the direction $-\theta^{[E]}$, to find the point $\theta^{[E]}(t)$ on this line where $d_\Phi(\theta^{[E]}(t), \tilde{\theta}_{[E]}) = \delta_E$ and then set $\lambda_E = t^{-1}$. Note that

$$f(t) \triangleq d_\Phi(\tilde{\theta}^{[E]} - t\theta^{[E]}, \tilde{\theta}^{[E]})$$

is zero at $t = 0$, monotonically increasing with t and diverges as t becomes large. Thus, the conditions for minimizing $\mathcal{F}(\theta^\dagger, \lambda)$ over λ_E are that either:

- $\lambda_E > 0$ and $d_\Phi(\tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]}, \tilde{\theta}_E) = \delta_E$ if $\theta^{[E]} \neq 0$, **OR**
- $\lambda_E = 0$ if $\theta^{[E]} = 0$.

For the optimal λ_E, we have from the condition $\frac{dh_E(\lambda_E)}{d\lambda_E} = 0$ that

$$\left[\Phi_E(\tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]}) - \Phi_E(\tilde{\theta}^{[E]}) + \delta_E\right] = (\lambda_E^{-1}\theta^{[E]})^T\Lambda_E^{-1}\left(\tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]}\right)$$

Substituting this into (5.31) gives

$$\Psi(\theta^{[E]}) = (\theta^{[E]})^T\Lambda_E^{-1}\left(\tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]}\right) \triangleq (\theta^{[E]})^T\hat{\eta}^{[E]}$$

where we have defined $\hat{\eta}^{[E]} \triangleq \Lambda_E^{-1}(\tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]})$ by the optimal choice of λ_E for $\theta^{[E]}$. This point $\hat{\eta}^{[E]}$ is on the boundary of the set of all $\eta^{[E]}$ such that $d_H(\eta^{[E]}, \tilde{\eta}_{[E]}) < \delta_E$ (the MER feasible set for $\eta^{[E]}$). Moreover, using information geometry, one can show that $\hat{\eta}^{[E]}$ is the furthest point of this set in the direction $-\theta^{[E]}$. That is,

$$\Psi_E(\theta^{[E]}) = \begin{cases} \text{maximum} & -(\theta^{[E]})^T\eta^{[E]} \\ \text{subject to} & d_H(\eta^{[E]}, \tilde{\eta}_{[E]}) \leq \delta_E \end{cases} \tag{5.33}$$

This shows a simple geometric interpretation of the function Ψ_E based on the corresponding feasible set for $\eta^{[E]}$ in MER. However, the line-search method described previously is still the more efficient method to actually evaluate Ψ_E.

Let us also consider the condition for minimizing $\mathcal{F}(\theta^\dagger, \lambda)$ with respect to $\theta^{[E]}$. If $\lambda_E > 0$, we find that:

$$\frac{\partial\mathcal{F}(\theta^\dagger, \lambda)}{\partial\theta^{[E]}} = \Lambda(\theta)_{[E]} - \Lambda_E(\tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]}) \tag{5.34}$$

If we identify $\eta \equiv \Lambda(\theta) = \Lambda(\sum_E[\theta^{[E]}])$ and $\eta^{[E]} \equiv \Lambda_E(\tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]})$, we see that optimality with respect to $\theta^{[E]}$ is equivalent to $\eta_{[E]} = \eta^{[E]}$. Also, the optimality conditions for λ_E may be restated as:

- $\lambda_E > 0$ and $d_H(\eta^{[E]}, \tilde{\eta}_{[E]}) = \delta_E$, **OR**
- $\lambda_E = 0$ and $d_H(\eta^{[E]}, \tilde{\eta}_{[E]}) < \delta_E$.

This shows the connection between optimality of MER-D and the constraints of MER being satisfied. Also, we see that this point $\eta = \Lambda^{-1}(\theta)$ satisfies the MER constraints and the minimum value of the dual problem is therefore equal to $H(\eta)$, which must then also be the maximum value of MER.

Robust Maximum-Likelihood

Note also that the global term, $\Phi(\theta)$, only depends on $\theta^\dagger$ through $\theta = \sum_E [\theta^{[E]}]_\mathcal{I}$. Again, it is really only this vector θ that we need to obtain, the precise manner in which this is split up to form the optimal choice of $\theta^\dagger$ is irrelevant. Thus, we could also view the MER dual problem as minimizing

$$\mathcal{F}(\theta) = \Phi(\theta) + \Psi(\theta) \tag{5.35}$$

with respect to θ where

$$\Psi(\theta) \triangleq \min_{\sum_E \theta^{[E]} = \theta} \sum_E \Psi_E(\theta^{[E]}) \tag{5.36}$$

This function $\Psi(\theta)$ captures the influence that the sample data, as summarized by $\tilde{\eta}$, has on the selection of θ. It is defined variationally as the minimum over all decompositions of θ into edge-wise terms of the sum of the edge-wise potentials $\Psi_E(\theta^{[E]})$. This global data-potential term can also be described geometrically with respect to the MER feasible set:

Lemma 5.4.1. *It holds that*

$$\Psi(\theta) = \left\{ \begin{array}{ll} \text{maximum} & -\theta^T \eta \\ \text{subject to} & \eta \in \mathcal{M}(\tilde{\eta}, \delta) \end{array} \right. \tag{5.37}$$

where $\mathcal{M}(\tilde{\eta}, \delta)$ represents the feasible set of MER.

In other words, $\Psi(\theta)$ is the maximum value of $-\theta^T \eta$ over all η vectors which are feasible in the MER problem. This then shows that minimizing $\mathcal{F}(\theta)$ is equivalent to the following "robust" version of maximum-likelihood parameter estimation:

$$\min_\theta \max_{\eta \in \mathcal{M}(\tilde{\eta}, \delta)} \left\{ \Phi(\theta) - \eta^T \theta \right\} \tag{5.38}$$

This is similar to the variational interpretation of maximum-likelihood parameter estimation, which minimizes $\mathcal{F}_0(\theta) \triangleq \Phi(\theta) - \tilde{\eta}^T \theta$ with respect to θ. In the robust method, one instead maximizes the "worst-case" likelihood over the uncertainty set $\eta \in \mathcal{M}(\tilde{\eta}, \delta)$ as specified in MER. This interpretation of a general class of maximum-entropy problems is described by [9], although their work specifically considers the ℓ_1-regularization method (ℓ_∞-relaxation in the maximum-entropy problem) and does not consider the MER formulation.

As described in Section 5.2.3, we may choose the δ parameters to ensure that the moments η of the distribution P that generated the sample data are contained by the MER feasible set with high probability. This then allows us to interpret MER-D as minimizing a high-probability upper-bound on $D(P, \hat{P})$, the divergence of our estimate $\hat{P}$ away from P. Such bounds on generalization error have been recently considered for other relaxations of the maximum entropy method [9, 72], such as using the ℓ_∞ constraints $\|\eta - \tilde{\eta}\|_\infty \leq \delta$ in the maximum entropy method, which is equivalent to

ℓ_1-regularized maximum-likelihood in the dual problem. One appealing feature of our information-theoretic approach to constraint relaxation is that it arises naturally from the large-deviations point of view. Roughly speaking, Sanov's theorem asserts that the MER constraint set $\{\eta^{[E]} | d_H(\eta^{[E]}, \tilde{\eta}_{[E]}) \leq \delta_E\}$ provides a good approximation to the "high-probability" level-sets $\{\eta^{[E]} | P(\tilde{\eta}^{[E]} | \eta_{[E]}) \leq \epsilon\}$ for small values of ϵ. This is one justification of our formulation.

Sparsity-Enforcing Regularized Maximum-Likelihood

We present another interpretation of MER-D that helps to explain why if favors sparse solutions in θ. The dual function can be rewritten as:

$$\mathcal{F}(\theta^\dagger, \lambda) = \mathcal{F}_0(\theta) + \sum_E \lambda_E \left[d_\Phi(\tilde{\theta}^{[E]}, \tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]}) - \delta_E \right] \tag{5.39}$$

The first term $\mathcal{F}_0(\theta) \triangleq \Phi(\theta) - \tilde{\eta}^T\theta$ is the usual free energy one minimizes in maximum-likelihood. Thus, we see $\mathcal{F}(\theta^\dagger, \lambda)$ includes additional penalty terms that are minimal at $\theta^{[E]} = 0$, so as to favor zero-potentials.

Performing the minimization over $\lambda \geq 0$, we obtain:

$$\mathcal{F}(\theta^\dagger) = \mathcal{F}_0(\theta) + \sum_E \Psi'_E(\theta^{[E]}) \tag{5.40}$$

where

$$\Psi'_E(\theta^{[E]}) \triangleq \max_{\lambda_E \geq 0} \left\{ \lambda_E \left[d_\Phi(\tilde{\eta}_{[E]}, \tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]}) - \delta_E \right] \right\} \tag{5.41}$$

Analogous to the geometric interpretation of Ψ_E, one can show that:

$$\Psi'_E(\theta^{[E]}) = \left\{ \begin{array}{ll} \text{maximum} & (\theta^{[E]})^T(\eta^{[E]} - \tilde{\eta}^{[E]}) \\ \text{subject to} & d_H(\eta^{[E]}, \tilde{\eta}_{[E]}) \leq \delta_E \end{array} \right. \tag{5.42}$$

Thus, Ψ'_E is a "shifted" version of Ψ being centered on the point $\tilde{\eta}^{[E]}$. Because $\tilde{\eta}^{[E]}$ is the center point of the MER feasible set for $\eta_{[E]}$, it then holds that Ψ'_E is a cone-like function with the following properties:

- $\Psi'_E(\theta^{[E]}) \geq 0$ for all θ_E.
- $\Psi'_E(\theta^{[E]}) = 0$ if and only if $\theta^{[E]} = 0$.
- $\Psi'_E(\alpha\theta^{[E]}) = \alpha\Psi'_E(\theta^{[E]})$ for all $\alpha \geq 0$.

We see that these potentials Ψ'_E strongly favor zero potentials. Thus, the overall-potential $\Psi'(\theta^\dagger) \triangleq \sum_E \Psi'_E(\theta^{[E]}$ favors sparse representations $\theta^\dagger$, somewhat similar to using an ℓ_1 norm $\|\theta^\dagger\|_1$. However, the regularization function $\Psi'(\theta^\dagger)$ differs from using an ℓ_1 norm in two regards. First, it regularizes subsets of parameters jointly over each edge of the graph, rather then single features. Second, it is entirely information

geometric in definition, such that its value is actually invariant to reparameterization of the exponential family, which is not at all true of $\|\theta\|_1$.

Performing the minimization over reparameterizations $\theta^\dagger$ for a given θ, we obtain:

$$\mathcal{F}(\theta) = \mathcal{F}_0(\theta) + \Psi'(\theta) \tag{5.43}$$

where Ψ' may be equivalently defined as either

$$\Psi'(\theta) = \min_{\sum_E [\theta^{[E]}] = \theta} \sum_E \Psi'_E(\theta^{[E]}) \tag{5.44}$$

or geometrically as

$$\Psi(\theta) = \begin{cases} \text{maximum} & \theta^T(\eta - \tilde{\eta}) \\ \text{subject to} & \eta \in \mathcal{M}(\tilde{\eta}, \delta) \end{cases} \tag{5.45}$$

Thus, optimization over decompositions of θ within the over-parameterized representation $\theta^\dagger$ also plays an important in how MER-D regularizes the choice of θ, which is another difference from using the ℓ_1-regularization of θ.

■ 5.4.2 Relaxed Iterative Scaling Algorithm

In this section we derive a block coordinate-descent method for solution of the MER dual problem. We recall that the iterative scaling method, commonly used for maximum-entropy modeling, is equivalent to performing block coordinate-descent with respect to $\mathcal{F}_0(\theta) = \Phi(\theta) - \tilde{\eta}^T\theta$. Thus, our procedure is analogous to this iterative scaling method and results in a similar update procedure. However, this relaxed version of the iterative scaling algorithm requires an additional line-search step to determine the optimal block coordinate-descent step. The updated edge potential is then a "damped" version of the edge potential that would occur in the standard algorithm. Also, we find that the relaxed algorithm includes an explicit model thinning step, that results in sparse graphical models when the MER constraints become inactive.

Iterative Scaling Revisited

First, we rederive the iterative scaling algorithm using the notation of this chapter. This helps to clarify our derivation of the relaxed algorithm and also facilitates comparison of the two algorithms.

Let $\theta = \sum_E [\theta^{[E]}]_{\mathcal{I}}$ and consider performing block coordinate-descent to minimize $\mathcal{F}_0(\theta)$ with respect to $\theta^{[E]}$ for each $E \in \tilde{\mathcal{G}}$. Computing derivatives with respect to $\theta^{[E]}$, we find

$$\frac{\partial \mathcal{F}_0(\theta)}{\partial \theta^{[E]}} = \Lambda(\theta)_{[E]} - \tilde{\eta}_{[E]}$$

Thus, the condition for minimizing the convex function $\mathcal{F}_0$ with respect to $\theta^{[E]}$ is that $\Lambda(\theta)_E = \tilde{\eta}_{[E]}$, that is, that the model's marginal moments on edge E should agree with

those of the sample data. Applying the operator Λ_E^{-1} to both sides of this equation, we obtain:

$$\Lambda_E^{-1}(\Lambda(\theta)_{[E]}) \triangleq \Pi_E(\theta) = \tilde{\theta}^{[E]} \tag{5.46}$$

where $\tilde{\theta}^{[E]} \triangleq \Lambda_E^{-1}(\tilde{\eta}_{[E]})$. The key insight which enables us to explicitly solve this equation is that the composite operation $\Pi_E(\theta) \triangleq \Lambda_E^{-1}(\Lambda(\theta)_E)$ represents the variable elimination calculation:

$$\exp\{\Pi_E(\theta)^T \phi_{[E]}(x_E)\} \doteq \sum_{x_{V \setminus E}} \exp\{\theta^T \phi(x)\}$$

where "$\doteq$" represents equality up to an multiplicative constant independent of x. Thus, we can "factor out" the effect due to $\theta^{[E]}$:

$$\exp\{\Pi_E(\theta)^T \phi_{[E]}(x_E)\} \doteq \exp\{(\theta^{[E]})^T \phi_{[E]}(x)\} \sum_{x_{V \setminus E}} \exp\{(\theta^{\setminus E})^T \phi(x)\}$$

where $\theta^{\setminus E} = \sum_{E' \neq E} [\theta^{[E']}]_{\mathcal{I}}$. This is compactly expressed by the identity

$$\Pi_E(\theta^{\setminus E} + \theta^{[E]}) = \Pi_E(\theta^{\setminus E}) + \theta^{[E]}.$$

Using this identity, (5.46) now becomes $\theta^{[E]} + \Pi_E(\theta^{\setminus E}) = \tilde{\theta}^{[E]}$. Solving for $\theta^{[E]}$, we obtain:

$$\theta^{[E]} = \tilde{\theta}^{[E]} - \Pi_E(\theta^{\setminus E})$$

This then gives us our formula for performing block-coordinate descent with respect to $\mathcal{F}_0(\theta)$. To relate this to the usual form of the iterative scaling algorihtm, we use $\Pi_E(\theta^{\setminus E}) = \Pi_E(\theta) - \theta^{[E]}$ to obtain the formula:

$$\theta^{[E]} \leftarrow \theta^{[E]} + (\tilde{\theta}^{[E]} - \Pi_E(\theta))$$

We now recognize this as corresponding to the parametric representation of the usual form of iterative scaling algorithm. Adding the correction term $(\tilde{\theta}^{[E]} - \Pi_E(\theta))$ to $\theta^{[E]}$ corresponds to multiplying $P(x)$ by $\frac{\tilde{P}(x_E)}{P(x_E)}$ in the standard "iterative proportional fitting" form of the algorithm.

The Relaxed Algorithm

We can now apply a similar strategy to minimize $\mathcal{F}(\theta^\dagger, \lambda)$. We begin by taking the derivative of $\mathcal{F}(\theta^\dagger, \lambda)$ with respect to $\theta^{[E]}$:

$$\frac{\partial \mathcal{F}(\theta^\dagger, \lambda)}{\partial \theta^{[E]}} = \Lambda(\theta)_{[E]} - \Lambda_E(\tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]})$$

The condition for optimality of $\theta^{[E]}$ is that $\Lambda(\theta)_{[E]} = \Lambda_E(\tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]})$. Applying the operator Λ_E^{-1} to both sides of this equation, we obtain the optimality condition

$$\Pi_E(\theta) = \tilde{\theta}^{[E]} - \lambda_E^{-1}\theta^{[E]}.$$

Using the identity $\Pi_E(\theta) = \Pi_E(\theta^{\backslash E}) + \theta^{[E]}$ and solving for $\theta^{[E]}$ we obtain:

$$\theta^{[E]} = \frac{1}{1 + \lambda_E^{-1}}(\tilde{\theta}^E - \Pi_E(\theta^{\backslash E})) \tag{5.47}$$

This gives a closed-form solution for performing block coordinate-descent on $\mathcal{F}(\theta^\dagger, \lambda)$ with respect to $\theta^{[E]}$. We could then use the line search method to separately optimize λ_E. This entails performing a line search to find λ such that

$$d_\Phi(\tilde{\theta}^{[E]} - \lambda^{-1}\theta^{[E]}, \tilde{\theta}^{[E]}) = \delta_E \tag{5.48}$$

This equation has a unique solution if $\theta^{[E]} \neq 0$. If $\theta^{[E]} = 0$, then we set $\lambda_E = 0$.

Rather than performing alternating minimizations over $\theta^{[E]}$ and λ_E separately, we may instead optimize over both parameters at once. Replacing $\theta^{[E]}$ in (5.48) by the expression for its optimal values as a function of λ from (5.47), we obtain the line-search problem of finding $\rho \in [0, 1]$ to solve:

$$d_\Phi(\hat{\theta}^{[E]}(\rho), \tilde{\theta}^{[E]}) = \delta_E \tag{5.49}$$

where

$$\hat{\theta}^{[E]}(\rho) \triangleq \rho\,\tilde{\theta}^{[E]} + (1 - \rho)\Pi_E(\theta^{\backslash E}). \tag{5.50}$$

In this line-search problem, we have introduced the parameter ρ to denote the quantity $\frac{\lambda_E}{1+\lambda_E}$ based on the Lagrange multiplier λ_E. Note that ρ serves to interpolate between $\hat{\theta}^{[E]}(0) = \Pi_E(\theta^{\backslash E})$ and $\hat{\theta}^{[E]}(1) = \tilde{\theta}^{[E]}$. If $d_\Phi(\Pi(\theta^{\backslash E}), \tilde{\theta}^{[E]}) \geq \delta_E$, then there exists a unique $\rho \in [0, 1)$ satisfying this condition. We can then solve for ρ using simple line-search methods. If $d_\Phi(\Pi(\theta^{\backslash E}), \tilde{\theta}^{[E]}) < \delta_E$, then there is no solution and we set $\rho = 0$. In either case, the optimal coordinate-descent step is then:

$$\lambda_E = \frac{\rho}{1 - \rho} \tag{5.51}$$

$$\theta^{[E]} = \rho(\tilde{\theta}^E - \Pi_E(\theta^{\backslash E})) \tag{5.52}$$

For $\rho = 0$, both λ_E and $\theta^{[E]}$ are set to zero. Note that $\rho = 1$, in the update (5.52), corresponds to the standard iterative scaling solution for $\theta^{[E]}$. Thus, we recover the standard iterative scaling method in the limit as $\delta_E \to 0$. Hence, we view this as a damped version of the standard algorihtm, where ρ is the adaptively determined damping factor.

Projection Interpretation of Relaxed Iterative Scaling

The standard iterative scaling algorithm can be interpreted as an iterative projection algorithm with respect to the moment parameters η of the model. In this interpretation, each iterative-scaling step is viewed as projecting the previous solution onto the submanifold of models which satisfy an (exact) marginal-matching constraint on edge

E. That is, given the previous solution η', one defines the next η as the solution to the information projection problem:

$$\begin{array}{ll} \text{minimize} & d_H(\eta, \eta') \\ \text{subject to} & \eta_{[E]} = \tilde{\eta}_{[E]} \end{array} \tag{5.53}$$

The relaxed version of iterative scaling that we developed to solve MER may also be described from the point of view of using information projections to enforce marginal constraints. However, there are two differences. First, in MER we instead enforce the *approximate* marginal-matching constraints on each edge $E \in \tilde{\mathcal{G}}$. Second, there is an additional *model thinning* step in the relaxed algorithm that does not correspond to an information projection. To be more precise, we may view the potential update on edge E in the relaxed method as having two distinct steps:

1. *Model Thinning.* First, we set $\theta^{[E]} = 0$ in the potential decomposition $\theta^{\dagger}$, which may be viewed as thinning the graphical model by removing edge E. This gives a new model θ'.

2. *Information Projection.* Next, we project $\eta' = \Lambda(\theta')$ to the set of moment vectors that satisfy the MER constraint on edge E:

$$\begin{array}{ll} \text{minimize} & d_H(\eta, \eta') \\ \text{subject to} & d_H(\eta_{[E]}, \tilde{\eta}_{[E]}) \leq \delta_E \end{array} \tag{5.54}$$

   This is implementing by adjusting the potential $\theta^{[E]}$ so that the constraint is satisfied.

The line-search in the relaxed iterative scaling procedure serves to identify the closest point to η' which satisfies the approximate marginal-matching constraint on edge E. The relaxed update of $\theta^{[E]}$ then accomplishes the information projection step. Thus, the relaxed algorithm does not simply project onto the MER feasible set. Rather, it alternates between thinning the graphical model by setting an edge's potential to zero and enforcing the MER constraints on that edge by performing an information projection. Note that if the thinned model θ' already satisfies the MER constraint on edge E, then the optimal solution of the information projection step is just $\eta = \eta'$. In that case, $\theta^{[E]}$ is still equal to zero after the information projection step. Roughly speaking, the relaxed algorithm may be seen as trying to find the thinnest graphical model that satisfies the MER constraints.

■ 5.5 Experimental Demonstrations

In this section, we describe the results of simulations that demonstrate the effectiveness of the MER framework in learning the Markov structure of Boltzmann and Gaussian models from sample data. The tolerance parameters used in the MER problem are set

in proportion to the number of parameters needed to specify the marginal distribution as follows:

$$\delta_E = \gamma \times \begin{cases} |E| + \binom{|E|}{2}, & \text{Gaussian} \\ 2^{|E|} - 1, & \text{Boltzmann} \end{cases} \tag{5.55}$$

Here, $\gamma > 0$ is an overall regularization parameter which controls the trade-off between complexity and accuracy in the resulting MER solution. Our motivation for setting δ proportional to the number of parameters associated to an edge is that, for large sample size, the expectation of $d_H(\eta_{[E]}, \tilde{\eta}_{[E]})$, where η are the actual moments and $\tilde{\eta}$ are the sample moments, is approximately equal to the number of parameters $|[E]|$ divided by the number of samples M, which also suggests choosing $\gamma \sim 1/M$. In the following examples, we explore the effect of varying γ. In practice, cross-validation methods might prove useful to determine γ that approximately minimizes an empirical estimate of the generalization error.

■ 5.5.1 Boltzmann model

We generated $M = 1000$ samples of a 10-node Boltzmann model displayed in Figure 5.1. This model includes a pairwise potential $(x_u - \frac{1}{2})(x_v - \frac{1}{2})$ for each pair of vertices that are linked by an edge, which defines a model with unbiased nodes. The empirical moments $\tilde{\eta}$ from these samples were provided as input to MER, where we impose marginal constraints on all singleton, doublet and triplet sets. Figure 5.1 shows the MER solutions for several values of γ. Notice the correspondence between the tolerance level and the amount of model-thinning. For small values of γ, the MER solution approaches the maximum-likelihood model in the full exponential family, resulting in a completely connected graph. As γ is increased, this allows the MER solution to become thinner. Eventually, for γ large enough, the solution corresponds to a completely disconnected graph. In this experiment, we recover the correct graphical structure of the test model for $\gamma = .015625$.

■ 5.5.2 Gaussian model

We describe two sets of experiments using Gaussian graphical models. In both of these experiments, we generate $M = 400$ samples of the test model. In the first experiment, we generate samples from a 16-node cyclic Gaussian model with constant node weights $J_{ii} = -2\theta_i = 1.0$ and edge weights $J_{ij} = -\theta_{ij} = -0.1875$ between nodes that are one or two steps away on the circle. Analogous to the Boltzmann case, Figure 5.2 shows the MER solution for various values of γ. In this case, for $\gamma = 0.0625$, we recover the correct graphical structure of the test model.

Bootstrap Method

Next, we demonstrate the bootstrap method to learn a Gaussian graphical model defined on 10×10 grid-structured model with edge weight $J_{ij} = -\theta_{ij} = -0.24$ between

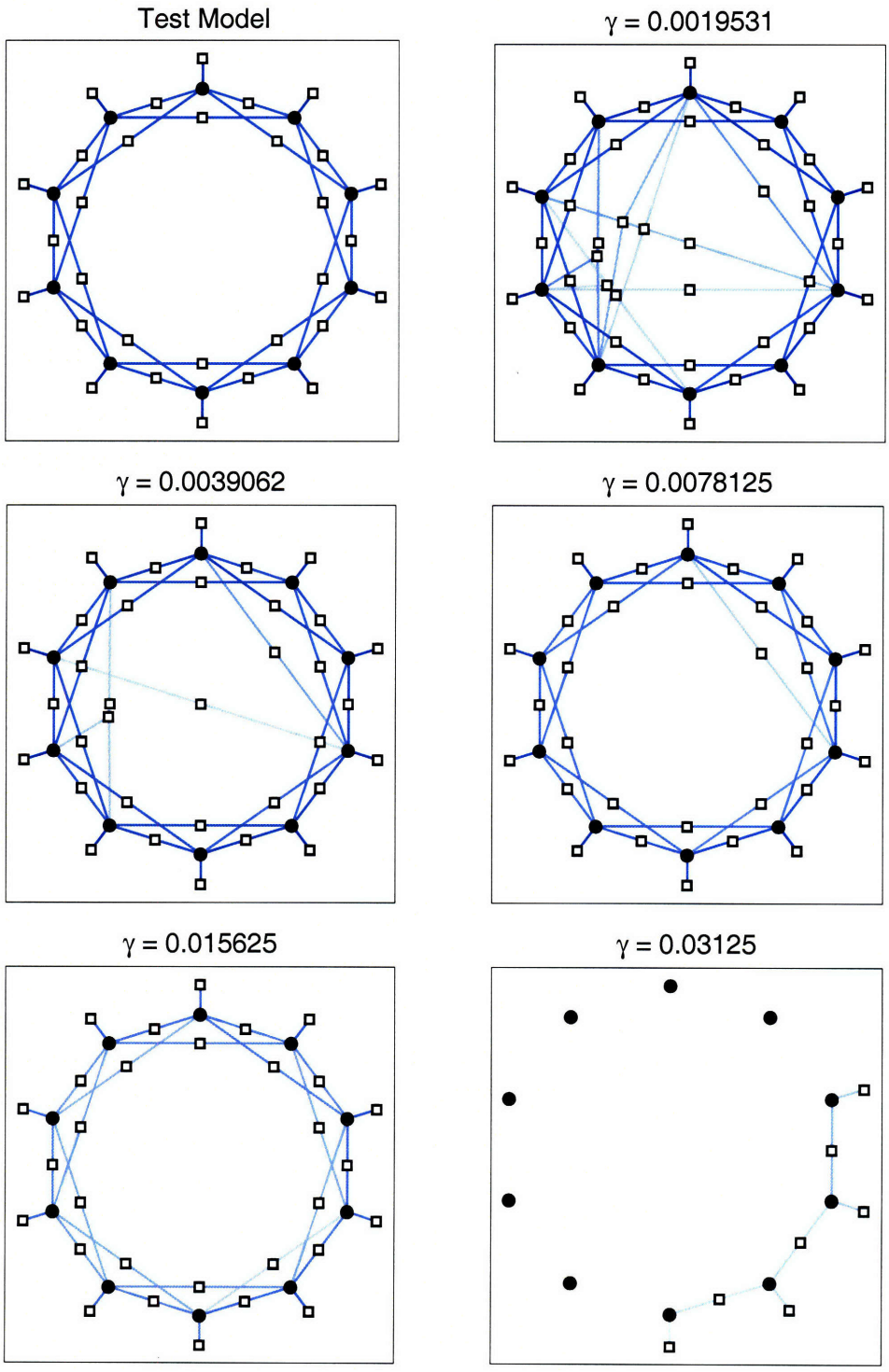

Figure 5.1. Graphs of the Boltzmann test model and MER solutions based on the sample moments for several values of γ.

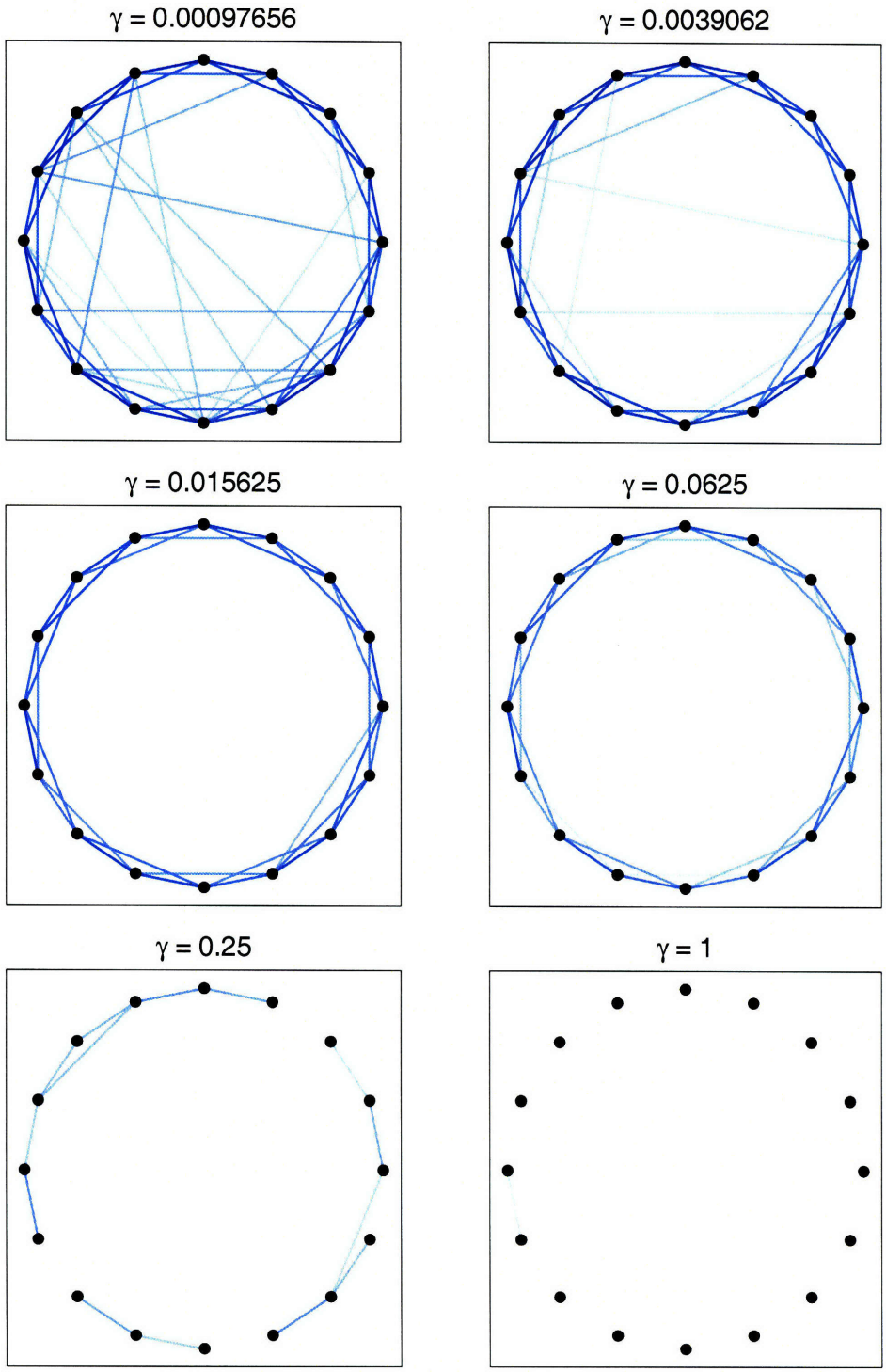

Figure 5.2. Graphs of the MER solution based on the sample moments for various values of γ in the Gaussian model.

nearest neighbors in the grid. Again, 400 samples were generated based on this model and the MER problem is solved for a fixed value of $\gamma = 0.08$. The initial MER problem is solved with 100 single-node constraints only, and at each successive step, the 50 most violated constraints are added. Figure 5.3 displays the resulting solutions obtained at each step of this bootstrap method.

In this case, directly solving the MER problem in the complete family (corresponding to the complete graph) would be computationally prohibitive because the Fisher information matrix in this complete model would have dimensions of roughly $10^4 \times 10^4$, which is difficult to store and to invert. However, our bootstrap method solves the MER problem on a sequence of thin graphs, obtaining the solution of the complete MER problem after only four steps of the bootstrap procedure. In this case, the final MER solution provides a good estimate of the correct graph structure, having just a few extra or missing edges.

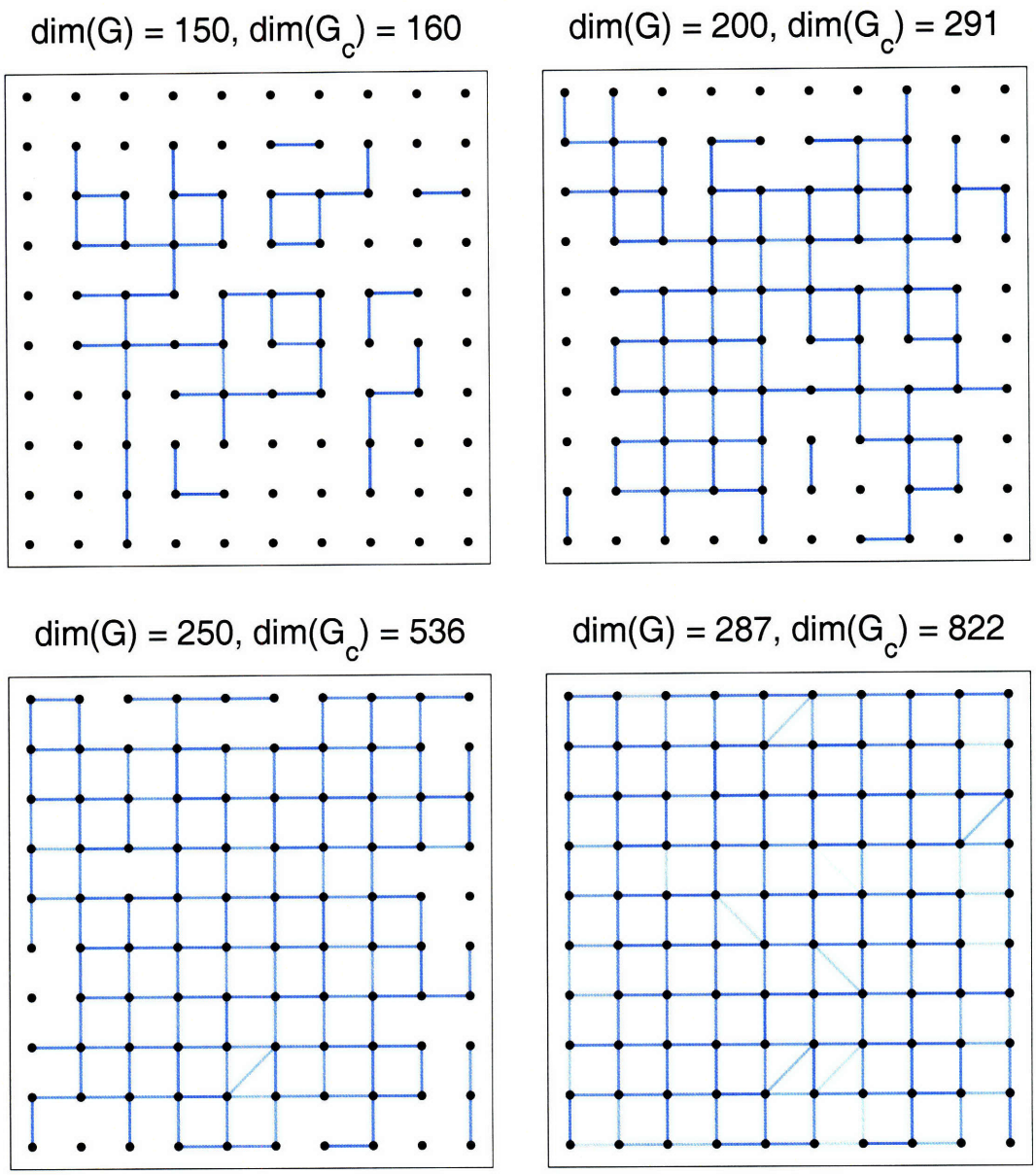

Figure 5.3. Illustration of the incremental approach for identifying the set of active constraints in the MER problem by solving a sequence of sub-problems defined on sub-graphs of the final solution (far right). Here, $dim(\mathcal{G})$ and $dim(\mathcal{G}_c)$ are the number of nodes plus the number of edges of the constraint graph $\mathcal{G}$ and its chordal super-graph $\mathcal{G}_c$ arising in the bootstrap method, which respectively determine the dimension of λ and η in each MER sub-problem.

Chapter 6

Conclusion

6.1 Summary

In this thesis, we have developed convex optimization methods to address two challenging problems commonly arising for graphical models:

1. Finding the most probable joint assignment (the MAP estimate) of all the random variables of a graphical model.

2. Learning a sparse graphical model to provide a good approximation to a more complex distribution, such as the sample distribution from training data.

Our approach to both of these problems is inspired by methods of convex optimization, information theory and statistical physics. In particular, maximum entropy and Lagrangian decomposition methods both play an important role in our approach to each of these problems.

6.1.1 Lagrangian Relaxation

To address the problem of MAP estimation, we have developed a Lagrangian relaxation method that decomposes a problem defined on an intractable graph into a set of decoupled problems defined on tractable subgraphs. We also have developed new methods for minimizing the resulting dual problem using iterative, distributed algorithms. The most interesting aspects of our formulation and methods are summarized below:

- An intractable graph is broken up into a set of tractable ones, either small connected subgraphs, embedded trees or other thin subgraphs. We then introduce Lagrange multipliers to enforce the constraint that, in the MAP estimate, duplicates of a variable should all be assigned consistently. These Lagrange multipliers then serve to parameterize how potentials of the intractable graphical model are split among the corresponding potentials in the graphical decomposition of this model. The Lagrangian dual function is then equal to the value of the MAP estimate in this decomposed model, which is tractably computed, and provides an upper-bound on the MAP value of the original problem for each possible potential decomposition. We then seek to minimize this dual function over all possible

decompositions of the potentials. If the minimal decomposition results in a consistent MAP estimate in the decomposed model, then there is no duality gap and we also recover the optimal MAP estimate of the original graphical model.

- To handle the fact that the Lagrangian dual function is non-differentiable, we propose a method inspired by statistical mechanics to obtain a smooth dual function using the log-sum-exp "soft-max" approximation to the max-function, defined by the free energy (log-partition function multiplied by temperature) of the Gibbs distribution based on the decomposed potential function. The temperature parameter of this Gibbs distribution serves as a smoothing parameter, with the non-differentiable dual function being recovered in the limit of zero temperature. Using this smoothing method in the dual problem corresponds to using maximum-entropy regularization in the linear-programming relaxation of MAP estimation based on the pseudo-marginal polytope.

- Minimizing the smoothed dual function with respect to the Lagrange multipliers is equivalent to minimizing the free energy of the decomposed graphical model over all possible decompositions of the potentials. We find that the optimal potential decomposition is characterized by the condition that the marginal distributions on duplicated subsets of variables should be equal in the Gibbs distribution defined on the decomposed graphical model. Furthermore, minimizing the smoothed dual function by block coordinate descent in the Lagrange multipliers is equivalent to an iterative projection method which iteratively enforces the marginal matching constraints by an information projection step. This leads to a simple iterative algorithm to minimize the smoothed dual function, one that is closely related to the well-known iterative scaling method for fitting a graphical model to a specified set of edge-wise marginal distributions.

- We perform this iterative marginal matching procedure at each temperature and gradually reduce the temperature towards zero ultimately to solve the Lagrangian dual problem. If this leads to a consistent MAP estimate in the decomposed model, then it is the optimal MAP estimate. In cases where there is a duality gap, we also propose a heuristic method to obtain an approximate solution from the marginal estimates produced at low (but non-zero) temperatures. Essentially, these low-temperature marginal estimates provide a way to "break ties" that occur in the zero-temperature solution.

- In the context of binary variable models, we also consider an adaptive method to select new subgraphs to include in the graphical decomposition of the model when there is a duality gap. This is similar to earlier work of Barahona [12, 13] as well as recent work of David Sontag [198]. However, our approach is based on a slightly different perspective. We use the fact that strong duality in the Lagrangian decomposition method is equivalent to satisfiability of a collection of

edgewise optimality constraints that arise from the optimal potential decomposition. Strong duality holds if and only if these edgewise constraints are jointly satisfiable. Hence, if there is a duality gap, we search for a minimal inconsistent subgraph such, such as a "frustrated cycle", and then include this as a subgraph in the dual decomposition approach. When it is tractable to identify these subgraphs, and to handle them exactly in the decomposition method, this reduces the duality gap.

- We also develop the general Lagrangian decomposition method for the class of "convex decomposable" Gaussian graphical models. These are Gaussian graphical models in which the information matrix may be expressed as a sum of positive definite matrices defined on small subsets of variables. We show that strong duality holds on this model class and develop a parametric form of the iterative scaling method to minimize the Lagrangian dual function and obtain the optimal MAP estimate. This convex decomposable property generalizes the "pairwise normalizable" condition that has been found to be important for convergence and correctness of other iterative methods in Gaussian graphical models [47,157]. Thus, our Lagrangian relaxation approach has been shown to succeed on a broader class of models than that is known to be tractable using other methods.

- Lastly, we consider a multi-scale approach to Lagrangian relaxation and demonstrate this method for Gaussian models. This approach involves introducing coarse-scale representations of an MRF, with variables that are defined from blocks of fine-scale variables. Then, relaxing the cross-scale constraints and using the Lagrangian decomposition method within each scale, we obtain a tractable dual problem. One motivation for this approach is to accelerate the rate at which information is propagated throughout the graph in the iterative scaling method so as to more efficiently minimize the dual function. It is also hoped this approach will later prove useful in discrete models by providing a tractable method to approximate effects involving large blocks of variables, possibly leading to reduced duality gaps in problems where these effects are important.

■ 6.1.2 Maximum Entropy Relaxation

To address the problem of learning graphical models, we have proposed an information-theoretic relaxation of the maximum entropy method, in which approximate marginal matching constraints are imposed over the edges of a (generalized) graph by requiring that the marginal distributions of the variables within each edge are close to specified marginals (e.g., from sample data) as measured by relative entropy. We then seek the maximum entropy model subject to these approximate marginal matching constraints. An important feature of this formulation is that it provides a convex approach to model selection, that is, solution of this convex optimization problem over a complex graphical models results in a simpler model defined on a sparse graph. The level of model thinning that occurs in this approach depends on the level of relaxation in the

marginal constraints. Thus, is it also possible to trade-off model fidelity and complexity in this approach. The most interesting aspects of our formulation and methods are summarized below:

- We show that the graphical structure of the optimal MER distribution is determined by the subset of approximate marginal matching constraints that are tight (satisfied with equality) in the final solution. In other words, edges for which the approximate marginal matching condition is satisfied laxly (with strict inequality) are thinned from the graph. In this way, we allow the maximum-entropy principle to select which subset of edges in a complex distribution are most important to obtain a good fit to the probability distribution being learned.

- We develop a "bootstrap" procedure to solve MER efficiently over a complex, densely connected graph by incrementally building up the solution by starting with a disconnected graph and adding a few edges at a time, the ones corresponding to the largest marginal divergences. If the MER solution is a thin graphical model, then we are able to recover the solution using this bootstrap method.

- An important aspect of this method is a primal-dual interior point method to solve MER efficiently on thin, chordal graphs. We modify the standard algorithm to take advantage of efficient inference algorithms for thin graphical models. This entails either exploiting sparsity of the Fisher information matrix in thin chordal graphs, or using related fast inference-based algorithms to implement the procedures of multiplication of a vector by either the Fisher information matrix or by its inverse.

- We use a Lagrangian decomposition method to derive a dual formulation of the MER problem as minimizing a convex function of the potentials of the graphical model. This results in robust maximum-likelihood approach using the MER feasible set to represent uncertainty in the sufficient statistic of the model. Our formulation of the MER constraints seems particularly well-suited from this point of view, as level-sets of relative entropy provide a good approximation to high-probability sets of the sample moments according to Sanov's theorem.

- This dual formulation of MER is also found to be equivalent to a regularized form of maximum likelihood, where an information-geometric regularization term is included that prefers selection of sparse graphical models (with many potentials set to zero). Importantly, unlike other regularization methods that have been proposed [9, 72, 147], our method is invariant to reparameterizations of the exponential family. That is, the probability distribution that is selected by MER depends only on the choice of exponential family (a set of probability distributions), not on the specifics of how we happen to parameterize it (e.g., using a Boltzmann representation versus an Ising representation for a binary model).

- We derive a block-coordinate descent method to minimize this dual function and find that it corresponds to a relaxed version of the standard iterative scaling algorihtm. This relaxed algorithm includes an additional line-search step, which is interpreted as applying a "damped" version of the step that would occur in the standard iterative scaling algorithm. This is also interpreted as an iterative projection method to enforce the MER approximate marginal matching constraints, but where there is an additional "model thinning" step that removes a potential if doing so does not violate the approximate marginal matching constraint on the corresponding edge.

■ 6.2 Recommendations for Further Work

This research suggests many other directions for further work and extensions of the methods developed so far. We summarize a few of the most promising ideas below.

■ 6.2.1 Extensions of Lagrangian Relaxation

Using ℓ_1-Regularization to Learn Good Decompositions

In our Lagrangian relaxation approach we must decide what set of subgraphs to be used in the relaxation. We have shown that it is essentially the set of maximal cliques used in these subgraphs that ultimately determines the value of the dual problem and whether or not there is a duality gap. In fact, the block relaxation method obtains the same dual value. Thus, we are faced with the question of what set of blocks one should choose. In a sense, simpler relaxations correspond to subspaces of more complex relaxations, where the potential decomposition assigns zero potentials to some blocks. In many regards, the problem of selecting which subsets of blocks to use in the Lagrangian decomposition method is analogous to the model selection problem of selecting what sets of (generalized) edges to use in a graphical model. Based on our work on the MER method for model selection, and its relation to regularized maximum-likelihood methods, it would seem that an analogous approach would be appropriate for resolving this "model selection" question in the Lagrangian decomposition method.

Let us recall the "reparameterization" view of Lagrangian relaxation. The block-decomposition version of our Lagrangian relaxation method is equivalent to the following problem:

$$\begin{array}{ll} \text{minimize} & \mathcal{F}_\beta(\theta^\dagger) \triangleq \beta^{-1} \sum_{E \in \tilde{\mathcal{G}}} \Phi_E(\beta \theta^{[E]}) \\ \text{subject to} & \sum_{E \in \tilde{\mathcal{G}}} \theta^{[E]} = \theta \end{array} \tag{6.1}$$

This is actually the "smoothed" version of the dual function with temperature parameter β^{-1}. Here, we are using notation similar to that used to describe the MER dual problem in Chapter 5. The overall parameter vector θ is being decomposed into edge-wise potentials $\theta^{[E]}$ defined over the (generalized) edges of a graph $\tilde{\mathcal{G}}$. Thus, $\tilde{\mathcal{G}}$ represents the set of blocks in the Lagrangian decomposition. This may be any super-graph of the graph $\mathcal{G}$ over which the original θ vector was defined (we zero-pad this vector to define

a model over $\tilde{\mathcal{G}}$). In the zero-temperature limit, this is equivalent to minimizing the MAP value in the decomposed problem, where each edge is maximized independently of the others.

As in our approach to MER, we may allow this graph $\tilde{\mathcal{G}}$ to be very dense and then modify the LR objective to include a regularization term on $\theta^\dagger$, the decomposed potential representation, that favors selection of sparse decompositions using a subset of the edges of $\tilde{\mathcal{G}}$ that are most useful for minimizing the dual. For instance, we could instead consider the convex optimization problem:

$$\begin{array}{ll} \text{minimize} & \hat{\mathcal{F}}_{\beta,\gamma}(\theta^\dagger) \triangleq \mathcal{F}_\beta(\theta^\dagger) + \gamma\|\theta^\dagger\|_1 \\ \text{subject to} & \sum_{E\in\tilde{\mathcal{G}}} \theta^{[E]} = \theta \end{array} \tag{6.2}$$

Here, $\gamma > 0$ is a regularization parameter that serves to control the sparsity of the decomposition. Of course, solving this problem directly is no more efficient than solving the LR problem over $\tilde{\mathcal{G}}$ directly, which may be costly due to $\tilde{\mathcal{G}}$ including many blocks and perhaps large blocks as well. However, as in our approach to solving MER, one may instead consider a "bootstrap" method to identify which subset of edges $E \in \tilde{\mathcal{G}}$ are actually used in the optimal regularized solution. In this way, the problem of deciding which are the most important blocks to use in the LR method is decided through solution of a convex optimization problem. In the regularized version of the problem, the moment-matching conditions that arise in minimizing the smoothed dual function will be replaced by approximate moment-matching conditions similar to MER. It would also be appealing to develop information-theoretic formulations of this idea (similar to MER and its dual problem) and a relaxed version of the iterative scaling method that we currently use to solve LR.

Lagrangian Relaxation for Conditionally-Gaussian Models

In this thesis, we have developed separate algorithms for solving Lagrangian relaxations of the MAP problem in either discrete or Gaussian graphical models. An obvious direction for further work is to address the more general class of *conditionally Gaussian graphical models* [144, 145], which combine these two models to obtain a hybrid class of graphical models having both discrete and continuous variables. The model is said to be *Conditionally Gaussian* (CG) if the conditional distribution of the continuous variables is Gaussian for each possible assignment of the discrete variables.

To provide a simple example, let us consider the class of *binary* CG graphical models, that is, where all the discrete variables are binary valued, e.g. $x_v \in \{0, 1\}$ as in the Boltzmann machine representation. We may parameterize this model as an exponential family by combining the features of the Gaussian and Boltzmann models and by also allowing for features that are products of discrete and continuous variables. Let $\phi^{(d)}(x^{(d)})$ and $\phi^{(c)}(x^{(c)})$ respectively denote the set of features defined on the binary variables $x^{(d)}$ and the continuous variables $x^{(c)}$ in these two models. Also, let each of these feature sets include a trivial feature $\phi_\emptyset(x) = 1$ corresponding to the empty set. Then, the most general CG model is obtained by defining its feature set to be the

product of the Gaussian and Boltzmann feature sets, that is, by define features

$$\phi_{\alpha,\beta}(x) \triangleq \phi_\alpha^{(d)}(x_\alpha^{(d)}) \times \phi_\beta^{(c)}(x_\beta^{(c)}) \tag{6.3}$$

for all $\alpha \in \mathcal{I}^{(c)}$ and $\beta \in \mathcal{I}^{(d)}$. For example, this model includes coupling terms between the discrete and continuous variables such as $x_u^{(d)} x_v^{(c)}$, $x_u^{(d)} (x_v^{(c)})^2$, $x_u^{(d)} x_v^{(c)} x_w^{(c)}$ and $x_u^{(d)} x_v^{(d)} (x_w^{(c)})^2$. Basically, we can define any monomial in the problem variables in which the degree of each binary variable is one and the sum of degrees of the continuous variables is no greater than two. This defines the full CG model without imposing any Markov structure. To defined a CG graphical model, we base the model on a sparse subset of these features corresponding to the edges of a graph or generalized graph (e.g., the set of cliques of a pairwise graph).

Let us see how this defines a Gaussian model for each configuration of the discrete variables. For each configuration of the discrete variables the energy function is a quadratic function of the continuous variables. Hence, it may be expressed in the information form, but where (h, J) depend on the discrete variables, e.g.:

$$h(x^{(d)}) = h(0) + \sum_\alpha \phi_\alpha^{(d)}(x_\alpha^{(d)}) \Delta h^{(\alpha)} \tag{6.4}$$

$$J(x^{(d)}) = J(0) + \sum_\alpha \phi_\alpha^{(d)}(x_\alpha^{(d)}) \Delta J^{(\alpha)} \tag{6.5}$$

where the contribution $(\Delta h^{(\alpha)}, \Delta J^{(\alpha)})$ is sparse, so that each α only contributes to the information parameters associated with those continuous variables that are linked to the discrete variables $x_\alpha^{(d)}$ in the graphical model. For this model to be well-posed (normalizable), it must hold that $J(x^{(d)})$ is positive definite for all $x^{(d)}$. For example, this condition is satisfied if we require that $J(0)$ is positive definite and that each $\Delta J^{(\alpha)}$ is positive semi-definite. Then, we see that this model is conditionally Gaussian.

To implement the Lagrangian decomposition method for a CG graphical model, we may begin by splitting the combined model into separate Gaussian and Boltzmann components. Then, these two components may be relaxed separately using the decomposition methods developed in the thesis. However, it remains to determine how to relax the cross-product potentials of the model that couple discrete and continuous variables. The simplest approach would be handle each such "mixed" term as a separate component of the relaxation. For example, the interaction between a single continuous variable x_c and a single binary variable x_d is described by a potential function of the form:

$$f(x_c, x_d; \theta, \lambda) = (\theta_1 + \lambda_1) x_c + (\theta_2 + \lambda_2) x_c^2 + (\theta_3 + \lambda_3) x_d + \theta_4 x_c x_d + \theta_5 x_c^2 x_d \tag{6.6}$$

The Lagrange multipliers λ_1, λ_2 and λ_3 serve to parameterize the potential decomposition. That is, to maintain a valid decomposition, we must subtract $\lambda_1 x_c + \lambda_2 x_c^2$ from the potential of the Gaussian model and subtract $\lambda_3 x_d$ from the potential of the Boltzmann

model. Using a similar approach as developed in the thesis, minimizing a regularized version of the dual function reduces to matching moments for duplicated model features. The novel aspect of this in the CG model is that we must now match moments between such compound Gaussian components and the purely Gaussian or purely discrete components of the relaxation. For example, we need to adjust the Lagrange multipliers (λ_1, λ_2) to force the mean and variance of the variable x_c to be the same in the the marginal distribution of the Gibbs distribution based on this cross-product term and in the Gaussian component of the decomposition. Note that the distribution of x_c in this cross-term component is now a mixture of two Gaussians. Hence, moment matching is no longer equivalent to marginal matching. Similarly, we adjust the parameter λ_3 to ensure that the marginal probabilities of x_d are the same in the cross-term component and in the Boltzmann component. Using these kinds of ideas, it should be possible to extend the methods of the thesis to handle the wider class of CG graphical models. The main complication will be in understanding how to modify the iterative scaling method to deal with compound Gaussian components of the decomposition.

Discrete-Gaussian Relaxations

Another related idea is to use Gaussian relaxations for inference in discrete models. For example, given an Ising model defined on the variables $x_v \in \{-1, +1\}$, we may introduce a redundant set of continuous variables $\tilde{x}_v \in \mathbb{R}$ and then split the potentials of the Ising model between the discrete and continuous variables. This leads to the dual problem of finding the optimal splitting of the potentials, between the discrete and continuous variables, so as to minimize the sum of the MAP values within each model. This idea turns out to be intimately related to semidefinite programming relaxations for MAP estimation.

There are a number of ways one might restrict how potentials are decomposed so as to ensure that the discrete component of the problem is then tractable. The simplest approach is to require that the objective function of the discrete variables is restricted to be completely separable (corresponding to a disconnected graph on the set of discrete variables). For example, consider the binary quadratic optimization problem:

$$\max_{x \in \{-1,+1\}^n} \left[-\frac{1}{2} x^T J x + h^T x \right] \tag{6.7}$$

The value of this problem is bounded above by:

$$\begin{aligned} \Psi(\lambda, \gamma) \quad = \quad & \max_{\tilde{x} \in \mathbb{R}^n} \left[-\frac{1}{2} \tilde{x}^T (J + \mathrm{Diag}(\gamma)) \tilde{x} + (h + \lambda)^T \tilde{x} \right] + \\ & \max_{x \in \{-1,+1\}} \left[-\frac{1}{2} x^T (-\mathrm{Diag}(\gamma)) x + (-\lambda)^T x \right] \end{aligned} \tag{6.8}$$

for all values of the Lagrange multipliers $\lambda, \gamma \in \mathbb{R}^n$. This comes about by relaxing the

constraints $\tilde{x}_v = x_v$ and $\tilde{x}_v^2 = 1$. The dual function is then given by

$$\Psi(\lambda, \gamma) = (h + \lambda)^T (J + \text{Diag}(\gamma))^{-1} (h + \lambda) + \sum_v (\gamma_v + |\lambda_v|) \tag{6.9}$$

if $J + \text{Diag}(\gamma) \succ 0$ and $\Psi(\lambda, \gamma) = +\infty$ otherwise. Then, the dual problem is to minimize this dual function over all $\lambda, \gamma \in \mathbb{R}^n$ such that $J + \text{Diag}(\gamma) \succ 0$. In fact, this is essentially a dual representation of the semi-definite programming (SDP) relaxation method of Goemans and Williamson [94].

However, rather than using the standard semidefinite programming method, it may be of interest to apply our physics-based methods to solve the dual problem. That is, we define a temperature parameter $\tau > 0$, and instead minimize the function:

$$\begin{aligned} \hat{\Psi}_\tau(\lambda, \gamma) &= (h + \lambda)^T (J + \text{Diag}(\gamma))^{-1} (h + \lambda) + \tau \log \det(J + \text{Diag}(\gamma)) \\ &\quad + \sum_v (\gamma_v + \tau \log \sum_{x_v} \exp\{-\tau^{-1} \lambda_v x_v\}) \end{aligned} \tag{6.10}$$

Here, we have added a log-determinant barrier function to enforce the positive-definiteness constraint and replaced the maximum over x by the smooth "log-sum-exp" approximation. The resulting regularized dual function is then equal to the sum of the log-partition function of a Gaussian model and that of an Ising model. Minimizing this function with respect to λ and γ then reduces to reparameterizing between these two models so as to minimize the sum, which reduces to matching moments between the discrete and continuous variables. That is, we adjust each γ_v so that $\mathbb{E}\{\tilde{x}_v^2\} = 1$ and adjust each λ_v so that $\mathbb{E}\{\tilde{x}_v\} = \mathbb{E}\{x_v\}$. This moment matching is performed for each value of the temperature τ. As we gradually reduce τ to zero, the optimal values of λ and γ converge to the minimum of the dual function.

To improve upon this simple method, we could also allow more structure in the discrete component of the decomposition. For instance, we could allow interactions among the discrete variables corresponding to some tractable subgraph, such as an embedded tree, thin subgraph, or a set of small, non-overlapping blocks. In this case, edges shared between the discrete and Gaussian model must also be appropriately optimized, by including a multiplier λ_{uv} that is adjusted so as to enforce the moment matching condition $\mathbb{E}\{\tilde{x}_u \tilde{x}_v\} = \mathbb{E}\{x_u x_v\}$. We could even use block relaxations within the discrete component of the model as we have done previously for discrete models. One other interesting possibility is suggested if the graphical model being relaxed is a planar Ising model. Then, we could allow both the Gaussian and Ising models to use the full planar graph $\mathcal{G}$ but now require that the discrete model is a zero-field Ising model (forcing the linear node potentials entirely into the Gaussian component). Then, using methods for exact computation of the log-partition function of the zero-field, planar Ising model [79, 87, 128, 132], we can efficiently evaluate the dual function. In this case, however, we cannot enforce the constraint $\mathbb{E}\{\tilde{x}_v\} = \{x_v\}$, but do still enforce the constraints $\mathbb{E}\{\tilde{x}_v^2\} = 1$ and edge-wise constraints $\mathbb{E}\{\tilde{x}_u \tilde{x}_v\} = \mathbb{E}\{x_u x_v\}$.

One benefit of using Gaussian relaxations of discrete models is that they provide performance guarantees using a simple randomized rounding algorithm. This amounts to drawing a random sample of the resulting Gaussian distribution of $\tilde{x}$, and then rounding it to an integral estimate: $\hat{x}_v = \text{sign}(\tilde{x}_v) \in \{-1, +1\}$ for all v. In the case of the simple relaxation in which the discrete variables are independent, this is equivalent to the Goemans-Williamson rounding method, which is guaranteed to come within .87 of the optimal MAP value [94].

■ 6.2.2 Extensions of Maximum Entropy Relaxation

Approximate Entropy for Learning Non-Thin Graphs

Perhaps the most important extension of MER is to allow for learning non-thin graphical models. Although exact inference is exponential in the tree-width of a graph, a number of principled approaches have been developed for approximate inference in non-thin graphs. Indeed, most recent applications of graphical models actually involve non-thin graphs, usually in combination with some approximate inference method. Hence, it seems to be overly restrictive that we should only attempt to learn thin graphical models. Moreover, the actual probability distributions that we seek to learn are often not thin, or even well-approximated by some thin graphical model. Thus, it is essential to consider extensions of MER using approximate inference methods to allow learning non-thin graphical models.

Here we present some simple extensions of our approach that would allow this. One challenge here is that approximate inference methods are typically tailored to the graph structure of a given model. In MER, we are trying to learn this graph structure and cannot rely on such information in advance. For this reason, we propose an approximation method that is agnostic with respect to graph structure. First, we consider the following "block" approximation to entropy. For block size s, we define the entropy approximation:

$$H(\eta) \approx \bar{H}_s(\eta) \triangleq \frac{n}{s}\binom{n}{s}^{-1} \sum_{|E|=s} H(\eta_{[E]}) \tag{6.11}$$

The idea here is that the average entropy per node $\frac{H(\eta)}{n}$ is approximated by the block-wise estimates $\frac{H(\eta_{[E]})}{|E|}$ averaged over all $\binom{n}{s}$ blocks of size s. We could use this as our regularization function in MER as a tractable alternative to $H(\eta)$. All of our approaches can easily extend to use this objective function and are then tractable to solve even if the solution of this modified MER problem is not a thin graphical model.

We suggest one other tree-based "block" approximation. Consider the graphical model in which a block E is fully connected and we also connect every other node to all nodes within this block. This is basically a simple tree model in which x_E defines a central "hub" node, and the other variables are modeled as being conditionally

independent given x_E. The entropy of this tree model is:

$$H_E^*(\eta) = \sum_{E'=E\cup v, v\notin E} H(\eta_{[E']}) - (n - |E| - 1)H(\eta_{[E]}) \tag{6.12}$$

We then obtain an entropy approximation by averaging over all such tree models based on "hub node" of size $s-1$:

$$\begin{aligned}
\bar{H}_s^*(\eta) &= \binom{n}{s-1}^{-1} \sum_{|E|=s-1} H_E^*(\eta) \\
&= \binom{n}{s-1}^{-1} \left\{ s \sum_{|E'|=s} H(\eta_{[E']}) - (n-s) \sum_{|E|=s-1} H(\eta_{[E]}) \right\} \\
&= (n-s+1)\bar{H}_s(\eta) - (n-s)\bar{H}_{s-1}(\eta)
\end{aligned}$$

Using these simple approximations, we expect MER can be used to successfully learn good approximations to non-thin graphical models. The complexity of these methods grows with the number of blocks $\binom{n}{s} \leq \frac{n^s}{s!}$ and with the complexity of inference within each block, e.g., $\mathcal{O}(|\mathbb{X}|^s)$ in discrete models. Some other ideas might involve using Gaussian approximations to the entropy or mixtures of trees, but where the mixture weights are adaptively chosen as a function of η.

Non-Convex Extensions for Improving Sparsity

Another possibility is to develop a *non-convex* extension of the MER dual problem aimed at improving the sparsity of the graphical model. As we have discussed, the MER dual problem minimizes the function:

$$\mathcal{F}(\theta^\dagger) = \mathcal{F}_0(\theta) + \sum_{E\in\tilde{\mathcal{G}}} \Psi_E(\theta^{[E]}) \tag{6.13}$$

where $\mathcal{F}_0(\theta) = \Phi(\theta) - \tilde{\eta}^T\theta$ is the objective function one minimizes in the maximum-likelihood method and $\sum_E \Psi'_E(\theta^{[E]})$ is a regularization term which, somewhat similar to the ℓ_1-regularization, favor selection of sparser models. A number of methods have been developed which instead use non-convex ℓ_p-regularization, with $0 < p < 1$, to obtain sparser solutions in parameter estimation [75]. However, this idea does not seem to have been explored in the context of graphical model selection. In MER, we might consider instead minimizing the function:

$$\mathcal{F}_p(\theta^\dagger) = \mathcal{F}_0(\theta) + \gamma \left(\sum_{E\in\tilde{\mathcal{G}}} \Psi_E^p(\theta^{[E]}) \right)^{\frac{1}{p}} \tag{6.14}$$

This formulation still enjoys the property that it is based on the information geometric terms $\Psi_E(\theta^{[E]})$, so as to be invariant to reparameterizations of the exponential family,

but it should also lead to stronger preference towards sparser graphical model. In fact, as $p \to 0$ the regularization term is equal to the number of edges of the selected subgraph $\hat{\mathcal{G}}$. Also similar to the ℓ_p-regularization method, it could prove useful to first solve this problem for $p = 1$, for which the problem is convex, and then use the continuation approach to "track" the solution as p is decreased. This may be helpful to find a good approximation to the global minima of the non-convex problem for $p < 1$.

Learning Hidden Variable and Multi-Scale Models

Finally we mention the important question of how to learn hidden variable and multi-scale graphical models. Often, the set of observed variables may not be distributed according to a Markov model. But, by including a small number of additional variables (that were not seen in the sample data) it then becomes possible to explain the distribution of the observed variables as the marginal distribution of a much sparser graphical model defined on both the observed and hidden variables. Thus, it is desirable to develop a principled extension of the MER method to learn such hidden variable models. One possible non-convex method would be to apply the MER idea within the context of the latent maximum entropy principle of [215] for learning latent-variable models. This would essentially involve combining the MER method with the expectation-maximization method [68] for maximum-likelihood modeling. However, it is the fact that MER is a convex method for learning model structure that makes it so appealing. It would be far preferably to find a convex method for learning hidden variable models. However, it is unclear at this time if that is possibly using some generalization of MER.

A related but simpler question is that of learning maximum-entropy models using a non-local features of x. In this thesis, we have formulated and solved MER in terms of locally defined features, e.g. x_v at a node or $x_u x_v$ on an edge. In some problems, the random variables are naturally regarded as being arranged spatially to form a random field, such as a two-dimensional grid in image processing. Then, one may consider introducing *coarse-scale* features of the random field and applying MER to learn the multi-scale model of the random field. In fact, an approach along these lines has recently been developed by Myung Choi [54]. However, I expect that there is more work to be done in this direction. In particular, here are few outstanding problems that remain to be resolved: obtaining a principled method to handle such degenerate multi-scale models in MER (the coarse-scale variables are really a deterministic function of fine-scale variables), to allow for non-deterministic upward models and to also learn the coarse-to-fine scale mapping that provides the best model.

Appendix A

Lagrangian Relaxation Using Subgraph Decompositions

In this appendix we provide further details concerning extension of the Lagrangian relaxation method to handle subgraph decompositions, which were introduced previously in Section 3.2.2. For the most part, the development of subgraph decompositions is identical to that for the block decomposition method, with some minor changes in the definitions of the matrices D, $\hat{D}$ and C, and some additional comments on how the iterative scaling method is implemented for subgraph decompositions.

■ A.1 Subgraph Decompositions Revisited

As discussed in Section 3.2.2, this approach involves breaking up the energy function $f(x) = \sum_{E \in \mathcal{G}} f_E(x_E)$ among a set of energy functions $f^{(k)}(x^{(k)}) = \sum_{E \in \mathcal{G}^{(k)}} f_E^{(k)}(x_E^{(k)})$ defined over subgraphs $\mathcal{G}^{(k)}$ of $\mathcal{G}$. This defines an objective function $f^\dagger(x^\dagger) = \sum_k f^{(k)}(x^{(k)})$ over an auxiliary graphical model defined on a larger graph $\mathcal{G}^\dagger$, which is comprised of a set of disconnected components corresponding to each of the subgraphs $\mathcal{G}^{(k)}$. The dual problem is then to minimize $g(f^\dagger) \triangleq \max f^\dagger(x^\dagger) = \sum_k \max f^{(k)}(x^{(k)})$ over all valid decompositions of the objective $f(x)$ between the component-wise objectives $f^{(k)}(x^{(k)})$ defined on each subgraph.

Exponential Family Description

Now, we describe this from the perspective of the exponential family representation. The original objective is represented by $f(x) = \theta^T \phi(x)$, where each feature $\phi_\alpha(x_{E_\alpha})$ is defined over some edge $E_\alpha \in \mathcal{G}$. We may express each of the subgraph energy functions as $f^{(k)}(x^{(k)}) = (\theta^{(k)})^T \phi^{(k)}(x^{(k)})$, where its feature set $\phi^{(k)}$ is restricted to the subset of features which have support within some edge of $\mathcal{G}^{(k)}$. Thus, each energy function $f^{(k)}(x^{(k)})$ is constrained to respect the graphical structure of the subgraph $\mathcal{G}^{(k)}$. This is important in the subgraph decomposition method so as to ensure that the dual problem is tractable to compute. The auxiliary objective is then given by $f^\dagger(x^\dagger) = (\theta^\dagger)^T \phi^\dagger(x^\dagger)$ where $\theta^\dagger = (\theta^{(k)}, k = 1, \ldots, m)$ is a vector consisting of the component-wise parameter vectors $\theta^{(k)}$, and $\phi^\dagger(x^\dagger) = (\phi^{(k)}(x^{(k)}), k = 1, \ldots, m)$ is the vector of all the sufficient

statistics on every component of $\mathcal{G}^\dagger$. Then, a valid decomposition of θ is any set of component-wise parameter vectors $\theta^{(k)}$ that satisfy $\sum_k [\theta^{(k)}]_{\mathcal{I}} = \theta$.

Although this is somewhat more involved than the simpler block decomposition method, we can use basically the same notation to describe the Lagrangian relaxation method. Let D denote the linear mapping from the original set of variable x, defined on the vertices of $\mathcal{G}$, to the set of auxiliary variables $x^\dagger = (x^{(k)}, k = 1, \ldots, m)$. That is, in $x^\dagger = Dx$, each variable x_v gets copied into $x^{(k)}$ for each subgraph $\mathcal{G}^{(k)}$ that contains vertex v. Similarly, let $\hat{D}$ denote the linear mapping from $\phi(x)$ to $\phi^\dagger(x^\dagger)$, such that, for consistent $x^\dagger$, we have $\phi^\dagger(x^\dagger) = \hat{D}\phi(x)$. The condition that $\theta^\dagger$ is a valid decomposition of θ is expressed as $\hat{D}^T\theta^\dagger = \theta$. Then it holds that $f^\dagger(Dx) = f(x)$ for all x.

Finally, we define the matrix C so as to define self-consistency among the features of $\phi^\dagger(x^\dagger)$. These constraints require that for any pair of features in $\phi^\dagger$ that are copies of the same feature in ϕ, the corresponding values of $\phi^\dagger(x^\dagger)$ should be equal (thereby forcing $x^\dagger$ to be consistent). That is, if the map $\phi^\dagger = \hat{D}\phi$ copies ϕ_α into $\phi_\alpha^{(k)}$ and $\phi_\alpha^{(l)}$ (the copies of feature α within subgraphs k and l), then we include a constraint $\phi_\alpha^{(k)}(x^{(k)}) - \phi_\alpha^{(l)}(x^{(l)}) = 0$. The set of all such consistency constraints is then encoded as $C\phi^\dagger(x^\dagger) = 0$.

Lagrangian Dual Problem

With these changes of notation, the description of the Lagrangian relaxation procedure and resulting dual problem is otherwise identical to the block decomposition method. However, it may be worthwhile to step through these once more, emphasizing their interpretation in the subgraph decomposition method.

Let $\tilde{\theta}^\dagger$ be any initial valid decomposition of θ among the subgraphs. Then, we consider the constrained MAP problem on $\mathcal{G}^\dagger$:

$$\text{(MAP-}\mathcal{G}^\dagger\text{)} \qquad \begin{array}{ll} \text{maximize} & \sum_k (\tilde{\theta}^{(k)})^T \phi^{(k)}(x^{(k)}) \\ \text{subject to} & C\phi^\dagger(x^\dagger) = 0 \end{array}$$

Introducing Lagrange multipliers to relax the self-consistency constraint $C\phi^\dagger(x^\dagger) = 0$ and maximizing over $x^\dagger$, we obtain the dual function:

$$g(\lambda) = \sum_k \max_{x^{(k)}} \left\{ (\tilde{\theta} + C^T\lambda)^T_{\mathcal{I}^{(k)}} \phi^{(k)}(x^{(k)}) \right\}$$

Now, the Lagrange multipliers serve to parameterize valid decompositions between the subgraphs. It is also tractable to compute the maximum of each subgraph's objective $f^{(k)}$, owing to its being defined over a thin graph $\mathcal{G}^{(k)}$. This gives an upper-bound of $f^* = \max f$ for all λ. Then, the Lagrangian dual problem is to minimize this upper-bound over all λ, which is equivalent to a minimizing $g(\theta^\dagger) = \sum_k \max_{x^{(k)}} (\theta^{(k)})^T \phi^{(k)}(x^{(k)})$ over all valid decompositions of θ among the subgraphs (now defined such that each $\theta^{(k)}$ respects the graphical structure of its subgraph $\mathcal{G}^{(k)}$, that is, it does not use any other edges of $\mathcal{G}$ that are not included in subgraph $\mathcal{G}^{(k)}$).

Gibbsian Smoothing and Maximum-Entropy Regularization

It is worth also commenting on the interpretation of the Gibbsian smoothing method and its dual interpretation as maximum-entropy regularization in the case of subgraph decompositions. The smoothed dual function may be defined as:

$$\hat{g}_\beta(\theta^\dagger) = \sum_k \log \sum_{x^{(k)}} \exp\{(\theta^{(k)})^T \phi^{(k)}(x^{(k)})\}$$

This is the free energy of a Gibbs distribution defined on the graph $\mathcal{G}^\dagger$, which is a sum of the free energies on each subgraph. Because these subgraphs are thin, this free energy can be computed efficiently using the recursive inference method. Then, the smoothed dual problem is to minimize this free energy over all valid decompositions, that is, over all $\theta^\dagger$ such that $\sum_k [\theta^{(k)}]_\mathcal{I} = \theta$, which is equivalently written $\hat{D}^T \theta^\dagger = \theta$.

This smoothed version of the dual problem is again related to a maximum-entropy regularized LP over the marginal polytope $\mathcal{M}(\mathcal{G}^\dagger)$:

$$\text{(Gibbs-}\mathcal{M}^\dagger\text{)} \qquad \begin{array}{ll} \text{maximize} & (\tilde{\theta}^\dagger)^T \eta^\dagger + \beta^{-1} H^\dagger(\eta^\dagger) \\ \text{subject to} & \eta^\dagger \in \mathcal{M}(\mathcal{G}^\dagger) \\ & C\eta^\dagger = 0 \end{array}$$

where $H^\dagger(\eta^\dagger) = \sum_k H^{(k)}(\eta^{(k)})$ represents the entropy of the Gibbs distribution defined on $\mathcal{G}^\dagger$. However, in order for each subgraph's entropy $H^{(k)}(\eta^{(k)})$ to be easily computed, we must also require that each of the subgraphs $\mathcal{G}^{(k)}$ represents a thin *chordal* graph. This then ensures that $H^{(k)}(\eta^{(k)})$ further decomposes using the junction tree decomposition (2.134) as a sum of entropies on maximal cliques minus a sum of entropies on separators of this chordal subgraph. Thanks to the consistency constraint $C\eta^\dagger = 0$, this optimization over $\mathcal{M}(\mathcal{G}^\dagger)$ is equivalent to one over $\mathcal{M}(\mathcal{G})$:

$$\text{(Gibbs-}\mathcal{M}\text{)} \qquad \begin{array}{ll} \text{maximize} & \theta^T \eta + \beta^{-1} H^\dagger(\hat{D}\eta) \\ \text{subject to} & \eta \in \hat{\mathcal{M}}(\mathcal{G}) \end{array}$$

If each subgraph is chordal, then $H^\dagger(\hat{D}\eta)$ now defines a block-based approximation to the entropy $H(\eta)$ and serves as a barrier function over the pseudo-marginal polytope $\hat{\mathcal{M}}$. However, it should be recognized that this entropy function $H^\dagger(\hat{D}\eta)$ is not simply a sum of block-wise entropies as in the block decomposition method. This is because the junction-tree entropy decomposition (2.134) of each chordal subgraph also includes negative entropy terms on the separators of its junction tree, which accounts for the fact that summing entropy over maximal cliques "double counts" entropy on these separators. However, the effective domain of this entropy function (over which it is well-defined) will be the same as in the block decomposition method using the set of all maximal cliques of the subgraphs.

In the temperature annealing method, this will ultimately lead to *exactly the same* zero-temperature solution as one would obtain using the block decomposition method

based on the set of maximal cliques appearing in the subgraphs. Only the *path* of solutions leading to that zero-temperature solutions is changed by using the subgraph decomposition method instead. However, in cases in which there is a duality gap, the subgraph decomposition method could lead to an improvement in the finite-temperature estimation heuristic for obtaining approximate MAP estimates.

■ A.2 Comments on Iterative Scaling Using Subgraphs

The iterative scaling algorithm we use to minimize the Lagrangian dual function in subgraph decompositions is all but identical to the one specified in Section 3.3.3 for block decompositions, with the following changes:

1. **Definition of Update Sets** In the block decomposition method, we performed updates on sets of nodes $S \subset V$ that are contained in multiple blocks $E \in \mathcal{G}$. In subgraph decompositions, we perform updates on subsets of nodes that are contained within multiple subgraphs, but with the additional restriction that S must be contained within some clique of each subgraph that participates in the corresponding update. This latter restriction is necessary to ensure that the update does not introduce any new edges in the subgraphs being updated.

2. **Recursive Inference** In the block decomposition method, it is feasible to compute marginals within each block by brute-force summation over all values of the other variables (provided the block size is kept small). In subgraph decompositions, we must instead use the recursive inference method to compute marginal distributions efficiently. This involves building a junction tree of each subgraph and performing tree-structured message-passing over the junction tree to obtain the marginal distributions. If each subgraph is thin, the computational complexity of these inference calculations is linear in the number of nodes contained within the subgraph (but exponential in the treewidth of the subgraph).

3. **Updating Messages** To implement the iterative scaling algorithm efficiently in the subgraph decomposition method, it is also important to store previously computed messages of the junction tree inference algorithm so as to avoid redundant calculations and minimize the cost of inference in the iterative scaling method. This basically means that we keep track of the last clique that was updated within each subgraph's junction tree, retaining all messages over this junction tree that are directed towards that clique. Then, when we next compute the marginal distribution of another clique within this same junction tree, we only need to recompute those messages along the directed path from the last clique that was updated to this new clique. All the other messages directed towards this new clique will have already been computed. This then provides an efficient implementation of the iterative scaling method using general subgraph decompositions.

Appendix B

Proof of Strong Duality in Ferromagnetic Models

The purpose of this appendix is to prove Proposition 3.2.4, which states that if there is duality gap in the pairwise relaxation of a pairwise model, then the model is not ferromagnetic (it contains at least one anti-ferromagnetic edge potential). A key part of the proof uses an *implication graph* based on the pairwise MAP sets $\hat{X}^{ij}$ over the edges $\{i,j\} \in \mathcal{G}$ of the graphical model. Implication graphs were used in [7] as the basis for a linear-time algorithm for solving the 2-SAT problem. This algorithm is also useful for checking if strong duality holds in pairwise relaxations of binary variable models. We first review this method, and then proceed to the proof.

We say that the pairwise MAP set $\hat{X}^{ij}$ is *ferromagnetic* if it is one of $\{++,--\}$, $\{++,--,+-\}$ or $\{++,--,-+\}$. It is *anti-ferromagnetic* if it is one of $\{+-,-+\}$, $\{+-,-+,++\}$ or $\{+-,-+,--\}$.

■ B.1 The Implication Graph and Test for Strong Duality

Each of the MAP sets $\hat{X}^{ij}$, viewed as a constraint on x, can be translated into a logically equivalent set of *implications* using the following set of rules. Let i_+ and i_- respectively denote the assertions $x_i = +1$ or $x_i = -1$. In the implication graph (to be defined), these will serve as two nodes that replace node i of the graphical model. In general, $\hat{X}^{ij}$ may be any non-empty subset of $\{++,--,+-,-+\}$. Thus, there are 15 possible values of $\hat{X}^{ij}$. However, many of these cannot lead to any inconsistency. For instance, we do not need to consider the case that $\hat{X}^{ij} = \{++,--,+-,-+\}$, as the constraint $(x_i, x_j) \in \hat{X}^{ij}$ is always satisfied. Similarly, if one or both nodes are resolved (e.g. $\hat{X}^{ij} = \{++\}$ or $\hat{X}^{ij} = \{+-,++\}$) then this edge cannot lead to any inconsistency.[1] Thus, it is only necessary to consider the following two classes of MAP sets involving just the unresolved nodes. We have the following equivalence between anti-ferromagnetic constraints $x \in \hat{X}^{ij}$ and logical implications between assertions i_+,

[1] As discussed in the sequel, an implication graph is consistent if there are no directed cycles of the graph that visit both i_+ and i_- for some i. However, if i is resolved than one of the two nodes i_+ or i_- is deleted and the other is an isolated vertex of the implication graph (with no edges to or from this vertex). Thus, there cannot be any cycle that visits a resolved node.

i_-, j_+ and j_-:

$$\begin{aligned}(x_i, x_j) \in \{-+, +-\} &\Leftrightarrow (i_+ \Leftrightarrow j_- \text{ and } i_- \Leftrightarrow j_+)\\ (x_i, x_j) \in \{-+, +-, --\} &\Leftrightarrow (i_+ \Rightarrow j_- \text{ and } j_+ \Rightarrow i_-)\\ (x_i, x_j) \in \{-+, +-, ++\} &\Leftrightarrow (i_- \Rightarrow j_+ \text{ and } j_- \Rightarrow i_+)\end{aligned} \tag{B.1}$$

These can be shown to be logically equivalent by considering the four possible values of $(x_i, x_j) \in \{++, --, -+, +-\}$ and verifying that the ones allowed on the left are all the values that do not contradict any of the implications on the right. Note that these anti-ferromagnetic constraints correspond to implications that can only force neighbors to have opposite values, e.g., no constraint forces j_+ if i_+ holds. On the other hand, for the ferromagnetic constraints we have:

$$(x_i, x_j) \in \{--, ++\} \Leftrightarrow (i_+ \Leftrightarrow j_+ \text{ and } i_- \Leftrightarrow j_-) \tag{B.2}$$

$$(x_i, x_j) \in \{--, ++, -+\} \Leftrightarrow (i_+ \Rightarrow j_+ \text{ and } i_- \Rightarrow j_-) \tag{B.3}$$

$$(x_i, x_j) \in \{--, ++, +-\} \Leftrightarrow (i_- \Rightarrow j_- \text{ and } i_+ \Rightarrow j_+) \tag{B.4}$$

Ferromagnetic constraints correspond to implications that can only force neighbors to have the same value.

Now, the *implication graph* is defined so as to encode the constraints $x \in \hat{X}^{ij}$ for all $\{i, j\} \in \mathcal{G}$. This graph is based on the vertex set $V_+ \cup V_- \triangleq \{v_+, v \in V\} \cup \{v_-, v \in V\}$ and with *directed* edges determined by the above equivalence between the optimality conditions $x \in \hat{X}^{ij}$ and implications between the nodes of the implication graph. For instance, $i_+ \Leftrightarrow j_-$ is encoded as a symmetric pair of edges (i_+, j_-) and (j_-, i_+), whereas $i_+ \Rightarrow j_-$ maps to a single edge (i_+, j_-). Note that all anti-ferromagnetic constraints result in edges between V_+ and V_- and all ferromagnetic constraints results in edges that either have both endpoints in V_+ or both endpoints in V_-.

With this encoding, the collection of pairwise MAP sets $\{\hat{X}^{ij}\}$ is satisfiable if and only if the implication graph is consistent. Consistency of the implication graph is defined by the condition that there is no directed cycle of the graph that visits both i_+ and i_- for some $i \in V$. This is the basis of the linear-time algorithm [7] for solving 2-SAT. It identifies the strongly-connected components of this directed graph and then checks if any component contains both i_+ and i_- for some node $i \in V$. Based on Proposition 3.2.1 and the preceding translation of MAP sets into implications, this also gives a linear time algorithm for checking if there is a duality gap in pairwise relaxations of binary variable models.

■ B.2 Proof of Proposition 3.2.4

Lemma B.2.1. *If $\hat{X}^{ij}$ is anti-ferromagnetic, then it holds that $f_{ij}(x_i, x_j)$ is anti-ferromagnetic, that is, $\sigma(f_{ij}) < 0$.*

Proof. Recall, from our discussion of Section 3.2.6, that $\sigma(f_{ij}) \triangleq f_{ij}(++) + f_{ij}(--) - f_{ij}(+-) - f_{ij}(-+)$ is *invariant to reparameterizations* of the model. In the Ising model,

this condition is equivalent to $\theta_{ij} < 0$. If $\hat{X}^{ij} = \{+-, -+\}$ then we must have that $f_{ij}(+-) = f_{ij}(-+) = \max f_{ij} \triangleq f_{ij}^*$ and both $f_{ij}(--)$ and $f_{ij}(++)$ are less than f_{ij}^*. Hence, $\sigma(f_{ij}) = f_{ij}(--) + f_{ij}(++) - 2f_{ij}^* < 0$. Similarly, if $\hat{X}^{ij} = \{+-, -+, ++\}$ then

$$f_{ij}(+-) = f_{ij}(-+) = f_{ij}(++) = f_{ij}^*,$$

and $f_{ij}(--) < f_{ij}^*$. Hence, we again find that $\sigma(f_{ij}) = f_{ij}(--) - f_{ij}^* < 0$. □

Lemma B.2.2. *If there is a duality gap, then there must exist at least one edge $\{i,j\} \in \mathcal{G}$ for which the MAP set $\hat{X}^{ij}$ is anti-ferromagnetic.*

Proof. We prove this using the implication graph construction. According to Proposition 3.2.1, strong duality $g^* = f^*$ holds if and only if the collection of pairwise MAP sets are jointly satisfiable. As discussed in the preceding section, this is equivalent to the existence of an inconsistent cycle in the implication graph based on the collection of edge-wise constraints $x \in \hat{X}^{ij}$ for all $\{i,j\} \in \mathcal{G}$. By definition, an inconsistent cycle of the implication graph is a directed cycle which, for some vertex $i \in V$, visits both the nodes i_+ and i_- in the implication graph. If there exists such an inconsistent cycle, clearly it must include at least one directed edge from some node in V_+ to some node in V_-, which corresponds to an edge $\{i,j\} \in \mathcal{G}$ with $\hat{X}^{ij}$ being either $\{-+, +-\}$ or $\{-+, +-, --\}$. Likewise, there must be at least one directed edge from V_- to V_+, corresponding to some edge where $\hat{X}^{ij}$ is either $\{-+, +-\}$ or $\{-+, +-, ++\}$. Thus, there exist at least one edge where $\hat{X}^{ij}$ is either $\{-+, +-\}$, $\{-+, +-, --\}$ or $\{-+, +-, ++\}$. □

Proof of Proposition 3.2.4 By Proposition 3.2.1, existence of a duality gap implies that the sets $\{\hat{X}^{ij}\}$ are not jointly satisfiable. By Lemma B.2.2, this implies that there is at least one edge for which $\hat{X}^{ij}$ is anti-ferromagnetic. Then, by Lemma B.2.1, this edge must have an anti-ferromagnetic edge potential. Hence, if there is a duality gap, the model cannot be ferromagnetic, and strong duality must always hold in ferromagnetic models. □

Appendix C

Möbius Transform and Fisher Information in Boltzmann Machines

This appendix is based on an unpublished technical note [121] written to support our work on the MER primal-dual interior point in Boltzmann machines [123] presented in Chapter 5 of the thesis. We have recently found the related work of [30] (see also references therein).

We consider the exponential family representation of Boltzmann machines (a collection of n binary-valued random variables) and study the algebraic structure of this family with respect to the Möbius transform. Also, we introduce a generalized class of Möbius transforms and introduce a "fast" algorithm for computing these transforms in $(n-1)2^{(n-1)}$ calculations (rather than $\mathcal{O}(2^{2n})$ calculations). We consider possible implications of this algorithm for inference and learning in Boltzmann machines.

■ C.1 Preliminaries

Throughout this note we will be concerned with functions that are either defined on the set of all subsets of $\{1,...,n\}$ or that are defined on $\{0,1\}^n$. Both of these domains have cardinality 2^n and can therefore be identified with the set $\{0,1,\ldots,2^n-1\}$.

To each subset $x \subseteq \{1,...,n\}$ we identify a bit-vector $x_n \ldots x_1 \in \{0,1\}^n$ with $x_i = 1$ if $i \in x$ and $x_i = 0$ otherwise. To each $x \in \{0,1\}^n$ we can also identify an integer $x = \sum_{i=1}^n x_i 2^{i-1}$. Thus, $x_n \ldots x_1$ is just the binary expansion of the integer x. Employing this convention, we may view x as specifying either an integer, a bit-vector or a subset. For example, with $n = 2$ we have the following "states":

$$\begin{aligned} 0 &= 00 = \emptyset \\ 1 &= 01 = \{1\} \\ 2 &= 10 = \{2\} \\ 3 &= 11 = \{1,2\} \end{aligned} \tag{C.1}$$

We will write $x \preceq y$ (equivalently, $y \succeq x$) if $x \subseteq y$ (equivalently, $x_i \leq y_i$ for $i = 1,...,n$).

A set-valued function $f(x)$ (equivalently, a function of n binary variables $f(x_1,...,x_n)$) can then be represented as a vector in $\mathbb{R}^{2^n}$ by defining the x-th element of the vector

representation of f (with indices running from 0 to $2^n - 1$) to be $f(x)$. Let $\delta(x)$ denote the standard basis vector of $\mathbb{R}^{2^n}$, e.g. for $n = 2$:

$$\begin{aligned}\delta(0) &= (1,0,0,0)^T \\ \delta(1) &= (0,1,0,0)^T \\ \delta(2) &= (0,0,1,0)^T \\ \delta(3) &= (0,0,0,1)^T\end{aligned} \tag{C.2}$$

Employing this convention, we may rewrite $f(x)$ as $f^T\delta(x)$.

■ C.2 The Möbius Transform

We consider a generalized class of Möbius transforms. Let $f \in \mathbb{R}^{2^n}$. We define the ω-transform by:

$$(L_n^\omega f)(x) = \sum_{y \preceq x} \omega^{|x \setminus y|} f(y) \tag{C.3}$$

where $|x \setminus y|$ is this number of elements of x that are not contained in y (the number of bits that are 1 in x and 0 in y). Note that this defines a family $\{L_n^\omega, \omega \in \mathbb{C}\}$ of linear operators on $\mathbb{R}^{2^n}$. The usual Möbius transform is given by $L_n \equiv L_n^1$ and the inverse Möbius transform by L_n^{-1}.

For example, let $n = 2$, $f \in \mathbb{R}^4$ and $g = L_2^\omega f$:

$$\begin{pmatrix} g(00) \\ g(01) \\ g(10) \\ g(11) \end{pmatrix} = \begin{pmatrix} 1 & & & \\ \omega & 1 & & \\ \omega & 0 & 1 & \\ \omega^2 & \omega & \omega & 1 \end{pmatrix} \begin{pmatrix} f(00) \\ f(01) \\ f(10) \\ f(11) \end{pmatrix} \tag{C.4}$$

We now summarize some interesting properties of the ω-transform.

Proposition C.2.1. *For $n = 0$, $L_0^\omega = 1$ for all ω. For $n \geq 1$:*

$$L_n^\omega = \begin{pmatrix} L_{n-1}^\omega & 0 \\ \omega L_{n-1}^\omega & L_{n-1}^\omega \end{pmatrix} \tag{C.5}$$

Proof. Let $g = L_n^\omega f$, $f = (f_1, f_2)$ and $g = (g_1, g_2)$ with $f, g \in \mathbb{R}^{2^n}$ and $f_i, g_i \in \mathbb{R}^{2^{n-1}} (i = 1, 2)$. We must show $g_1 = L_{n-1}^\omega f_1$ and $g_2 = \omega L_{n-1}^\omega f_1 + L_{n-1}^\omega f_2$. For $x, y \in \{0, \ldots, 2^n - 1\}$, $x \leq y$ implies that $x \preceq y$ (bitwise). Thus,

$$g_1(x) = \sum_{y : y < 2^{n-1}, y \preceq x} \omega^{|x \setminus y|} f_1(y) = (L_{n-1}^\omega f_1)(x) \tag{C.6}$$

Let $\pi_n(x) = x \mod 2^{n-1}$. For $x \in \{2^{n-1}, \ldots, 2^n - 1\}$ this has the effect of setting the

n-th bit of x to zero, i.e. $\pi_n(x_n x_{n-1} \dots x_1) = 0 x_{n-1} \dots x_1$. Now, for $x \geq 2^{n-1}$ we have:

$$\begin{aligned} g_2(x) &= \sum_{y: y<2^{n-1}, y \preceq x} \omega^{|x \setminus y|} f_1(y) + \sum_{y: y \geq 2^{n-1}, y \preceq x} \omega^{|x \setminus y|} f_2(\pi_n(y)) \\ &= \sum_{y: y<2^{n-1}, y \preceq \pi_n(x),} \omega^{|\pi_n(x) \setminus y)|+1} f_1(y) + \sum_{y: y \geq 2^{n-1}, \pi_n(y) \preceq \pi_n(x)} \omega^{|\pi_n(x) \setminus \pi_n(y)|} f_2(\pi_n(y)) \\ &= \omega(L^{\omega}_{n-1} f_1)(x) + (L^{\omega}_{n-1} f_2)(x) \end{aligned} \tag{C.7}$$

which proves the proposition. □

This result provides the basis for computing fast ω-transforms. Given $f = (f_1, f_2) \in \mathbb{R}^{2^n}$ with $f_1, f_2 \in \mathbb{R}^{2^{n-1}}$ we can compute the transform of f as $\tilde{f} = (\tilde{f}_1, \omega \tilde{f}_1 + \tilde{f}_2)$ where $\tilde{f}_1$ and $\tilde{f}_2$ are the ω-transforms of f_1 and f_2. Computing these recursively, we obtain an algorithm to apply the operator L^{ω}_n which requires $(n-1)2^{n-1}$ computations rather than $\mathcal{O}(2^{2n})$.

Proposition C.2.2. *The ω-transforms form a commutative group:*

1. $L^{\alpha}_n L^{\beta}_n = L^{\alpha+\beta}_n = L^{\beta}_n L^{\alpha}_n$.
2. $L^0_n = I_{2^n}$.
3. $(L^{\omega}_n)^{-1} = L^{-\omega}_n$.

Proof. We show (1) as follows. First, for $n = 0$ it trivially holds that $L^{\alpha}_0 L^{\beta}_0 = 1 = L^{\alpha+\beta}_0$. Then, by Proposition 1 and induction on n we have:

$$\begin{aligned} L^{\alpha}_{n+1} L^{\beta}_{n+1} &= \begin{pmatrix} L^{\alpha}_n & 0 \\ \alpha L^{\alpha}_n & L^{\alpha}_n \end{pmatrix} \begin{pmatrix} L^{\beta}_n & 0 \\ \beta L^{\beta}_n & L^{\beta}_n \end{pmatrix} \\ &= \begin{pmatrix} L^{\alpha}_n L^{\beta}_n & 0 \\ (\alpha+\beta) L^{\alpha}_n L^{\beta}_n & L^{\alpha}_n L^{\beta}_n \end{pmatrix} \\ &= \begin{pmatrix} L^{\alpha+\beta}_n & \\ (\alpha+\beta) L^{\alpha+\beta}_n & L^{\alpha+\beta}_n \end{pmatrix} \\ &= L^{\alpha+\beta}_{n+1} \end{aligned} \tag{C.8}$$

The remaining points are then self-evident. □

There also is an "upper" ω-transform defined as:

$$(U^{\omega}_n f)(x) = \sum_{y \succeq x} \omega^{|y \setminus x|} f(y) \tag{C.9}$$

Note that the sum is now over all supersets of a set rather than the subsets. However, it turns out it is just the transpose of the "lower" ω-transform, i.e. $U^{\omega}_n = (L^{\omega}_n)^T$. There also is a fast $(n-1)2^{(n-1)}$ implementation of upper ω-transforms.

■ C.3 Boltzmann Machines

We now consider the family of probability distributions on $\{0,1\}^n$ which may be parameterized in the form of an exponential family:

$$p(x) = p(x_1, \ldots, x_n) = \exp\{\theta^T \phi(x)\} \tag{C.10}$$

with exponential parameters $\theta \in \mathbb{R}^{2^n}$ and sufficient statistics $\phi(x) = (\phi_a(x), a \subseteq \{1, ..., n\})$ defined by $\phi_a(x) = \prod_{i \in a} x_i$. Note that $\phi_a(x) = 1$ if $x_i = 1$ for all $i \in a$ and is zero otherwise. The moment parameters of the family are defined by $\eta = \sum_x p(x)\phi(x)$.

Both η and θ are naturally viewed as vectors in $\mathbb{R}^{2^n}$ indexed by subsets of $\{1, ..., n\}$. Parameter $\theta(s)$ is the multiplier of $\phi_s(x)$. The moment $\eta(s)$ is the probability that $x_i = 1$ for all $i \in s$. Likewise, the probability mass function p may also be viewed as an element of $\mathbb{R}^{2^n}$ indexed by joint states $(x_1, \ldots, x_n)$ of the n binary variables.

Next, we show that these three vector representations p, η and θ are connected by the Möbius transform:

Proposition C.3.1. *We have the following Möbius transform relations:*

1. $\phi(x) = L_n^T \delta(x)$ *and* $\delta(x) = L_n^{-T} \phi(x)$.
2. $\eta = L_n^T p$ *and* $p = L_n^{-T} \eta$.
3. $p = \exp(L_n \theta)$ *and* $\theta = L_n^{-1} \log p$.

Proof. (1) Element s of $\phi(x)$ is one if $s \preceq x$ (bitwise) and is zero otherwise. On the other hand, we have

$$(L_n^T \delta(x))(s) = \sum_{t \succeq s} \delta(x)(t) \tag{C.11}$$

which also is one if $s \preceq x$ and is zero otherwise. (2) $\eta = \mathbb{E}\{\phi(x)\} = \mathbb{E}\{L_n^T \delta(x)\} = L_n^T \mathbb{E}\{\delta(x)\} = L_n^T p$. (3) $\log p(x) = (\log p)^T \delta(x) = \theta^T \phi(x) = \theta^T L_n^T \delta(x) = (L_n \theta)^T \delta(x)$ for all x. Hence, $\log p = L_n \theta$. □

Thus, inference (computing η from θ) corresponds to the map:

$$\Lambda(\theta) = L_n^T \exp(L_n \theta) \tag{C.12}$$

while learning (computing θ from η) corresponds to the inverse map:

$$\Lambda^{-1}(\eta) = L_n^{-1} \log(L_n^{-T} \eta) \tag{C.13}$$

Both maps require $n2^n$ calculations using the fast Möbius transform.

Normalization We should point out that all three representations are over-parameterized as we have not yet imposed the normalization constraint $1^T p = 1$. In the η parameterization, normalization reduces to the requirement that $1^T L_n^{-T} \eta = (L_n^{-1} 1)^T \eta = \delta(\emptyset)^T \eta = \eta(\emptyset) = 1$. In the θ parameterization, we must have:

$$-\theta(\emptyset) = \Phi(\theta_{\setminus\emptyset}) \equiv \log \sum_x \exp \sum_{s \neq \emptyset} \theta_s \phi_s(x) \tag{C.14}$$

The function Φ is known as the *cumulant generating function* of the exponential family (in statistical physics it is called the *log-partition function*).

Marginalization The marginal distribution on a subset of random variables s is a function on the set of all joint states of $x_s = (x_i, i \in s)$ defined as:

$$p_s(x_s) = \sum_{y : y_i = x_i \forall i \in s} p(y) \tag{C.15}$$

for each $x_s \in \{0,1\}^{|s|} \equiv \mathbb{R}^{2^{|s|}}$. This defines a linear map $\Sigma_s : \mathbb{R}^{2^n} \to \mathbb{R}^{2^{|s|}}$, i.e. $p_s = \Sigma_s p$.

We now consider marginalization in the context of the η parameterization. Each subset $s \subseteq \{1, \ldots, n\}$ determines an injective map $\sigma_s : \{1, \ldots, |s|\} \to \{1, \ldots, n\}$ defined by ordering the elements, i.e. $s = \{\sigma_s(k), k = 1, \ldots, |s|\}$ where $\sigma_s(1) < \sigma_s(2) < \cdots < \sigma_s(|s|)$. Thus, given $t \subseteq s$ we have that $\sigma^{-1}(t)$ is the corresponding subset of $\{1, \ldots, |s|\}$ (an element of $\{0, \ldots, 2^{|s|}\}$). Now, we may define Π_s by $(\Pi_s f)(\sigma^{-1}(t)) = f(t)$ for all $t \preceq s$. Thus, Π_s just gathers the elements of f which correspond to subsets of s. We now show that $\eta_s = \Pi_s \eta$ are precisely the moment parameters associated to the marginal distribution p_s, i.e.:

Proposition C.3.2. $L_{|s|}^T \Sigma_s = \Pi_s L_n^T$.

Proof. For $t \subseteq s$ we have:

$$\begin{aligned} \eta(t) &= \sum_{u \in \{0,1\}^{|s|} : u_i = 1, \forall i \in \sigma^{-1}(t)} \; \sum_{v \in \{0,1\}^{n-|s|}} p(\sigma_s(u) \cup \sigma_{\setminus s}(v)) \\ &= \sum_{u : u_i = 1, \forall i \in \sigma^{-1}(t)} p_s(u) \\ &= \eta_s(\sigma^{-1}(t)) \end{aligned} \tag{C.16}$$

□

■ C.4 Fisher Information

We consider the second moment of the statistics $\phi(x)$:

$$\begin{aligned} K(\theta) &= \mathbb{E}_\theta\{\phi(x)\phi^T(x)\} \\ &= L_n^T \mathbb{E}_\theta\{\delta(x)\delta^T(x)\} L \\ &= L_n^T \mathrm{Diag}(p_\theta) L_n \end{aligned} \tag{C.17}$$

which is a symmetric positive semi-definite matrix. Also, we define the inverse matrix parameterized by η:

$$\begin{aligned} K^*(\eta) &\equiv K^{-1}(\Lambda^{-1}(\eta)) \\ &= L_n^{-1}\mathrm{Diag}(1/p_\eta)L_n^{-T} \end{aligned} \tag{C.18}$$

Note that both matrices submit to a "fast" implementation as a linear operator employing the fast Möbius transforms. It is easily verified that:

$$\begin{aligned} K(\theta) &= \frac{\partial \Lambda(\theta)}{\partial \theta} \\ K^*(\eta) &= \frac{\partial \Lambda^{-1}(\eta)}{\partial \eta} \end{aligned} \tag{C.19}$$

Hence, we expect these linear operators may prove useful in variational methods for inference and learning. In fact, these matrices are closely related to the Fisher information matrices associated with the η and θ parameterizations (imposing normalization eliminates $\theta_\emptyset$ and $\eta_\emptyset$):

$$\begin{aligned} G(\theta) &= (K(\theta) - \Lambda(\theta)\Lambda(\theta)^T)_{\setminus\emptyset,\setminus\emptyset} \\ G^*(\eta) &= K^*(\eta)_{\setminus\emptyset,\setminus\emptyset} \end{aligned} \tag{C.20}$$

Note also, $G(\theta)$ is the Schur complement of $K(\theta)$ obtained by eliminating the first row/column associated with the $\theta_\emptyset$ parameter since

$$K = \begin{pmatrix} 1 & \eta_{\setminus\emptyset}^T \\ \eta_{\setminus\emptyset} & K_{\setminus\emptyset,\setminus\emptyset} \end{pmatrix} \tag{C.21}$$

and $G = K_{\setminus\emptyset,\setminus\emptyset} - \eta_{\setminus\emptyset}\eta_{\setminus\emptyset}^T$.

Appendix D

Fisher Information in Gaussian Graphical Models

This appendix is based on an unpublished technical note [120], written to support our work on the primal-dual interior point method for solving MER in Gaussian graphical models [123] presented in Chapter 5 of the thesis.

We summarize various derivations, formulas and computational algorithms relevant to the Fisher information matrix of Gaussian graphical models with respect to either an exponential parameterization (related to the information form) or the corresponding moment parameterization.

■ D.1 Gauss-Markov Models

The probability density of a Gaussian random vector $x \in \mathbb{R}^n$ may be expressed in *information form*:

$$p(x) \propto \exp\{-\frac{1}{2}x^T Jx + h^T x\}$$

with parameters $h \in \mathbb{R}^n$ and $J \in \mathbb{R}^{n \times n}$, where J is a symmetric positive-definite matrix. The mean and covariance of x are given by:

$$\begin{aligned} \mu &\triangleq \mathbb{E}_p[x] = J^{-1}h \\ K &\triangleq \mathbb{E}_p[(x-\mu)^T(x-\mu)] = J^{-1} \end{aligned}$$

where we denote expectation of $f(x)$ with respect to p by $\mathbb{E}_p[f] \triangleq \int p(x)f(x)dx$. In this note, we focus on the zero-mean case $\mu = 0$. Thus, $h = 0$ as well.

A Gaussian distribution is *Markov* on a graph $\mathcal{G} = (V, E)$, with vertices $V = \{1, \ldots, n\}$, if and only if $J_{ij} = 0$ for all $\{i, j\} \notin E$. Thus, Markov models have a reduced information parameterization based on a sparse J matrix.

■ D.2 Exponential Family and Fisher Information

The Gaussian distributions can also be represented as an *exponential family*:

$$p(x) = \exp\{\theta^T \phi(x) - \Phi(\theta)\}$$

with parameters $\theta \in \mathbb{R}^d$ and sufficient statistics $\phi : \mathbb{R}^n \to \mathbb{R}^d$. To obtain the information form we define statistics:

$$\phi(x) = (x_i^2, i \in V) \cup (x_i x_j, \{i,j\} \in E)$$

The corresponding exponential parameters then correspond to elements of the information matrix:

$$\theta = (-\frac{1}{2} J_{ii}, i \in V) \cup (-J_{ij}, \{i,j\} \in E)$$

An implicit parameterization of the exponential family is given by the *moment parameters* defined by $\eta = \mathbb{E}_p[\phi]$. In the Gauss-Markov family on $\mathcal{G}$, the moment parameters correspond to elements of the covariance matrix K:

$$\eta = (K_{ii}, i \in V) \cup (K_{ij}, \{i,j\} \in E)$$

The θ and η parameters are related by a bijective map: $\eta = \Lambda(\theta) \triangleq p_\theta[\phi]$.

The *cumulant generating function* of this exponential family is

$$\begin{aligned} \Phi(\theta) &\triangleq \log \int \exp\{\theta^T \phi(x)\} dx \\ &= \frac{1}{2}\{\log\det J^{-1}(\theta) + n \log 2\pi\} \end{aligned}$$

which serves to normalize the probability distribution. Derivatives of this function yield cumulants of the distribution. In particular, the gradient $\nabla\Phi(\theta) = (\frac{\partial\Phi(\theta)}{\partial\theta_i})$ is equal to the moment map:

$$\nabla\Phi(\theta)_i = \frac{\partial\Phi(\theta)}{\partial\theta_i} = \Lambda_i(\theta) = \eta_i$$

The Hessian matrix $\nabla^2\Phi(\theta) = (\frac{\partial^2\Phi(\theta)}{\partial\theta_i\partial\theta_j})$ is equal to the covariance of the sufficient statistics, which incidentally is also the *Fisher information* of the exponential family with respect to θ parameters:

$$G(\theta) \triangleq \nabla^2\Phi(\theta) = p_\theta[(\phi(x) - \Lambda(\theta))(\phi(x) - \Lambda(\theta))^T] = p_\theta[\nabla \log p_\theta(x) \nabla \log p_\theta(x)^T]$$

Note, since $\nabla\Phi(\theta) = \Lambda(\theta)$, the Fisher information matrix is also equal to the Jacobian $D\Lambda(\theta) = (\frac{\partial\eta_i}{\partial\theta_j})$ and therefore describes the first-order sensitivity of the moments to perturbations in θ parameters: $\delta\eta \approx G(\theta)\delta\theta$.

Consider the function $f(X) = \log\det X$. Boyd and Vandenberge derive first and second order differential analysis. The gradient is $\nabla f(X) = X^{-1}$, that is to say that $f(X + dX) \approx f(X) + \mathrm{Tr}(X^{-1}dX)$ where $\mathrm{Tr}(AB)$ is the natural inner product of two matrices A and B. The Hessian is naturally described by the action of a quadratic form on the vector space of symmetric matrices:

$$\nabla^2 f(X)(dX, dY) = -\mathrm{Tr}(X^{-1}dXX^{-1}dY)$$

This is related to the fact that for the function $F(X) = X^{-1}$ we have $dF = -X^{-1}dXX^{-1}$, a matrix generalization of $d(1/x) = -dx/x^2$.

Using these relations, we can evaluate explicit formulas for the elements of the Fisher information matrix $G(\theta)$ in a Gauss-Markov model. We start by writing $J(\theta)$ as

$$J(\theta) = -2\sum_{i\in V} \theta_i e_i e_i^T - \sum_{\{i,j\}\in E} \theta_{ij}(e_i e_j^T + e_j e_i^T) \tag{D.1}$$

where $\{e_i\}$ are the standard basis vectors of $\mathbb{R}^n$. Note that $V \cup E$ serves as the index set of θ, which has dimension $d = |V| + |E|$. Thus, $\frac{\partial J(\theta)}{\partial \theta_i} = -2e_i e_i^T$ and $\frac{\partial J(\theta)}{\partial \theta_{ij}} = -(e_i e_j^T + e_j e_i^T)$. Now, we evaluate the matrix elements of $G(\theta)$ using the chain rule and the quadratic form for the Hessian of the log-determinant function. For example, the element corresponding to edges $\{i,j\}$ and $\{k,l\}$ is obtained as:

$$\begin{aligned} G_{ij,kl} &= \tfrac{1}{2}\mathrm{Tr}(J^{-1}\frac{\partial J}{\partial \theta_{ij}} J^{-1}\frac{\partial J}{\partial \theta_{kl}}) \\ &= \tfrac{1}{2}\mathrm{Tr}(K(e_i e_j^T + e_j e_i^T)K(e_k e_l^T + e_l e_k^T)) \\ &= K_{il}K_{jk} + K_{ik}K_{jl} \end{aligned}$$

Here, we have used $\mathrm{Tr}(Ke_i e_j^T Ke_k e_l^T) = \mathrm{Tr}(e_l^T Ke_i e_j^T Ke_k) = K_{li}K_{jk}$. Similarly, we obtain $G_{ij,k} = 2K_{ik}K_{jk}$ for edge-to-vertex terms and $G_{i,k} = 2K_{ik}^2$ for vertex-to-vertex terms.

These formulas for the elements of G are also valid for the Fisher information in any Markov sub-family of the full Gaussian model, because the Fisher information in the Markov model is a submatrix of the Fisher information in the full model. This nesting relation follows from the fact that the Markov sub-family is obtained by constraining some θ parameters to zero, so the Hessian of the resulting cumulant-generating function is a submatrix of the Hessian in the unconstrained model.

In practice, we may not need to explicitly evaluate the Fisher information matrix. For instance, we may only wish to apply the Fisher information matrix to a given vector $\Delta\theta$ in order to compute the first-order change in moments $\Delta\eta = G(\theta)\Delta\theta$. Often, this can be done more efficiently than explicitly computing $G(\theta)$ and evaluating the matrix-vector product. For instance, in the fully connected model, the explicit computation requires $\mathcal{O}(d^2) = \mathcal{O}(n^4)$ computations to compute $G(\theta)$ and evaluate the matrix-vector product. But, using the fact that $G(\theta) = \frac{\partial \eta}{\partial \theta}$ and that $d(X^{-1}) = -X^{-1}dXX^{-1}$ we can more efficiently evaluate $\Delta\eta$ as follows:

1. Let $\Delta J = J(\theta + \Delta\theta) - J(\theta)$.
2. Compute $K = J^{-1}$ and $\Delta K = -K\Delta JK$.
3. Let $\Delta\eta_i = \Delta K_{ii}$ for all $i \in V$ and $\Delta\eta_{ij} = \Delta K_{ij}$ for all $\{i,j\}$.

This approach only requires $\mathcal{O}(n^3)$, which is about n times faster than the explicit computation. In a similar manner, we can more efficiently apply the Fisher information

matrix in models with thin graphical structure that allow efficient computations of the moment parameters. We discuss this further in Section 4.

■ D.3 Moments, Entropy and Fisher Information

Entropy is defined $h(p) = -\mathbb{E}_p[\log p]$ and provides a measure of randomness or uncertainty of a random variable with distribution p. A Gaussian density with covariance K has entropy:

$$h(K) = \frac{1}{2}(\log\det K + n\log 2\pi e)$$

In the maximum-entropy approach to modeling, one seeks the probability distribution that maximizes entropy subject to linear moment constraints:

$$\begin{aligned} \max \quad & h(p) \\ \text{s.t.} \quad & \mathbb{E}_p[\phi] = \eta \end{aligned}$$

where ϕ are a set of selected statistics and η are the prescribed expected values of those statistics (for example, the empirical averages of those statistics taken from some data set). A well-known maximum-entropy principle asserts that the solution to this problem (when it exists) will be of the form of an exponential family based on ϕ where the θ are chosen to realize the prescribed moments. Let p_η denote the maximum-entropy model with moments η and $h(\eta)$ denote it's entropy.

There is an important duality principle underlying the maximum-entropy principle. The negative entropy $\Psi(\eta) \triangleq -h(\eta)$ and the cumulant generating function $\Phi(\theta)$ are convex-conjugate functions:

$$\begin{aligned} \Phi(\theta) &= \sup_\eta\{\theta^T\eta - \Psi(\eta)\} \\ \Psi(\eta) &= \sup_\theta\{\eta^T\theta - \Phi(\theta)\} \end{aligned}$$

Moreover, the optimal value of η and θ in these two variational problems are respectively $\eta = \Lambda(\theta)$ and $\theta = \Lambda^{-1}(\eta)$.

As a consequence of this duality, it holds that the gradient of negative entropy function evaluated at η is equal to the corresponding θ parameters:

$$\nabla\Psi(\eta) = \Lambda^{-1}(\eta) = \theta$$

Also, the Hessian of the negative entropy function $\Psi(\eta)$ is equal to the *inverse* of the Hessian of $\Phi(\theta)$ evaluated for the corresponding $\theta = \Lambda^{-1}(\eta)$:

$$\nabla^2\Psi(\eta) = \nabla^2\Phi(\Lambda^{-1}(\eta))^{-1}$$

Recall that $G(\theta) = \nabla^2\Phi(\theta)$ is the Fisher information associated with the θ parameters. Similarly, $G^*(\eta) \triangleq \nabla^2\Psi(\eta)$ is the Fisher information in the η parameterization:

$$G^*(\eta) = \mathbb{E}_\eta\left[\nabla\log\Psi(\eta)\nabla\log\Psi(\eta)^T\right]$$

We can also use the Hessian of the log-determinant function to compute the Fisher information $G^*(\eta)$ in the full Gaussian model (not imposing any Markov constraints). Then, K is fully parameterized by η:

$$K(\eta) = \sum_i \eta_i e_i e_i^T + \sum_{\{i,j\}} \eta_{ij}(e_i e_j^T + e_j e_i^T)$$

Thus, $\frac{\partial K}{\partial \eta_i} = e_i e_i^T$ and $\frac{\partial K}{\partial \eta_{ij}} = (e_i e_j^T + e_j e_i^T)$. For example, the edge-to-edge terms are given by:

$$\begin{aligned} G^*_{ij,kl} &= \frac{1}{2}\mathrm{Tr}(K^{-1}\frac{\partial K}{\partial \eta_{ij}}K^{-1}\frac{\partial K}{\partial \eta_{kl}}) \\ &= \frac{1}{2}\mathrm{Tr}(J(e_i e_j^T + e_j e_i^T)J(e_k e_l^T + e_l e_k^T)) \\ &= J_{ik}J_{jl} + J_{il}J_{jk} \end{aligned}$$

Similarly, we obtain $G^*_{ij,k} = J_{ik}J_{jk}$ and $G^*_{i,k} = \frac{1}{2}J^2_{ik}$. Note, these formulas differ slightly from the analogous formulas for $G(\theta)$. Again, it is worth noting that $\Delta\theta = G^*\Delta\eta$ can be more efficiently computed using: $\Delta\eta \to \Delta K \to \Delta J = -J\Delta K J \to \Delta\theta$. This is an $\mathcal{O}(n^3)$ computation whereas explicitly building and multiplying by G^* is $\mathcal{O}(n^4)$. Moreover, if J and ΔK are sparse with respect to a graph, then these computations can be implemented very efficiently, requiring $\mathcal{O}(n)$ computations in graphs with bounded degree.

However, it must be noted that the preceding specifications for G^* are only valid in the full Gaussian family. In particular, the Fisher information $\hat{G}^*$ of the Gauss-Markov family on $\mathcal{G}$ is *not* simply a sub-matrix of the full G^*, but rather is the corresponding Schur complement:

$$\hat{G}^* = G^*_{\mathcal{G},\mathcal{G}} - G^*_{\mathcal{G},\setminus\mathcal{G}}(G^*_{\setminus\mathcal{G},\setminus\mathcal{G}})^{-1}G^*_{\setminus\mathcal{G},\mathcal{G}}$$

For a sparse J matrix, we see that the full G^* becomes sparse. However, due to fill in the Schur complement, the matrix $\hat{G}^*$ will typically become a full matrix.[1] Nonetheless, the fast algorithm for G^* may still be useful as an approximate preconditioner in iterative methods (by approximating $\hat{G}^*$ by the submatrix $G^*_{\mathcal{G},\mathcal{G}}$, which neglects the "fill" term in the Schur complement).

■ D.4 Chordal Graphs and Junction Trees

In this section we consider some specialized algorithms which implement Fisher information operations efficiently in Gauss-Markov models defined on chordal graphs.

We recall that a graph is *chordal* if for each cycle of length four or more there exists a corresponding *chord*, an edge linking two non-consecutive vertices of the cycle. One equivalent characterization of a chordal graph is that is has a junction tree. Let $\mathcal{C}$ denote the set of cliques of the graph where a *clique* is any completely connected

[1] Although, we show an important exception to the rule is shown in the following section.

subsets of vertices. A tree $T = (\mathcal{C}, E_T)$, formed by linking cliques of the graph, is called a *junction tree* if for any two cliques C_α and C_β, every clique along the unique path in T from C_α to C_β is contained in the intersection of the endpoints $C_\alpha \cap C_\beta$.

Given a junction tree of the graph, we also define a corresponding set of *separators* $\mathcal{S}$, one for each edge (C_α, C_β) of the junction tree, defined as the intersection of the endpoints of the edge $S_{\alpha,\beta} = C_\alpha \cap C_\beta$. Incidentally, these are also separators in the original chordal graph. In fact, the set of conditional independence relations implied by the chordal graph $\mathcal{G}$ is exactly equivalent to those implied by the the junction tree (viewed as a Markov-tree model).

Sparsity of $G^*(\eta)$ on Chordal Graphs

Given a probability distribution $p(x)$ which is Markov on a chordal graph $\mathcal{G}$, we may express the joint probability distribution as the product of clique marginals divided by the product of separator marginals:

$$p(x) = \frac{\prod_{C \in \mathcal{C}} p_C(x_C)}{\prod_{S \in \mathcal{S}} p_S(x_S)}$$

The entropy submits to a similar decomposition in terms of the marginal entropies on the cliques and separators of the graph.

$$h(\eta) = \sum_C h_C(\eta_C) - \sum_S h_S(\eta_S)$$

Differentiating with respect to moment parameters and negating the result we obtain:

$$\Lambda^{-1}(\eta) = \sum_C \Lambda_C^{-1}(\eta_C) - \sum_S \Lambda_S^{-1}(\eta_S)$$

Here, we have used the property that $\nabla h(\eta) = \Lambda^{-1}(\eta)$, both for the global entropy and each of the marginal entropies. This relates the global map Λ^{-1} to local mappings which have a closed form solution. In the Gaussian model, this is equivalent to the computation:

$$J = \sum_C (K_C)^{-1} - \sum_S (K_S)^{-1}$$

where the terms on the right are appropriately zero-padded to be consistent with J. Thus, evaluation of $\theta = \Lambda^{-1}(\eta)$ has a simple closed form solution in chordal graphs.

If we differentiate again, we obtain a corresponding decomposition of the Fisher information matrix $G^*(\eta) = D\Lambda^{-1}(\eta) = -\nabla^2 h(\eta)$ in terms of marginal Fisher information terms over the cliques and separators of the graph:

$$G^*(\eta) = \sum_C G_C^*(\eta_C) - \sum_S G_S^*(\eta_S)$$

Using the fact that the marginal Fisher informations in moment parameters are inverses of the marginal Fisher information in exponential parameters, and those are submatrices of $G(\theta)$, we obtain:

$$G^* = \sum_C (G_C)^{-1} - \sum_S (G_S)^{-1}$$

Note, the relationship between G^* and G is analogous to that between J and K. This relation arises because the matrix G^* is itself a symmetric, positive-definite matrix defined over a chordal graph (an augmented version of $\mathcal{G}$ with vertex set $V \cup E$, corresponding to statistics ϕ, and with edges linking any two statistics that have support inside a common clique of $\mathcal{G}$).

Based on our earlier analysis of the Fisher information in the full Gaussian family, we can provide explicit formula for those marginal terms. Moreover, the resulting matrix is sparse, reflecting the same sparsity structure as in the underlying graph. Perhaps more importantly, we can now efficiently compute $\Delta\eta = G^*(\eta)\Delta\theta$ which, in matrix form, can be expressed as:

$$\Delta J = -\sum_C K_C^{-1} \Delta K_C K_C^{-1} + \sum_S K_S^{-1} \Delta K_S K_S^{-1}$$

This is an $\mathcal{O}(nw^3)$ computations where w is the width of the graph. This is more efficient than explicitly computing the sparse matrix $G^*(\eta)$ and performing the matrix-vector product $G^*(\eta)\Delta\eta$, which would require $\mathcal{O}(nw^4)$ operations (although, in graphs with very low width, the latter method may still be acceptable).

Similar as before, given a non-chordal graph $\mathcal{G}_{nc}$, we can express its Fisher information matrix (of the moment parameterization) as the Schur complement of the Fisher information of any chordal graph $\mathcal{G}_c$ that contains $\mathcal{G}_{nc}$ as a subgraph. This suggests that the preceding fast implementations of Fisher information for chordal graphs may serve as a useful approximation to the Fisher information of its embedded subgraphs.

Recursive Inference on Junction Trees

Let $T = (\mathcal{C}, E_T)$ be a junction tree of a chordal graph. We obtain a directed version of T by selecting an arbitrary clique as the root of the tree and orienting the edges away from this root node. For a given node α of the junction tree, let $\pi(\alpha)$ denote the parent of α. At each non-root node α, we split the corresponding clique C_α into disjoint subsets $S_\alpha = C_\alpha \cap C_{\pi(\alpha)}$ and $R_\alpha = C_\alpha \setminus C_{\pi(\alpha)}$. At the root node we define these as $S_\alpha = \emptyset$ and $R_\alpha = C_\alpha$.

We specify our recursive inference procedure in two passes: an "upward" leaf-to-root pass and a "downward" root-to-leaf pass. The input to this procedure is the sparse J matrix defined over a chordal graph. The output is a sparse K matrix, which contains the corresponding subset of elements in the covariance matrix.

Upward Pass The upward pass begins at the leaves of the tree and works its way up the tree performing the following computations:

$$\begin{aligned} Q_\alpha &= (J_{R_\alpha,R_\alpha})^{-1} \\ A_\alpha &= -Q_\alpha J_{R_\alpha,S_\alpha} \\ J_{S_\alpha,S_\alpha} &\leftarrow J_{S_\alpha,S_\alpha} + J_{S_\alpha,R_\alpha} A_\alpha \end{aligned}$$

In the last step, the principle submatrix of J indexed by S_α is overwritten. This serves to propagate information to the parent of node α in the junction tree. These computations are clearly equivalent to Gaussian elimination in the J matrix:

$$J_{S_\alpha,S_\alpha} \leftarrow J_{S_\alpha,S_\alpha} - J_{S_\alpha,R_\alpha} J_{R_\alpha,R_\alpha}^{-1} J_{R_\alpha,S_\alpha}$$

Thus, when we compute Q_α at non-leaf nodes, the value of J_{R_α,R_α} used above is the result of already having eliminating the descendents of node α.

We store the intermediate computations A_α and Q_α at each node of the tree as these can be reused in the following downward pass. In fact, these parameters may be interpreted as specifying the equivalent directed forward model:

$$x_{R_\alpha} = A_\alpha x_{S_\alpha} + w_\alpha$$

where w_α is zero-mean Gaussian with covariance Q_α. Essentially, the upward pass may be viewed as reparameterizing the joint distribution in the form:

$$\begin{aligned} p(x) &= \prod_\alpha p(x_{R_\alpha}|x_{S_\alpha}) \\ p(x_{R_\alpha}|x_{S_\alpha}) &\propto \exp[-\tfrac{1}{2}(x_{R_\alpha} - A_\alpha x_{S_\alpha})^T Q_\alpha^{-1}(x_{R_\alpha} - A_\alpha x_{S_\alpha})] \end{aligned}$$

It is a simple exercise to verify that this yields an equivalent information form.

Downward Pass The downward pass begins at the root of the tree and works it way back down the tree to the leaves performing the following computations in the order shown:

$$\begin{aligned} K_{R_\alpha,S_\alpha} &\leftarrow A_\alpha K_{S_\alpha,S_\alpha} \\ K_{S_\alpha,R_\alpha} &\leftarrow (K_{S_\alpha,R_\alpha})^T \\ K_{R_\alpha,R_\alpha} &\leftarrow K_{R_\alpha,S_\alpha} A_\alpha^T + Q_\alpha \end{aligned}$$

If a non-redundant symmetric storage format is used for K (e.g., only the "upper" triangular part of K is actually stored), the second step above can be omitted. This iteration computes the subset of elements of $K = J^{-1}$ corresponding to the vertices and edges of the chordal graph. The downward pass simply propagates covariances in a causal fashion according to the forward model:

$$K_{R_\alpha,R_\alpha} = \mathbb{E}_p[x_{R_\alpha} x_{R_\alpha}^T] = \mathbb{E}_p[(A_\alpha x_{S_\alpha} + w_\alpha)(A_\alpha x_{S_\alpha} + w_\alpha)^T] = A_\alpha K_{S_\alpha,S_\alpha} A_\alpha^T + Q_\alpha$$

$$K_{R_\alpha,S_\alpha} = \mathbb{E}_p[x_{R_\alpha} x_{S_\alpha}^T] = \mathbb{E}_p[(A_\alpha x_{S_\alpha} + w_\alpha) x_{S_\alpha}^T] = A_\alpha K_{S_\alpha,S_\alpha}$$

Note that this form of recursive inference only requires one matrix inverse per node in the junction tree. We also comment that the entire computation can be performed "in place" in the sparse J matrix (by over-writing the elements of J by the corresponding values of K in the downward pass) so that we do not have to simultaneously provide storage for two sparse matrices.

Differential Propagation There also is a linearized version of this algorithm which computes the first-order perturbation ΔK as a result of a change of ΔJ in the information matrix. The input to the procedure is a pair of sparse matrices J and ΔJ both defined over a chordal graph. In the differential version of the algorithm, we perform the following computations (in addition to those computations already listed above).

Upward Pass:

$$\begin{aligned}
\Delta Q_\alpha &= -Q_\alpha \Delta J_{R_\alpha,R_\alpha} Q_\alpha \\
\Delta A_\alpha &= -(Q_\alpha \Delta J_{R_\alpha,R_\alpha} + \Delta Q_\alpha J_\alpha) \\
\Delta J_{S_\alpha,S_\alpha} &\leftarrow \Delta J_{S_\alpha,S_\alpha} + \Delta J_{S_\alpha,R_\alpha} A_\alpha + J_{S_\alpha,R_\alpha} \Delta A_\alpha
\end{aligned}$$

Downward Pass:

$$\begin{aligned}
\Delta K_{R_\alpha,S_\alpha} &\leftarrow \Delta A_\alpha K_{S_\alpha,S_\alpha} + A_\alpha \Delta K_{S_\alpha,S_\alpha} \\
\Delta K_{R_\alpha,R_\alpha} &\leftarrow \Delta K_{R_\alpha,S_\alpha} A_\alpha^T + K_{R_\alpha,S_\alpha} \Delta A_\alpha^T + \Delta Q_\alpha
\end{aligned}$$

Upon completion, this computes the perturbations in the moment parameters defined on the chordal graph.

This linearized computation of ΔK over a chordal graph given J and ΔJ is equivalent to computing $\Delta\eta = G(\theta)\Delta\theta$ given θ and $\Delta\theta$ in the corresponding exponential family representation of the Gauss-Markov family defined on that chordal graph. Thus, these recursive algorithms may be viewed as an efficient method to multiply by the Fisher information matrix $G(\theta)$. The complexity of this algorithm is $\mathcal{O}(nw^3)$, where w is the width of the chordal graph (the size of the largest clique). Note that $G(\theta)$ is typically a full $d \times d$ matrix (where $d = |V| + |E|$), hence explicit evaluation of $G(\theta)\Delta\eta$ would require $\mathcal{O}(d^2)$ computations.

Bibliography

[1] P. Abbeel, D. Koller, and A.Y. Ng. Learning factor graphs in polynomial time and sample complexity. *The Journal of Machine Learning Research*, 7:1743–1788, 2006.

[2] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, December 1974.

[3] S. Amari. Information geometry on hierarchy of probability distributions. *IEEE Transactions on Information Theory*, 47(5), 2001.

[4] S. Amari, K. Kurata, and H. Nagaoka. Information geometry of Boltzmann machines. *IEEE Transactions on Neural Networks*, 3(2), 1992.

[5] S. Amari and H. Nagaoka. *Methods of Information Geometry.* Oxford University Press, 2000.

[6] E. Amir. Efficient approximation for triangulation of minimum treewidth. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pages 7–15, 2001.

[7] B. Aspvall, M. Plass, and R. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8(3), 1979.

[8] F. Bach and M.I. Jordan. Thin junction trees. In *Neural Information Processing Systems*, 2001.

[9] O. Banerjee, L. Ghaoui, A. d'Aspermont, and G. Natsoulis. Convex optimization techniques for fitting sparse Gaussian graphical models. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 89–96, 2006.

[10] F. Barahona. On the computational complexity of Ising spin glass models. *Journal of Physics A*, 15:3241–3253, 1982.

[11] F. Barahona. Finding ground states in random-field Ising ferromagnets. *Journal of Physics A*, 18:673–675, 1985.

[12] F. Barahona, M. Grötschel, M. Jünger, and G. Reinelt. An application of combinatorial optimization to statistical physics and layout design. *Operations Research*, 36(3):493–513, 1988.

[13] F. Barahona and A.R. Mahjoub. On the cut polytope. *Mathematical Programming*, 36:157–173, 1986.

[14] F. Barahona, R. Maynard, R. Rammal, and J.P. Uhry. Morphology of ground states of two-dimensional frustration model. *Journal of Physics A*, 15:673–699, 1982.

[15] O.E. Barndorff-Nielsen. *Information and Exponential Families in Statistical Theory*. Wiley, 1978.

[16] L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occuring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41:164–171, 1970.

[17] H.H. Bauschke, J.M. Borwein, and P.L. Combettes. Essential smoothness, essential strict convexity, and Legendre functions in Banach spaces. *Communications in Contemporary Mathematics*, 3(4):615–647, 2001.

[18] H.H. Bauschke and J.M. Bowein. Legendre functions and the method of random Bregman projections. *Journal of Convex Analysis*, 4, 1997.

[19] R.E. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[20] C. Berge. *Graphs and Hypergraphs*. North-Holland, 1973.

[21] A. Berry, J. Blaire, P. Heggernes, and B. Peyton. Maximum cardinality search for computing minimal triangulations of graphs. *Algorithmica*, 39:287–298, 2004.

[22] D.P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.

[23] D.P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[24] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1995.

[25] D. Bertsimas and J.N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

[26] D. Bertsimas and R. Weismantel. *Optimization over Integers*. Dynamic Ideas, 2005.

[27] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36:192–236, 1974.

[28] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society B*, 48(3):259–302, 1986.

[29] J. Bieche, R. Maynard, R. Rammal, and J.P. Uhry. On the ground states of the frustration model of a spin glass by a matching method of graph theory. *Journal of Physics A*, 13:2553–2576, 1980.

[30] A. Bjorklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets Möbius: Fast subset convolution. In *Proceedings of the thirty-ninth annual ACM Symposium on Theory of Computing (STOC'07)*, 2007.

[31] H.L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal of Computing*, 25:1305–1317, 1996.

[32] H.L. Bodlaender and A. Koster. Combinatorial optimization on graphs of bounded treewidth. *The Computer Journal*, 2007.

[33] B. Bollobás. *Modern Graph Theory.* Springer-Verlag, 2000.

[34] E. Boros, Y. Crama, and P.L. Hammer. Upper-bounds for quadratic 0-1 maximization. *Operations Research Letters*, 9:73–79, 1990.

[35] E. Boros and P.L. Hammer. The max-cut problem and quadratic 0-1 optimization; polyhedral aspects, relaxations and bounds. *Annals of Operations Research*, 33(3):1991, 1991.

[36] E. Boros and P.L. Hammer. Pseudo-Boolean optimization. *Discrete Applied Mathematics*, 123:155–225, 2002.

[37] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[38] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[39] L.M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Physics*, 7:200–217, 1967.

[40] P. Brémaud. *Markov chains, Gibbs fields, Monte Carlo simulation, and queues.* Springer-Verlag, 1999.

[41] W. Buntine. A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering*, 8(2), 1996.

[42] B. Carré. *Graphs and Networks.* Oxford University Press, 1979.

[43] E. Castillo and J.M. Gutiérrez. *Expert Systems and Probabilistic Network Models.* Springer-Verlag, 1997.

[44] M. Cetin, L. Chen, J.W. Fisher III, A. Ihler, R.L. Moses, M.J. Wainwright, and A.S. Willsky. Distributed fusion in sensor networks. *IEEE Signal Processing Magazine*, 23(4):42–55, July 2006.

[45] V. Chandrasekaran, J.K. Johnson, and A.S. Willsky. Adaptive embedded subgraph algorithms using walk-sum analysis. In *Advances in Neural Information Processing Systems*, December 2007.

[46] V. Chandrasekaran, J.K. Johnson, and A.S. Willsky. Maximum entropy relaxation for graphical model selection given inconsistent statistics. In *IEEE 2007 Statistical Signal Processing Workshop (SSP 2007)*, August 2007.

[47] V. Chandrasekaran, J.K. Johnson, and A.S. Willsky. Estimation in Gaussian graphical models using tractable subgraphs: a walk-sum analysis. *IEEE Transactions on Signal Processing*, 56(5):1916–1930, 2008.

[48] P. Chardiere and A. Sutter. A decomposition method for quadratic zero-one programming. *Management Science*, 41(4):704–712, 1995.

[49] A. Chechetka and C. Guestrin. Efficient principled learning of thin junction trees. In *Neural Information Processing Systems*, 2007.

[50] C. Chekuri, S. Khanna, J. Naor, and L. Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 109–118, 2001.

[51] L. Chen, M.J. Wainwright, M. Cetin, and A.S. Willsky. Data association based on optimization in graphical models with application to sensor networks. *Mathematical and Computer Modelling*, 43:1114–1135, 2006.

[52] N.N. Chentsov. A systematic theory of exponential families of probability distributions. *Theory of Probability and its Applications*, 11:425–425, 1966.

[53] N.N. Chentsov. *Statistical Decision Rules and Optimal Inference.* American Mathematical Society, 1982.

[54] M. Choi, V. Chandrasekaran, and A. Willsky. Ieee international conference on acoustics, speech and signal processing (icassp). April 2008.

[55] M.J. Choi, V. Chandrasekaran, D.M. Malioutov, J.K. Johnson, and A.S. Willsky. Multiscale stochastic modeling for tractable inference and data assimilation. *Computer Methods in Applied Mechanics and Engineering*, 2008.

[56] C. Chow and C. Liu. Approximating discrete probability distributions. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[57] W. Cook and A. Rohe. Computing minimum-weight perfect matchings. *INFORMS Journal of Computing*, 11(2), 1999.

[58] G.F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

[59] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* John Wiley and Sons, Inc., 2nd edition, 2006.

[60] R.G. Cowell, A.P. Dawid, S.L. Lauritzen, and D.J. Spiegelhalter. *Probabilistic Networks and Expert Systems.* Springer-Verlag, 1999.

[61] I. Csiszár. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3:146–158, 1975.

[62] I. Csiszár and P.C. Shields. *Information Theory and Statistics: A Tutorial.* Foundations and Trends in Communication and Information Theory. NOW Publisher, Inc., 2004.

[63] I. Csiszár and G. Tusnády. Information geometry and alternating minimization procedures. *Statistics and Decisions*, pages 205–237, 1984.

[64] J. Dahl, L. Vandenberghe, and V. Roychowdhury. Covariance selection for nonchordal graphs via chordal embedding. 2007.

[65] K. Daoudi, A.B. Frakt, and A.S. Willsky. Multiscale autoregressive models and wavelets. *IEEE Trans. Information Theory*, 45(3):828–845, April 1999.

[66] V. Delouille, R. Neelamani, and R. Baraniuk. Robust distributed estimation in sensor networks using the embedded polygons algorithm. In *Proceedings of the 3rd international symposium on information processing in sensor networks*, pages 405–413, 2004.

[67] A.P. Dempster. Covariance selection. *Biometrics*, 28:157–175, 1972.

[68] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[69] A. Deshpande, M.N. Garofalakis, and M.I. Jordan. Efficient stepwise selection in decomposable models. In *Proceedings of Conference on Uncertainty in Artificial Intelligence*, pages 128–135, 2001.

[70] M. Diasparra and H. Gzyl. Entropic approach to interior point solution of linear programs. *Applied Mathematics and Computation*, 10:339–347, November 2003.

[71] R.L. Dobrushin. Prescribing a system of random variables by conditional distribution. *Theory of Probability and its Applications*, 15:458–486, 1970.

[72] M. Dudik, S.J. Phillips, and R.E. Schapire. Maximum entropy density estimation with generalized regularization and an application to species distribution modeling. *The Journal of Machine Learning Research*, 8:1217–1260, 2007.

[73] J. Edmonds. Paths, trees and flowers. *Canadian Jounal of Mathematics*, 17:449–467, 1965.

[74] B. Efron. The geometry of exponential families. *The Annals of Statistics*, 6(2):362–376, 1978.

[75] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the Americal Statistical Association*, 96(456), December 2001.

[76] J. Feldman, M.J. Wainwright, and D.R. Karger. Using linear programming to decode binary linear codes. *IEEE Transactions on Information Theory*, 51:954–972, 2005.

[77] W. Fenchel. On conjugate convex functions. *Canadian Journal of Mathematics*, 1:73–77, 1949.

[78] W. Fenchel. Convex cones, sets, and functions, 1951. unpublished notes.

[79] M.E. Fisher. On the Dimer solution of planar Ising models. *Journal of Mathematical Physics*, 7(10):1776–1781, 1966.

[80] M.L. Fisher. The Lagrangian relaxation method for solving integer programming problems. *Management Science*, 27(1), 1981.

[81] R.A. Fisher. *Statistical Methods and Scientific Inference.* Hafner, 3rd edition, 1973.

[82] L.R. Ford and D.R. Fulkerson. *Flows in Networks.* Princeton University Press, 1962.

[83] G.D. Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, March 1973.

[84] T. Frese, C.A. Bouman, and K. Sauer. Multiscale Bayesian methods for discrete tomograpy. In G.T. Herman and A. Kuba, editors, *Discrete Tomograpy: Foundations, Algorithms, and Applications.* Birkhäuser, 1999.

[85] B.J. Frey. *Graphical Models for Machine Learning and Digital Communication.* MIT Press, 1998.

[86] R.G. Gallagher. *Information Theory and Reliable Communication.* John Wiley, 1968.

[87] A. Galluccio, M. Loebl, and J. Vondrák. New algorithm for the Ising problem: partition function for finite lattice graphs. *Physics Review Letters*, 84:5924–5927, 2000.

[88] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. In M.A. Fischler and O. Firschein, editors, *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, pages 564–584. Morgan Kaufmann, 1987.

[89] A.M. Geoffrion. Lagrangean relaxation for integer programming problems. In *Mathematical Programming Study 2: Approaches to Integer Programming*, pages 81–114. North-Holland Publishing Company, 1974.

[90] J.W. Gibbs. *Elementary Principles in Statistical Mechanics.* Longmans Green and Company, 1928.

[91] B. Gidas. A renormalization group approach to image processing problems. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11, February 1989.

[92] A. Globerson and T. Jaakkola. Approximate inference using planar graph decomposition. In *Advances in Neural Information Processing Systems*, volume 20, 2006.

[93] A. Globerson and T. Jaakkola. Fixing max-product: convergent message passing algorithms for MAP-LP relaxations. In *Advances in Neural Information Processing Systems*, volume 21, 2007.

[94] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, November 1995.

[95] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):377–388, April 1996.

[96] G.H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, 3rd edition, 1996.

[97] I.J. Good. Maximum entropy for hypothesis formation, especially for multidimensional contingency tables. *Annals of Mathematical Statistics*, 34:911–934, 1963.

[98] D.M. Greig, B.T. Porteous, and A.H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society B*, 51(2):271–279, 1989.

[99] G.R. Grimmett. A theorem about random fields. *Bulletin of the London Mathematical Society*, 5:81–84, 1973.

[100] M. Guignard and S. Kim. Lagrangean decomposition: A model yielding stronger Lagrangean bounds. *Mathematical Programming*, 39:215–228, 1987.

[101] X. Guyon. *Random Fields on a Network: Modeling, Statistics and Applications.* Springer-Verlag, 1995.

[102] F. Hadlock. Finding a maximum cut of a planar graph in polynomial time. *SIAM Journal of Computing*, 4(3):221–225, September 1975.

[103] P.L. Hammer, P. Hansen, and B. Simone. Roof duality, complementation and persistency in quadratic 0-1 optimization. *Mathematical Programming*, 28:121–155, 1984.

[104] J.M. Hammersley and P.E. Clifford. *Markov fields on finite graphs and lattices.* 1971. unpublished manuscript.

[105] F. Harary. *Graph Theory.* Addison-Wesley, 1969.

[106] D. Heckerman. A tutorial on learning with Bayesian networks. Technical Report Technical Report MSR-TR-95-06, Microsoft Research, Redmond, WA, 1995. Revised June, 1996.

[107] T. Heskes. Stable fixed points of loopy belief propagation are minima of the Bethe free energy. In *Advances in Neural Information Processing Systems*, volume 15, 2003.

[108] T. Heskes. On the uniqueness of loopy belief propagation fixed-points. *Neural Computation*, 16:2379–2413, 2004.

[109] T. Heskes. Convexity arguments for efficient minimization of the Bethe and Kikuchi free energies. *Journal of Artificial Intelligence Research*, 2006.

[110] R.A. Horn and C.R. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[111] J. Idier. Convex half-quadratic criteria and interacting auxiliary variables for image restoration. *IEEE Trans. Image Processing*, 10(7):1001–1009, July 2001.

[112] J. Idier and Y. Goussard. Mulitchannel seismic deconvolution. *IEEE Transactions on Geoscience and Remote Sensing*, 31(5):961–979, September 1993.

[113] A.T. Ihler, J.W. Fischer III, and A.S. Willsky. Loopy belief propagation: Convergence and effects of message errors. *The Journal of Machine Learning Research*, 6:905–936, December 2005.

[114] C.T. Ireland and S. Kullback. Contingency tables with given marginals. *Biometrika*, 55:179–188, 1968.

[115] H. Ishikawa. Exact optimization for Markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, 2003.

[116] R.B. Israel. *Convexity in the Theory of Lattice Gases.* Princeton University Press, 1979.

[117] E.T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106:620–630, 1957.

[118] J.K. Johnson. Estimation of GMRFs by recursive cavity modeling. Master's thesis, Massachusetts Institute of Technology, March 2003.

[119] J.K. Johnson. Lagrangian-relaxed max-product approach to multiple hypothesis tracking. Stochastic Systems Group Seminar, 2004. Research performed at Alphatech, Inc.

[120] J.K. Johnson. Fisher information in Gaussian graphical models. Available at `http://ssg.mit.edu/group/jasonj`, September 2006.

[121] J.K. Johnson. On Möbius transforms and Boltzmann machines. Available at `http://ssg.mit.edu/group/jasonj`, August 2006.

[122] J.K. Johnson. Walk-summable Gauss-Markov random fields. Available at `http://ssg.mit.edu/group/jasonj`, August 2006.

[123] J.K. Johnson, V. Chandrasekaran, and A.S. Willsky. Learning Markov structure by maximum entropy relaxation. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, March 2007.

[124] J.K. Johnson, D.M. Malioutov, and A.S. Willsky. Walk-sum interpretation and analysis of Gaussian belief propagation. In *Advances in Neural Information Processing Systems*, volume 18, pages 579–586, December 2005.

[125] J.K. Johnson, D.M. Malioutov, and A.S. Willsky. Lagrangian relaxation for MAP estimation in graphical models. In *Proceedings of the 45th Annual Allerton Conference on Communication, Control and Computing*, September 2007.

[126] J.K. Johnson and A.S. Willsky. A recursive model-reduction method for estimation in Gaussian Markov random fields. *IEEE Transactions on Information Theory*, 17(1):70–83, 2008.

[127] M.I. Jordan, editor. *Learning in Graphical Models.* MIT Press, 1999.

[128] M. Kac and J.C. Ward. A combinatorial solution of the two-dimensional Ising model. *Physical Review*, 88(6):1332–1337, 1952.

[129] L.P. Kadanoff. *Statistical Physics: Statics, Dynamics and Renormalization.* World Scientific, 2000.

[130] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME — Journal of Basic Engineering*, 82:35–45, March 1961.

[131] D. Karger and N. Srebro. Learning Markov networks: Maximum bounded-treewidth graphs. In *Symposium on Discrete Algorithms*, pages 392–401, 2001.

[132] P.W. Kasteleyn. Dimer statistics and phase transitions. *Journal of Mathematical Physics*, 4:287–293, 1963.

[133] J. Kleinberg and Éva Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. *Journal of the ACM*, 49:616–639, 2002.

[134] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1568–1583, 2006.

[135] V. Kolmogorov and M. Wainwright. On the optimality of tree-reweighted max-product message passing. In *Uncertainty in Artificial Intelligence*, 2005.

[136] V. Kolmogorov and R. Zabih. What energy function can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[137] N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *IEEE 11th International Conference on Computer Vision*, pages 14–21, October 2007.

[138] N. Komodakis and G. Tziritas. Approximate labeling via graphs cuts based on linear programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1436–1453, 2007.

[139] J.J. Kosowsky and A.L. Yuille. The invisible hand algorithm: solving the assignment problem with statistical mechanics. *Neural Networks*, 7(3):477–490, 1994.

[140] V.A. Kovalevsky and V.K. Koval. A diffusion algorithm for decreasing energy of max-sum labeling problem. Technical report, Glushkov Institute of Cybernetics, Kiev, USSR, 1975.

[141] J.B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. In *Proceedings of the American Mathematical Society*, volume 7, February 1956.

[142] S. Kullback. *Information Theory and Statistics.* John Wiley and Sons, Inc., 1959.

[143] S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, March 1951.

[144] S.L. Lauritzen. Propagation of probabilities, means and variances in mixed graphical association models. *Journal of the American Statistical Association*, 87(420):1098–1108, 1992.

[145] S.L. Lauritzen. *Graphical Models.* Oxford University Press, 1996.

[146] S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society B*, 50:157–224, 1988.

[147] S. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of Markov networks using L1-regularization. In *Neural Information Processing Systems*, 2006.

[148] T. Lengauer. *Combinatorial algorithms for integrated circuit layout.* John Wiley & Sons, Inc., 1990.

[149] Stan Z. Li. *Markov Random Field Modeling in Computer Vision.* Springer-Verlag, 2nd edition, 2001.

[150] X. Li and S. Fang. On the entropic regularization method for solving min-max problems with applications. *Mathematical methods of operations research*, 46:119–130, 1997.

[151] A. Linhares, H.H. Yanasse, and J.R.A. Torreao. Linear gate assignment: A fast statistical mechanics approach. *IEEE Transactions on computer-aided design of integrated circuits and systems*, 18(12):1750–1758, December 1999.

[152] R.J. Lipton and R.E. Tarjan. Applications of a planar separator theorem. *SIAM Journal of Computing*, 9:615–627, 1980.

[153] D.J.C. MacKay. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003.

[154] D. Malioutov. *Inference in Gaussian Graphical Models.* PhD thesis, MIT, July 2008.

[155] D.M. Malioutov, J.K. Johnson, and A.S. Willsky. GMRF variance approximation using spliced wavelet bases. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 1101–1104, April 2006.

[156] D.M. Malioutov, J.K. Johnson, and A.S. Willsky. Low-rank variance estimation in large-scale GMRF models. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 676–679, May 2006.

[157] D.M. Malioutov, J.K. Johnson, and A.S. Willsky. Walk-sums and belief propagation in Gaussian graphical models. *The Journal of Machine Learning Research*, 7:2003–2030, 2006.

[158] M. Mézard and A. Montanari. *Constraint Satisfaction Networks in Physics and Computation (Draft)*. Clarendon Press, 2006.

[159] P. Michelon and N. Maculan. Lagrangean decomposition for integer and nonlinear programming with linear constraints. *Mathematical Programming*, 52:303–313, 1991.

[160] C. Moallemi and B. Van Roy. Convergence of the min-sum algorithm for quadratic programming. Technical report, Stanford University, March 2006. (Revised, May 2008).

[161] C. Moallemi and B. Van Roy. Convergence of the min-sum algorithm for convex optimization. In *Proceedings of the 45th Allerton Conference on Communication, Control and Computing*, September 2007.

[162] C.C. Moallemi and B. Van Roy. Consensus propagation. *IEEE Transactions on Information Theory*, 52(11):4753–4766, 2006.

[163] J.M. Mooij and H.J. Kappen. Sufficient conditions for convergence of the sum-product algorithm. *IEEE Transactions of Information Theory*, 53(12):4422–4437, December 2007.

[164] M. Mucha and P. Sankowski. Maximum matchings via Gaussian elimination. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science*, pages 248–255, October 2004.

[165] M. Mucha and P. Sankowski. Maximum matchings in planar graphs via Gaussian elimination. *Algorithmica*, 2006.

[166] M.K. Murray and J.W. Rice. *Differential Geometry and Statistics*. Chapman and Hall, 1993.

[167] M. Narasimhan and J. Bilmes. PAC-learning bounded tree-width graphical models. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 410–417, 2004.

[168] M. Nikolova and M. Ng. Analysis of half-quadratic minimization methods for signal and image recovery. *SIAM Journal of Scientific Computing*, 27:937–956, 2005.

[169] M. Opper and D. Saad, editors. *Advanced Mean Field Methods*. MIT Press, 2001.

[170] M. Padberg. The Boolean quadric polytope: Some characteristics, facets and relatives. *Mathematical Programming*, 45:139–172, 1989.

[171] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, 1982.

[172] G. Pardella and F. Liers. Exact ground states of huge planar Ising spin glasses. Available online at `http://arxiv.org` [arXiv:0801.3143v2], January 2008.

[173] G. Parisi. *Statistical field theory.* Addison-Wesley, 1988.

[174] Y. Pawitan. *In All Likelihood: Statistical Modelling and Inference Using Likelihood.* Oxford University Press, 2001.

[175] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[176] D.L. Phlam, C. Xu, and J.L. Prince. A survey of current methods in medical image segmentation. In M.L. Yarmush K.R. Diller and M. Toner, editors, *Annual Review of Biomedical Engineering*, volume 2, pages 315–337. 2000.

[177] S.D. Della Pietra, V.J. Della Pietra, and J.D. Lafferty. Inducing feature on random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

[178] A. Rangarajan, A. Yuille, and E. Mjolsness. Convergence properties of the softassign quadratic assignment algorithm. *Neural Computation*, 11(6):1455–1474, August 1999.

[179] T. Richardson and R. Urbanke. *Modern Coding Theory.* Cambridge University Press, 2007.

[180] R.T. Rockafellar. *Convex Analysis.* Princeton University Press, 1970.

[181] R.T. Rockafellar. *Conjugate Duality and Optimization.* SIAM, 1974.

[182] D.J. Rose. A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. In R.C. Reed, editor, *Graph Theory and Computing*, pages 183–217. Academic Press, 1972.

[183] D.J. Rose, R.E. Tarjan, and G.S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal of Computing*, 5:266–283, 1976.

[184] W. Rudin. *Principles of Mathematical Analysis.* McGraw-Hill, Inc., 1964.

[185] H. Rue and L. Held. *Gaussian Markov Random Fields.* Chapman and Hall, 2005.

[186] L. Rüschendorf. Convergence of the iterative proportional fitting procedure. *The Annals of Statistics*, 23(4):1160–1174, 1995.

[187] S. Sanghavi, D. Malioutov, and A. Willsky. Linear programming analysis of loopy belief propagation for weighted matching. In *Advances in Neural Information Processing Systems*, 2007.

[188] S. Sanghavi, D. Shah, and A. Willsky. Message passing for max-weight independent set. In *Advances in Neural Information Processing Systems*, 2007.

[189] J. Schiff, D. Antonelli, A.G. Dimakis, D. Chu, and M.J. Wainwright. Robust message-passing for statistical inference. In *Proceedings of the 6th International symposium on information processing in sensor networks*, pages 109–118, 2007.

[190] I.D. Schizas, A. Ribeiro, and G.B. Giannakis. Consensus in ad hoc WSNs with noisy links—Part I: Distributed estimation of deterministic signals. *IEEE Transactions on Signal Processing*, 56(1):350–364, January 2008.

[191] M.I. Schlesinger. False minima of the algorithm for minimizing energy of max-sum labeling problem. Technical report, Glushkov Institute of Cybernetics, Kiev, USSR, 1976.

[192] M.I. Schlesinger. Syntactic analysis of two-dimensional visual signals in noisy conditions (in Russian). *Kibernetika*, 4:113–130, 1976.

[193] C.E. Shannon and W. Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, 1948.

[194] O. Shental, A.J. Weiss, N. Shental, and Y. Weiss. Generalized belief propagation receiver for near-optimal detection of two-dimensional channels with memory. In *IEEE Information Theory Workshop*, pages 225–229, 2004.

[195] B. Simon. *The Statistical Mechanics of Lattice Gases*, volume I. Princeton University Press, 1993.

[196] C. Simone, M. Diehl, M. Jünger, P. Mutzel, G. Reinelt, and G. Rinaldi. Exact ground states of two-dimensional $\pm J$ Ising spin glasses. *Journal of Statistical Physics*, 84:1363–1371, 1996.

[197] A.H.S. Solberg, T. Taxt, and A.K. Jain. A Markov random field model for classification of multisource satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 34(1):100–113, January 1996.

[198] D. Sontag and T. Jaakkola. New outer bounds on the marginal polytope. In *Advances in Neural Information Processing Systems*, volume 21, 2007.

[199] T.P. Speed and H.T. Kiiveri. Gaussian Markov distributions over finite graphs. *The Annals of Statistics*, 14(1):138–150, 1986.

[200] G. Storvik and G. Dahl. Lagrangian-based methods for finding MAP solutions for MRF models. *IEEE Transactions on Image Processing*, 9(3):469–479, 2000.

[201] E.B. Sudderth, M.J. Wainwright, and A.S. Willsky. Embedded trees: estimation of Gaussian processes on graphs with cycles. *IEEE Transactions on Signal Processing*, 52(11), November 2004.

[202] S.C. Tatikonda and M.I. Jordan. Loopy belief propagation and gibbs measures. In *Advances in Neural Information Processing Systems*, volume 18, pages 493–500, August 2002.

[203] C.K. Thomas and A.A. Middleton. Matching Kasteleyn cities for spin glass ground states. *Physical Review B*, 76, 2007.

[204] G. Toulouse. Theory of the frustration effect in spin glasses I. *Communications in Physics*, 2:115–119, 1977.

[205] U. Trottenberg, C. Oosterlee, and A. Schuller. *Multigrid.* Academic Press, 2001.

[206] R.S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, 1962.

[207] A.J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269, April 1967.

[208] P.O. Vontobel. Interior-point algorithms for linear-programming decoding. In *Proceedings of the Information Theory and Applications Workshop*, January 2008.

[209] P.O. Vontobel and R. Koetter. On low-complexity linear-programming decoding of LDPC codes. *European Transactions on Telecommunications*, 18:509–517, 2007.

[210] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky. Tree-based reparameterization framwork for analysis of sum-product and related algorithms. *IEEE Transactions on Information Theory*, 49(5):1120–1146, May 2003.

[211] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky. MAP estimation via agreement on trees: Message-passing and linear programming. *IEEE Transactions of Information Theory*, 51:3697–3717, 2005.

[212] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky. A new class of upper bound on the log partition function. *IEEE Transactions on Information Theory*, 51(7):2313–2335, 2005.

[213] M.J. Wainwright and M.I. Jordan. Graphical models, exponential families and variational inference. Technical Report 649, UC Berkeley, Department of Statistics, 2003.

[214] M.J. Wainwright, P. Ravikumar, and J. Lafferty. High-dimensional graphical model selection using ℓ_1-regularized logistic regression. In *Neural Information Processing Systems*, 2006.

[215] S. Wang, R. Rosenfeld, Y. Zhao, and D. Schuurmans. The latent maximum entropy principle. In *IEEE International Symposium on Information Theory*, 2002.

[216] L.J. Watters. Reduction of integer polynomial programs to zero-one linear programming problems. *Operations Research*, 15:1171–1174, 1967.

[217] Y. Weiss and W.T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, 13:2173–2200, 2001.

[218] Y. Weiss and W.T. Freeman. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):736–744, 2001.

[219] Y. Weiss, C. Yanover, and T. Meltzer. MAP estimation, linear programming and belief propagation with convex free energies. In *Uncertainty in Artificial Intelligence*, 2007.

[220] T. Werner. A linear programming approach to max-sum problem: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:1165–1179, 2007.

[221] W. Wiegerinck and T. Heskes. Fractional belief propagation. In *Advances in Neural Information Processing Systems*, volume 15, 2003.

[222] A.S. Willsky. Multiresolution Markov models for signal and image processing. *Proceedings of the IEEE*, 90(8):1396–1458, 2002.

[223] J. Woods. Markov image modeling. *IEEE Transactions on Automatic Control*, 23(5):846–850, October 1978.

[224] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the 4th international symposium on information processing in sensor networks*, 2005.

[225] M. Yannakis. Computing the minimum fill-in is NP-complete. *SIAM Journal of Algebraic Discrete Methods*, 2:77–79, 1981.

[226] C. Yanover, T. Meltzer, and Y. Weiss. Linear programming relaxations and belief propagation – An empirical study. *The Journal of Machine Learning Research*, 7:1887–1907, 2006.

[227] J.S. Yedidia, W.T. Freeman, and Y.W. Weiss. Bethe free energy, Kikuchi approximations, and belief propagation algorithms. Technical Report TR-2001-16, Mitsubishi Electric Research Laboratories, May 2001.

[228] J.S. Yedidia, W.T. Freeman, and Y.W. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312, July 2005.

[229] J.M. Yeomans. *Statistical Mechanics of Phase Transitions*. Oxford University Press, 1992.

[230] F. Yu, F. Tu, H. Tu, and K. Pattipati. A Lagrangian relaxation algorithm for finding the MAP configuration in QMR-DT. *IEEE Transactions on Systems, Man and Cybernetics — Part A: Systems and Humans*, 37(5):746–756, September 2007.

[231] A.L. Yuille. CCCP algorithm to minimize the Bethe and Kikuchi free energies: convergent alternatives to belief propagation. *Neural Computation*, 14:1691–1722, 2002.