now

the essence of knowledge

Graphical Models, Exponential Families, and Variational Inference

Martin J. Wainwright[1] and Michael I. Jordan[2]

[1] *Department of Statistics, and Department of Electrical Engineering and Computer Science, University of California, Berkeley 94720, USA, wainwrig@stat.berkeley.edu*
[2] *Department of Statistics, and Department of Electrical Engineering and Computer Science, University of California, Berkeley 94720, USA, jordan@stat.berkeley.edu*

Abstract

The formalism of probabilistic graphical models provides a unifying framework for capturing complex dependencies among random variables, and building large-scale multivariate statistical models. Graphical models have become a focus of research in many statistical, computational and mathematical fields, including bioinformatics, communication theory, statistical physics, combinatorial optimization, signal and image processing, information retrieval and statistical machine learning. Many problems that arise in specific instances — including the key problems of computing marginals and modes of probability distributions — are best studied in the general setting. Working with exponential family representations, and exploiting the conjugate duality between the cumulant function and the entropy for exponential families, we develop general variational representations of the problems of computing likelihoods, marginal probabilities and most probable configurations. We describe how a wide variety

of algorithms — among them sum-product, cluster variational methods, expectation-propagation, mean field methods, max-product and linear programming relaxation, as well as conic programming relaxations — can all be understood in terms of exact or approximate forms of these variational representations. The variational approach provides a complementary alternative to Markov chain Monte Carlo as a general source of approximation methods for inference in large-scale statistical models.

1

Introduction

Graphical models bring together graph theory and probability theory in a powerful formalism for multivariate statistical modeling. In various applied fields including bioinformatics, speech processing, image processing and control theory, statistical models have long been formulated in terms of graphs, and algorithms for computing basic statistical quantities such as likelihoods and score functions have often been expressed in terms of recursions operating on these graphs; examples include phylogenies, pedigrees, hidden Markov models, Markov random fields, and Kalman filters. These ideas can be understood, unified, and generalized within the formalism of graphical models. Indeed, graphical models provide a natural tool for formulating variations on these classical architectures, as well as for exploring entirely new families of statistical models. Accordingly, in fields that involve the study of large numbers of interacting variables, graphical models are increasingly in evidence.

Graph theory plays an important role in many computationally oriented fields, including combinatorial optimization, statistical physics, and economics. Beyond its use as a language for formulating models, graph theory also plays a fundamental role in assessing computational

complexity and feasibility. In particular, the running time of an algorithm or the magnitude of an error bound can often be characterized in terms of structural properties of a graph. This statement is also true in the context of graphical models. Indeed, as we discuss, the computational complexity of a fundamental method known as the *junction tree algorithm* — which generalizes many of the recursive algorithms on graphs cited above — can be characterized in terms of a natural graph-theoretic measure of interaction among variables. For suitably sparse graphs, the junction tree algorithm provides a systematic solution to the problem of computing likelihoods and other statistical quantities associated with a graphical model.

Unfortunately, many graphical models of practical interest are not "suitably sparse," so that the junction tree algorithm no longer provides a viable computational framework. One popular source of methods for attempting to cope with such cases is the *Markov chain Monte Carlo* (MCMC) framework, and indeed there is a significant literature on the application of MCMC methods to graphical models [e.g., 28, 93, 202]. Our focus in this survey is rather different: we present an alternative computational methodology for statistical inference that is based on *variational methods*. These techniques provide a general class of alternatives to MCMC, and have applications outside of the graphical model framework. As we will see, however, they are particularly natural in their application to graphical models, due to their relationships with the structural properties of graphs.

The phrase "variational" itself is an umbrella term that refers to various mathematical tools for optimization-based formulations of problems, as well as associated techniques for their solution. The general idea is to express a quantity of interest as the solution of an optimization problem. The optimization problem can then be "relaxed" in various ways, either by approximating the function to be optimized or by approximating the set over which the optimization takes place. Such relaxations, in turn, provide a means of approximating the original quantity of interest.

The roots of both MCMC methods and variational methods lie in statistical physics. Indeed, the successful deployment of MCMC methods in statistical physics motivated and predated their entry into

statistics. However, the development of MCMC methodology specifically designed for statistical problems has played an important role in sparking widespread application of such methods in statistics [88]. A similar development in the case of variational methodology would be of significant interest. In our view, the most promising avenue toward a variational methodology tuned to statistics is to build on existing links between variational analysis and the exponential family of distributions [4, 11, 43, 74]. Indeed, the notions of convexity that lie at the heart of the statistical theory of the exponential family have immediate implications for the design of variational relaxations. Moreover, these variational relaxations have particularly interesting algorithmic consequences in the setting of graphical models, where they again lead to recursions on graphs.

Thus, we present a story with three interrelated themes. We begin in Section 2 with a discussion of graphical models, providing both an overview of the general mathematical framework, and also presenting several specific examples. All of these examples, as well as the majority of current applications of graphical models, involve distributions in the exponential family. Accordingly, Section 3 is devoted to a discussion of exponential families, focusing on the mathematical links to convex analysis, and thus anticipating our development of variational methods. In particular, the principal object of interest in our exposition is a certain *conjugate dual* relation associated with exponential families. From this foundation of conjugate duality, we develop a general variational representation for computing likelihoods and marginal probabilities in exponential families. Subsequent sections are devoted to the exploration of various instantiations of this variational principle, both in exact and approximate forms, which in turn yield various algorithms for computing exact and approximate marginal probabilities, respectively. In Section 4, we discuss the connection between the Bethe approximation and the sum-product algorithm, including both its exact form for trees and approximate form for graphs with cycles. We also develop the connections between Bethe-like approximations and other algorithms, including generalized sum-product, expectation-propagation and related moment-matching methods. In Section 5, we discuss the class of mean field methods, which arise from a qualitatively

different approximation to the exact variational principle, with the added benefit of generating lower bounds on the likelihood. In Section 6, we discuss the role of variational methods in parameter estimation, including both the fully observed and partially observed cases, as well as both frequentist and Bayesian settings. Both Bethe-type and mean field methods are based on nonconvex optimization problems, which typically have multiple solutions. In contrast, Section 7 discusses variational methods based on convex relaxations of the exact variational principle, many of which are also guaranteed to yield upper bounds on the log likelihood. Section 8 is devoted to the problem of mode computation, with particular emphasis on the case of discrete random variables, in which context computing the mode requires solving an integer programming problem. We develop connections between (reweighted) max-product algorithms and hierarchies of linear programming relaxations. In Section 9, we discuss the broader class of conic programming relaxations, and show how they can be understood in terms of semidefinite constraints imposed via moment matrices. We conclude with a discussion in Section 10.

The scope of this survey is limited in the following sense: given a distribution represented as a graphical model, we are concerned with the problem of computing marginal probabilities (including likelihoods), as well as the problem of computing modes. We refer to such computational tasks as problems of "probabilistic inference," or "inference" for short. As with presentations of MCMC methods, such a limited focus may appear to aim most directly at applications in Bayesian statistics. While Bayesian statistics is indeed a natural terrain for deploying many of the methods that we present here, we see these methods as having applications throughout statistics, within both the frequentist and Bayesian paradigms, and we indicate some of these applications at various junctures in the survey.

2

Background

We begin with background on graphical models. The key idea is that of *factorization*: a graphical model consists of a collection of probability distributions that factorize according to the structure of an underlying graph. Here, we are using the terminology "distribution" loosely; our notation p should be understood as a mass function (density with respect to counting measure) in the discrete case, and a density with respect to Lebesgue measure in the continuous case. Our focus in this section is the interplay between probabilistic notions such as conditional independence on one hand, and on the other hand, graph-theoretic notions such as cliques and separation.

2.1 Probability Distributions on Graphs

We begin by describing the various types of graphical formalisms that are useful. A graph $G = (V, E)$ is formed by a collection of vertices $V = \{1, 2, \dots, m\}$, and a collection of edges $E \subset V \times V$. Each edge consists of a pair of vertices $s, t \in E$, and may either be *undirected*, in which case there is no distinction between edge (s, t) and edge (t, s), or *directed*, in which case we write $(s \rightarrow t)$ to indicate the direction. See Appendix A.1 for more background on graphs and their properties.

In order to define a graphical model, we associate with each vertex $s \in V$ a random variable X_s taking values in some space $\mathcal{X}_s$. Depending on the application, this state space $\mathcal{X}_s$ may either be continuous, (e.g., $\mathcal{X}_s = \mathbb{R}$) or discrete (e.g., $\mathcal{X}_s = \{0, 1, \ldots, r-1\}$). We use lower-case letters (e.g., $x_s \in \mathcal{X}_s$) to denote particular elements of $\mathcal{X}_s$, so that the notation $\{X_s = x_s\}$ corresponds to the event that the random variable X_s takes the value $x_s \in \mathcal{X}_s$. For any subset A of the vertex set V, we define the subvector $X_A := (X_s, \ s \in A)$, with the notation $x_A := (x_s, \ s \in A)$ corresponding to the analogous quantity for values of the random vector X_A. Similarly, we define $\otimes_{s \in A} \mathcal{X}_s$ to be the Cartesian product of the state spaces for each of the elements of X_A.

2.1.1   Directed Graphical Models

Given a directed graph with edges $(s \to t)$, we say that t is a child of s, or conversely, that s is a parent of t. For any vertex $s \in V$, let $\pi(s)$ denote the set of all parents of given node $s \in V$. (If a vertex s has no parents, then the set $\pi(s)$ should be understood to be empty.) A directed cycle is a sequence $(s_1, s_2, \ldots, s_k)$ such that $(s_i \to s_{i+1}) \in E$ for all $i = 1, \ldots, k-1$, and $(s_k \to s_1) \in E$. See Figure 2.1 for an illustration of these concepts.

Now suppose that G is a directed acyclic graph (DAG), meaning that every edge is directed, and that the graph contains no directed
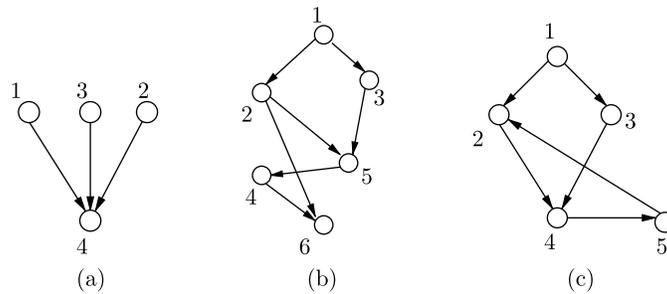

Fig. 2.1 (a) A simple directed graphical model with four variables (X_1, X_2, X_3, X_4). Vertices $\{1, 2, 3\}$ are all parents of vertex 4, written $\pi(4) = \{1, 2, 3\}$. (b) A more complicated directed acyclic graph (DAG) that defines a partial order on its vertices. Note that vertex 6 is a child of vertex 2, and vertex 1 is an ancestor of 6. (c) A forbidden directed graph (nonacyclic) that includes the directed cycle $(2 \to 4 \to 5 \to 2)$.

cycles. For any such DAG, we can define a partial order on the vertex set V by the notion of ancestry: we say that node s is an ancestor of u if there is a directed path $(s, t_1, t_2, \ldots, t_k, u)$ (see Figure 2.1(b)). Given a DAG, for each vertex s and its parent set $\pi(s)$, let $p_s(x_s \mid x_{\pi(s)})$ denote a nonnegative function over the variables $(x_s, x_{\pi(s)})$, normalized such that $\int p_s(x_s \mid x_{\pi(s)}) dx_s = 1$. In terms of these local functions, a *directed graphical model* consists of a collection of probability distributions (densities or mass functions) that factorize in the following way:

$$p(x_1, x_2, \ldots, x_m) = \prod_{s \in V} p_s(x_s \mid x_{\pi(s)}). \tag{2.1}$$

It can be verified that our use of notation is consistent, in that $p_s(x_s \mid x_{\pi(s)})$ is, in fact, the conditional probability of $\{X_s = x_s\}$ given $\{X_{\pi(s)} = x_{\pi(s)}\}$ for the distribution $p(\cdot)$ defined by the factorization (2.1). This follows by an inductive argument that makes use of the normalization condition imposed on $p_s(\cdot)$, and the partial ordering induced by the ancestor relationship of the DAG.

2.1.2 Undirected Graphical Models

In the undirected case, the probability distribution factorizes according to functions defined on the *cliques* of the graph. A clique C is a fully connected subset of the vertex set V, meaning that $(s, t) \in E$ for all $s, t \in C$. Let us associate with each clique C a *compatibility function* $\psi_C : (\otimes_{s \in C} \mathcal{X}_s) \to \mathbb{R}_+$. Recall that $\otimes_{s \in C} \mathcal{X}_s$ denotes the Cartesian product of the state spaces of the random vector X_C. Consequently, the compatibility function ψ_C is a local quantity, defined only for elements x_C within the clique.

With this notation, an *undirected graphical model* — also known as a *Markov random field* (MRF), or a *Gibbs distribution* — is a collection of distributions that factorize as

$$p(x_1, x_2, \ldots, x_m) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C), \tag{2.2}$$

where Z is a constant chosen to ensure that the distribution is normalized. The set $\mathcal{C}$ is often taken to be the set of all *maximal cliques* of the graph, i.e., the set of cliques that are not properly contained

within any other clique. This condition can be imposed without loss of generality, because any representation based on nonmaximal cliques can always be converted to one based on maximal cliques by redefining the compatibility function on a maximal clique to be the product over the compatibility functions on the subsets of that clique. However, there may be computational benefits to using a nonmaximal representation — in particular, algorithms may be able to exploit specific features of the factorization special to the nonmaximal representation. For this reason, we do not necessarily restrict compatibility functions to maximal cliques only, but instead define the set $\mathcal{C}$ to contain all cliques. (Factor graphs, discussed in the following section, allow for a finer-grained specification of factorization properties.)

It is important to understand that for a general undirected graph the compatibility functions ψ_C need not have any obvious or direct relation to marginal or conditional distributions defined over the graph cliques. This property should be contrasted with the directed factorization (2.1), where the factors correspond to conditional probabilities over the child-parent sets.

2.1.3   Factor Graphs

For large graphs, the factorization properties of a graphical model, whether undirected or directed, may be difficult to visualize from the usual depictions of graphs. The formalism of *factor graphs* provides an alternative graphical representation, one which emphasizes the factorization of the distribution [143, 157].

Let F represent an index set for the set of factors defining a graphical model distribution. In the undirected case, this set indexes the collection $\mathcal{C}$ of cliques, while in the directed case F indexes the set of parent–child neighborhoods. We then consider a bipartite graph $G' = (V, F, E')$, where V is the original set of vertices, and E' is a new edge set, joining only vertices $s \in V$ to factors $a \in F$. In particular, edge $(s, a) \in E'$ if and only if x_s participates in the factor indexed by $a \in F$. See Figure 2.2(b) for an illustration.

For undirected models, the factor graph representation is of particular value when $\mathcal{C}$ consists of more than the maximal cliques. Indeed, the compatibility functions for the nonmaximal cliques do not have
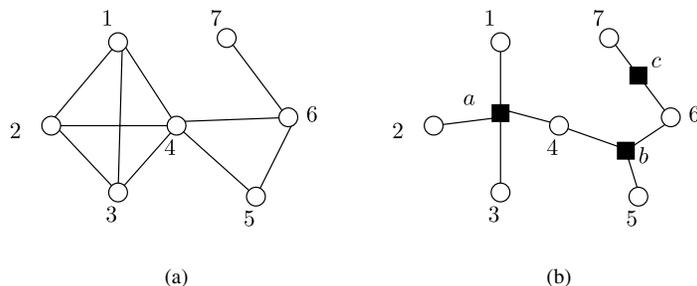
Fig. 2.2 Illustration of undirected graphical models and factor graphs. (a) An undirected graph on $m = 7$ vertices, with maximal cliques $\{1,2,3,4\}$, $\{4,5,6\}$, and $\{6,7\}$. (b) Equivalent representation of the undirected graph in (a) as a factor graph, assuming that we define compatibility functions only on the maximal cliques in (a). The factor graph is a bipartite graph with vertex set $V = \{1,\dots,7\}$ and factor set $F = \{a,b,c\}$, one for each of the compatibility functions of the original undirected graph.

an explicit representation in the usual representation of an undirected graph — however, the factor graph makes them explicit.

2.2 Conditional Independence

Families of probability distributions as defined as in Equations (2.1) or (2.2) also have a characterization in terms of conditional independencies among subsets of random variables — the *Markov properties* of the graphical model. We only touch upon this characterization here, as it is not needed in the remainder of the survey; for a full treatment, we refer the interested reader to Lauritzen [153].

For undirected graphical models, conditional independence is identified with the graph-theoretic notion of *reachability*. In particular, let A, B, and C be an arbitrary triple of mutually disjoint subsets of vertices. Let us stipulate that X_A be independent of X_B given X_C if there is no path from a vertex in A to a vertex in B when we remove the vertices C from the graph. Ranging over all possible choices of subsets A, B, and C gives rise to a list of conditional independence assertions. It can be shown that this list is always consistent (i.e., there exist probability distributions that satisfy all of these assertions); moreover, the set of probability distributions that satisfy these assertions is exactly the set of distributions defined by (2.2) ranging over all possible choices of compatibility functions.

Thus, there are two equivalent characterizations of the family of probability distributions associated with an undirected graph. This equivalence is a fundamental mathematical result, linking an algebraic concept (factorization) and a graph-theoretic concept (reachability). This result also has algorithmic consequences, in that it reduces the problem of assessing conditional independence to the problem of assessing reachability on a graph, which is readily solved using simple breadth-first search algorithms [56].

An analogous result holds in the case of directed graphical models, with the only alteration being a different notion of reachability [153]. Once again, it is possible to establish an equivalence between the set of probability distributions specified by the directed factorization (2.1), and that defined in terms of a set of conditional independence assertions.

2.3   Statistical Inference and Exact Algorithms

Given a probability distribution p defined by a graphical model, our focus will be solving one or more of the following *computational inference problems*:

(a) Computing the likelihood of observed data.
(b) Computing the marginal distribution $p(x_A)$ over a particular subset $A \subset V$ of nodes.
(c) Computing the conditional distribution $p(x_A \mid x_B)$, for disjoint subsets A and B, where $A \cup B$ is in general a proper subset of V.
(d) Computing a mode of the density (i.e., an element $\widehat{x}$ in the set $\arg\max_{x \in \mathcal{X}^m} p(x)$).

Clearly problem (a) is a special case of problem (b). The computation of a conditional probability in (c) is similar in that it also requires marginalization steps, an initial one to obtain the numerator $p(x_A, x_B)$, and a further step to obtain the denominator $p(x_B)$. In contrast, the problem of computing modes stated in (d) is fundamentally different, since it entails maximization rather than integration. Nonetheless, our variational development in the sequel will highlight some important

connections between the problem of computing marginals and that of computing modes.

To understand the challenges inherent in these inference problems, consider the case of a discrete random vector $x \in \mathcal{X}^m$, where $\mathcal{X}_s = \{0, 1, \ldots, r-1\}$ for each vertex $s \in V$. A naive approach to computing a marginal at a single node — say $p(x_s)$ — entails summing over all configurations of the form $\{x' \in \mathcal{X}^m \mid x'_s = x_s\}$. Since this set has r^{m-1} elements, it is clear that a brute force approach will rapidly become intractable. Even with binary variables ($r = 2$) and a graph with $m \approx 100$ nodes (a small size for many applications), this summation is beyond the reach of brute-force computation. Similarly, in this discrete case, computing a mode entails solving an integer programming problem over an exponential number of configurations. For continuous random vectors, the problems are no easier[1] and typically harder, since they require computing a large number of integrals.

For undirected graphs without cycles or for directed graphs in which each node has a single parent — known generically as *trees* in either case — these inference problems can be solved exactly by recursive "message-passing" algorithms of a dynamic programming nature, with a computational complexity that scales only linearly in the number of nodes. In particular, for the case of computing marginals, the dynamic programming solution takes the form of a general algorithm known as the *sum-product algorithm*, whereas for the problem of computing modes it takes the form of an analogous algorithm known as the *max-product algorithm*. We describe these algorithms in Section 2.5.1. More generally, as we discuss in Section 2.5.2, the *junction tree algorithm* provides a solution to inference problems for arbitrary graphs. The junction tree algorithm has a computational complexity that is exponential in a quantity known as the *treewidth* of the graph.

2.4 Applications

Before turning to these algorithmic issues, however, it is helpful to ground the discussion by considering some examples of applications of

---

[1] The Gaussian case is an important exception to this statement.

graphical models. We present examples of the use of graphical models in the general areas of Bayesian hierarchical modeling, contingency table analysis, and combinatorial optimization and satisfiability, as well specific examples in bioinformatics, speech and language processing, image processing, spatial statistics, communication and coding theory.

2.4.1   Hierarchical Bayesian Models

The Bayesian framework treats all model quantities — observed data, latent variables, parameters, nuisance variables — as random variables. Thus, in a graphical model representation of a Bayesian model, all such variables appear explicitly as vertices in the graph. The general computational machinery associated with graphical models applies directly to Bayesian computations of quantities such as marginal likelihoods and posterior probabilities of parameters. Although Bayesian models can be represented using either directed or undirected graphs, it is the directed formalism that is most commonly encountered in practice. In particular, in hierarchical Bayesian models, the specification of prior distributions generally involves additional parameters known as hyperparameters, and the overall model is specified as a set of conditional probabilities linking hyperparameters, parameters and data. Taking the product of such conditional probability distributions defines the joint probability; this factorization is simply a particular instance of Equation (2.1).

There are several advantages to treating a hierarchical Bayesian model as a directed graphical model. First, hierarchical models are often specified by making various assertions of conditional independence. These assertions imply other conditional independence relationships, and the reachability algorithms (mentioned in Section 2.1) provide a systematic method for investigating all such relationships. Second, the visualization provided by the graph can be useful both for understanding the model (including the basic step of verifying that the graph is acyclic), as well for exploring extensions. Finally, general computational methods such as MCMC and variational inference algorithms can be implemented for general graphical models, and hence apply to hierarchical models in graphical form. These advantages and others have

led to the development of general software programs for specifying and manipulating hierarchical Bayesian models via the directed graphical model formalism [93].

2.4.2 Contingency Table Analysis

Contingency tables are a central tool in categorical data analysis [1, 80, 151], dating back to the seminal work of Pearson, Yule, and Fisher. An m-dimensional contingency table with r levels describes a probability distribution over m random variables $(X_1, \ldots, X_m)$, each of which takes one of r possible values. For the special case $m = 2$, a contingency table with r levels is simply a $r \times r$ matrix of nonnegative elements summing to one, whereas for $m > 2$, it is a multi-dimensional array with r^m nonnegative entries summing to one. Thus, the table fully specifies a probability distribution over a random vector $(X_1, \ldots, X_m)$, where each X_s takes one of r possible values.

Contingency tables can be modeled within the framework of graphical models, and graphs provide a useful visualization of many natural questions. For instance, one central question in contingency table analysis is distinguishing different orders of interaction in the data. As a concrete example, given $m = 3$ variables, a simple question is testing whether or not the three variables are independent. From a graphical model perspective, this test amounts to distinguishing whether the associated interaction graph is completely disconnected or not (see panels (a) and (b) in Figure 2.3). A more subtle test is to distinguish whether the random variables interact in only a pairwise manner (i.e., with factors ψ_{12}, ψ_{13}, and ψ_{23} in Equation (2.2)), or if there is actually a three-way interaction (i.e., a term ψ_{123} in the factorization (2.2)). Interestingly, these two factorization choices cannot be distinguished

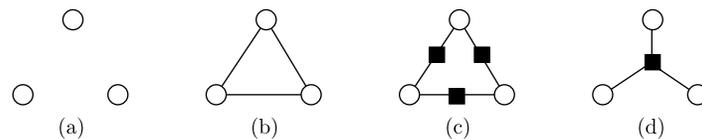(a)   (b)   (c)   (d)

Fig. 2.3 Some simple graphical interactions in contingency table analysis. (a) Independence model. (b) General dependent model. (c) Pairwise interactions only. (d) Triplet interactions.

using a standard undirected graph; as illustrated in Figure 2.3(b), the fully connected graph on 3 vertices does not distinguish between pairwise interactions without triplet interactions, versus triplet interaction (possibly including pairwise interaction as well). In this sense, the factor graph formalism is more expressive, since it can distinguish between pairwise and triplet interactions, as shown in panels (c) and (d) of Figure 2.3.

2.4.3   Constraint Satisfaction and Combinatorial Optimization

Problems of constraint satisfaction and combinatorial optimization arise in a wide variety of areas, among them artificial intelligence [66, 192], communication theory [87], computational complexity theory [55], statistical image processing [89], and bioinformatics [194]. Many problems in satisfiability and combinatorial optimization [180] are defined in graph-theoretic terms, and thus are naturally recast in the graphical model formalism.

Let us illustrate by considering what is perhaps the best-known example of satisfiability, namely the 3-SAT problem [55]. It is naturally described in terms of a factor graph, and a collection of binary random variables $(X_1, X_2, \ldots, X_m) \in \{0,1\}^m$. For a given triplet $\{s,t,u\}$ of vertices, let us specify some "forbidden" configuration $(z_s, z_t, z_u) \in \{0,1\}^3$, and then define a triplet compatibility function

$$\psi_{stu}(x_s, x_t, x_u) = \begin{cases} 0 & \text{if } (x_s, x_t, x_u) = (z_s, z_t, z_u), \\ 1 & \text{otherwise.} \end{cases} \tag{2.3}$$

Each such compatibility function is referred to as a *clause* in the satisfiability literature, where such compatibility functions are typically encoded in terms of logical operations. For instance, if $(z_s, z_t, z_u) = (0,0,0)$, then the function (2.3) can be written compactly as $\psi_{stu}(x_s, x_t, x_u) = x_s \vee x_t \vee x_u$, where $\vee$ denotes the logical OR operation between two Boolean symbols.

As a graphical model, a single clause corresponds to the model in Figure 2.3(d); of greater interest are models over larger collections of binary variables, involving many clauses. One basic problem in

satisfiability is to determine whether a given set of clauses F are satisfiable, meaning that there exists some configuration $x \in \{0,1\}^m$ such that $\prod_{(s,t,u)\in F} \psi_{stu}(x_s, x_t, x_u) = 1$. In this case, the factorization

$$p(x) = \frac{1}{Z} \prod_{(s,t,u)\in F} \psi_{stu}(x_s, x_t, x_u)$$

defines the uniform distribution over the set of satisfying assignments.

When instances of 3-SAT are randomly constructed — for instance, by fixing a clause density $\alpha > 0$, and drawing $\lceil \alpha m \rceil$ clauses over m variables uniformly from all triplets — the graphs tend to have a locally "tree-like" structure; see the factor graph in Figure 2.9(b) for an illustration. One major problem is determining the value α^* at which this ensemble is conjectured to undergo a phase transition from being satisfiable (for small α, hence with few constraints) to being unsatisfiable (for sufficiently large α). The survey propagation algorithm, developed in the statistical physics community [170, 171], is a promising method for solving random instances of satisfiability problems. Survey propagation turns out to be an instance of the sum-product or belief propagation algorithm, but as applied to an alternative graphical model for satisfiability problems [41, 162].

2.4.4 Bioinformatics

Many classical models in bioinformatics are instances of graphical models, and the associated framework is often exploited in designing new models. In this section, we briefly review some instances of graphical models in both bioinformatics, both classical and recent. Sequential data play a central role in bioinformatics, and the workhorse underlying the modeling of sequential data is the *hidden Markov model* (HMM) shown in Figure 2.4(a). The HMM is in essence a dynamical version of a finite mixture model, in which observations are generated conditionally on a underlying latent ("hidden") state variable. The state variables, which are generally taken to be discrete random variables following a multinomial distribution, form a Markov chain. The graphical model in Figure 2.4(a) is also a representation of the state-space model underlying Kalman filtering and smoothing, where the state variable is a
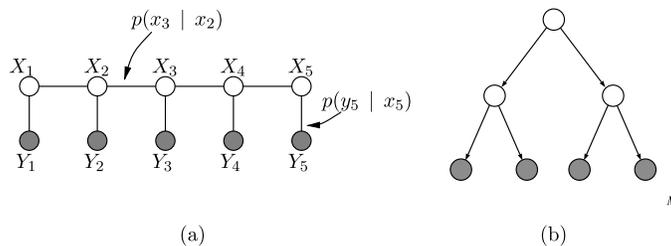
Fig. 2.4 (a) The graphical model representation of a generic hidden Markov model. The shaded nodes $\{Y_1,\ldots,Y_5\}$ are the observations and the unshaded nodes $\{X_1,\ldots,X_5\}$ are the hidden state variables. The latter form a Markov chain, in which X_s is independent of X_u conditional on X_t, where $s < t < u$. (b) The graphical model representation of a phylogeny on four extant organisms and M sites. The tree encodes the assumption that there is a first speciation event and then two further speciation events that lead to the four extant organisms. The box around the tree (a "plate") is a graphical model representation of replication, here representing the assumption that the M sites evolve independently.

Gaussian vector. These models thus also have a right to be referred to as "hidden Markov models," but the terminology is most commonly used to refer to models in which the state variables are discrete.

Applying the junction tree formalism to the HMM yields an algorithm that passes messages in both directions along the chain of state variables, and computes the marginal probabilities $p(x_t, x_{t+1}\,|\,y)$ and $p(x_t\,|\,y)$. In the HMM context, this algorithm is often referred to as the *forward–backward algorithm* [196]. These marginal probabilities are often of interest in and of themselves, but are also important in their role as expected sufficient statistics in an expectation–maximization (EM) algorithm for estimating the parameters of the HMM. Similarly, the maximum *a posteriori* state sequence can also be computed by the junction tree algorithm (with summation replaced by maximization) — in the HMM context the resulting algorithm is generally referred to as the *Viterbi algorithm* [82].

Gene-finding provides an important example of the application of the HMM [72]. To a first order of approximation, the genomic sequence of an organism can be segmented into regions containing genes and intergenic regions (that separate the genes), where a gene is defined as a sequence of nucleotides that can be further segmented into meaningful intragenic structures (exons and introns). The boundaries between these segments are highly stochastic and hence difficult to find reliably.

Hidden Markov models have been the methodology of choice for attacking this problem, with designers choosing states and state transitions to reflect biological knowledge concerning gene structure [44]. Hidden Markov models are also used to model certain aspects of protein structure. For example, membrane proteins are specific kinds of proteins that embed themselves in the membranes of cells, and play important roles in the transmission of signals in and out of the cell. These proteins loop in and out of the membrane many times, alternating between hydrophilic amino acids (which prefer the environment of the membrane) and hydrophobic amino acids (which prefer the environment inside or outside the membrane). These and other biological facts are used to design the states and state transition matrix of the *transmembrane hidden Markov model*, an HMM for modeling membrane proteins [141].

Tree-structured models also play an important role in bioinformatics and language processing. For example, phylogenetic trees can be treated as graphical models. As shown in Figure 2.4(b), a phylogenetic tree is a tree-structured graphical model in which a set of observed nucleotides (or other biological characters) are assumed to have evolved from an underlying set of ancestral species. The conditional probabilities in the tree are obtained from evolutionary substitution models, and the computation of likelihoods are achieved by a recursion on the tree known as "pruning" [79]. This recursion is a special case of the junction tree algorithm.

Figure 2.5 provides examples of more complex graphical models that have been explored in bioinformatics. Figure 2.5(a) shows a *hidden*
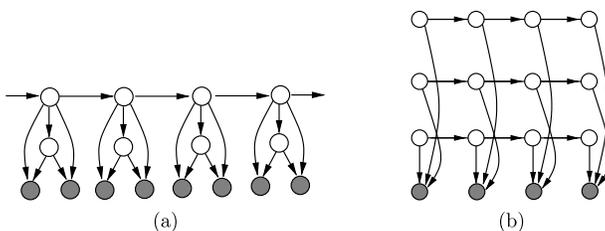

Fig. 2.5 Variations on the HMM used in bioinformatics. (a) A phylogenetic HMM. (b) The factorial HMM.

Markov phylogeny, an HMM in which the observations are sets of nucleotides related by a phylogenetic tree [165, 193, 218]. This model has proven useful for gene-finding in the human genome based on data for multiple primate species [165]. Figure 2.5(b) shows a *factorial HMM*, in which multiple chains are coupled by their links to a common set of observed variables [92]. This model captures the problem of *multi-locus linkage analysis* in genetics, where the state variables correspond to phase (maternal or paternal) along the chromosomes in meiosis [232].

2.4.5   Language and speech processing

In language problems, HMMs also play a fundamental role. An example is the part-of-speech problem, in which words in sentences are to be labeled as to their part of speech (noun, verb, adjective, etc.). Here the state variables are the parts of speech, and the transition matrix is estimated from a corpus via the EM algorithm [163]. The result of running the Viterbi algorithm on a new sentence is a tagging of the sentence according to the hypothesized parts of speech of the words in the sentence. Moreover, essentially all modern speech recognition systems are built on the foundation of HMMs [117]. In this case the observations are generally a sequence of short-range speech spectra, and the states correspond to longer-range units of speech such as phonemes or pairs of phonemes. Large-scale systems are built by composing elementary HMMs into larger graphical models.

The graphical model shown in Figure 2.6(a) is a *coupled HMM*, in which two chains of state variables are coupled via links between the
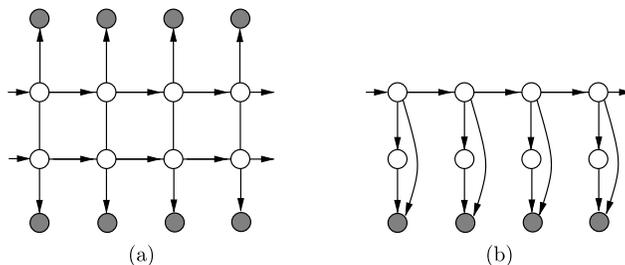

(a)   (b)

Fig. 2.6 Extensions of HMMs used in language and speech processing. (a) The coupled HHM. (b) An HMM with mixture-model emissions.

chains; this model is appropriate for fusing pairs of data streams such as audio and lip-reading data in speech recognition [208]. Figure 2.6(b) shows an HMM variant in which the state-dependent observation distribution is a finite mixture model. This variant is widely used in speech recognition systems [117].

Another model class that is widely studied in language processing are so-called "bag-of-words" models, which are of particular interest for modeling large-scale corpora of text documents. The terminology bag-of-words means that the order of words in a document is ignored — i.e., an assumption of exchangeability is made. The goal of such models is often that of finding latent "topics" in the corpus, and using these topics to cluster or classify the documents. An example bag-of-words model is the *latent Dirichlet allocation* model [32], in which a topic defines a probability distribution on words, and a document defines a probability distribution on topics. In particular, as shown in Figure 2.7, each document in a corpus is assumed to be generated by sampling a Dirichlet variable with hyperparameter α, and then repeatedly selecting a topic according to these Dirichlet probabilities, and choosing a word from the distribution associated with the selected topic.[2]

2.4.6 Image Processing and Spatial Statistics

For several decades, undirected graphical models or Markov random fields have played an important role in image processing
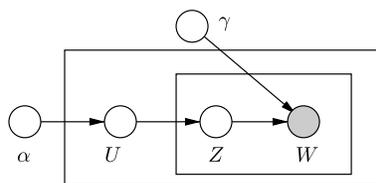
Fig. 2.7 Graphical illustration of the latent Dirichlet allocation model. The variable U, which is distributed as Dirichlet with parameter α, specifies the parameter for the multinomial "topic" variable Z. The "word" variable W is also multinomial conditioned on Z, with γ specifying the word probabilities associated with each topic. The rectangles, known as *plates*, denote conditionally independent replications of the random variables inside the rectangle.

---

[2] This model is discussed in more detail in Example 3.5 of Section 3.3.

Fig. 2.8 (a) The 4-nearest neighbor lattice model in 2D is often used for image modeling. (b) A multiscale quad tree approximation to a 2D lattice model [262]. Nodes in the original lattice (drawn in white) lie at t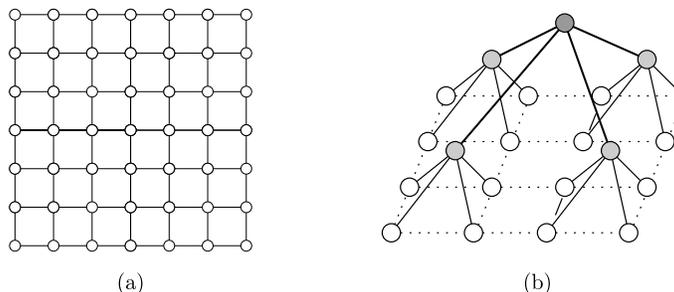he finest scale of the tree. The middle and top scales of the tree consist of auxiliary nodes (drawn in gray), introduced to model the fine scale behavior.

[e.g., 58, 89, 107, 263], as well as in spatial statistics more generally [25, 26, 27, 201]. For modeling an image, the simplest use of a Markov random field model is in the pixel domain, where each pixel in the image is associated with a vertex in an underlying graph. More structured models are based on feature vectors at each spatial location, where each feature could be a linear multiscale filter (e.g., a wavelet), or a more complicated nonlinear operator.

For image modeling, one very natural choice of graphical structure is a 2D lattice, such as the 4-nearest neighbor variety shown in Figure 2.8(a). The potential functions on the edges between adjacent pixels (or more generally, features) are typically chosen to enforce local smoothness conditions. Various tasks in image processing, including denoising, segmentation, and super-resolution, require solving an inference problem on such a Markov random field. However, exact inference for large-scale lattice models is intractable, which necessitates the use of approximate methods. Markov chain Monte Carlo methods are often used [89], but they can be too slow and computationally intensive for many applications. More recently, message-passing algorithms such as sum-product and tree-reweighted max-product have become popular as an approximate inference method for image processing and computer vision problems [e.g., 84, 85, 137, 168, 227].

An alternative strategy is to sidestep the intractability of the lattice model by replacing it with a simpler — albeit approximate — model.

For instance, multiscale quad trees, such as that illustrated in Figure 2.8(b), can be used to approximate lattice models [262]. The advantage of such a multiscale model is in permitting the application of efficient tree algorithms to perform exact inference. The trade-off is that the model is imperfect, and can introduce artifacts into image reconstructions.

2.4.7 Error-Correcting Coding

A central problem in communication theory is that of transmitting information, represented as a sequence of bits, from one point to another. Examples include transmission from a personal computer over a network, or from a satellite to a ground position. If the communication channel is noisy, then some of the transmitted bits may be corrupted. In order to combat this noisiness, a natural strategy is to add redundancy to the transmitted bits, thereby defining codewords. In principle, this coding strategy allows the transmission to be decoded perfectly even in the presence of some number of errors.

Many of the best codes in use today, including turbo codes and low-density parity check codes [e.g., 87, 166], are based on graphical models. Figure 2.9(a) provides an illustration of a very small parity check code, represented here in the factor graph formalism [142]. (A somewhat larger code is shown in Figure 2.9(b)). On the left, the six white nodes represent the bits x_i that comprise the codewords (i.e., binary sequences of length six), whereas the gray nodes represent noisy observations y_i associated with these bits. On the right, each of the black square nodes corresponds to a factor ψ_{stu} that represents the parity of the triple $\{x_s, x_t, x_u\}$. This parity relation, expressed mathematically as $x_s \oplus x_t \oplus x_u \equiv z_{stu}$ in modulo two arithmetic, can be represented as an undirected graphical model using a compatibility function of the form:

$$\psi_{stu}(x_s, x_t, x_u) := \begin{cases} 1 & \text{if } x_s \oplus x_t \oplus x_u = 1 \\ 0 & \text{otherwise.} \end{cases}$$

For the code shown in Figure 2.9, the parity checks range over the set of triples $\{1,3,4\}$, $\{1,3,5\}$, $\{2,4,6\}$, and $\{2,5,6\}$.

The decoding problem entails estimating which codeword was transmitted on the basis of a vector $y = (y_1, y_2, \ldots, y_m)$ of noisy observations.
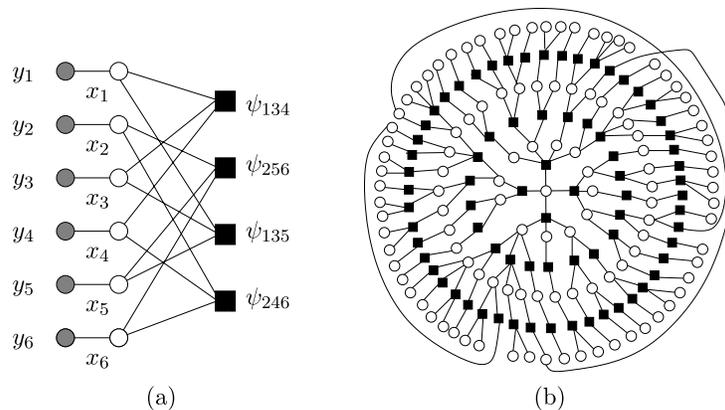
Fig. 2.9 (a) A factor graph representation of a parity check code of length $m = 6$. On the left, circular white nodes $\bigcirc$ represent the unobserved bits x_i that define the code, whereas circular gray nodes represent the observed values y_i, received from the channel. On the right, black squares $\blacksquare$ represent the associated factors, or parity checks. This particular code is a $(2,3)$ code, since each bit is connected to two parity variables, and each parity relation involves three bits. (b) A large factor graph with a "locally tree-like" structure. Random constructions of factor graphs on m vertices with bounded degree have cycles of typical length $\asymp \log m$; this tree-like property is essential to the success of the sum-product algorithm for approximate decoding [200, 169].

With the specification of a model for channel noise, this decoding problem can be cast as an inference problem. Depending on the loss function, optimal decoding is based either on computing the marginal probability $p(x_s = 1 \mid y)$ at each node, or computing the most likely codeword (i.e., the mode of the posterior). For the simple code of Figure 2.9(a), optimal decoding is easily achievable via the junction tree algorithm. Of interest in many applications, however, are much larger codes in which the number of bits is easily several thousand. The graphs underlying these codes are not of low treewidth, so that the junction tree algorithm is not viable. Moreover, MCMC algorithms have not been deployed successfully in this domain.

For many graphical codes, the most successful decoder is based on applying the sum-product algorithm, described in Section 2.6. Since the graphical models defining good codes invariably have cycles, the sum-product algorithm is not guaranteed to compute the correct marginals, nor even to converge. Nonetheless, the behavior of this

approximate decoding algorithm is remarkably good for a large class of codes. The behavior of sum-product algorithm is well understood in the asymptotic limit (as the code length m goes to infinity), where martingale arguments can be used to prove concentration results [160, 199]. For intermediate code lengths, in contrast, its behavior is not as well understood.

2.5   Exact Inference Algorithms

In this section, we turn to a description of the basic exact inference algorithms for graphical models. In computing a marginal probability, we must sum or integrate the joint probability distribution over one or more variables. We can perform this computation as a sequence of operations by choosing a specific ordering of the variables (and making an appeal to Fubini's theorem). Recall that for either directed or undirected graphical models, the joint probability is a factored expression over subsets of the variables. Consequently, we can make use of the distributive law to move individual sums or integrals across factors that do not involve the variables being summed or integrated over. The phrase "exact inference" refers to the (essentially symbolic) problem of organizing this sequential computation, including managing the intermediate factors that arise. Assuming that each individual sum or integral is performed exactly, then the overall algorithm yields an exact numerical result.

In order to obtain the marginal probability distribution of a single variable X_s, it suffices to choose a specific ordering of the remaining variables and to eliminate — that is, sum or integrate out — variables according to that order. Repeating this operation for each individual variable would yield the full set of marginals. However, this approach is wasteful because it neglects to share intermediate terms in the individual computations. The sum-product and junction tree algorithms are essentially dynamic programming algorithms based on a calculus for sharing intermediate terms. The algorithms involve "message-passing" operations on graphs, where the messages are exactly these shared intermediate terms. Upon convergence of the algorithms, we obtain marginal probabilities for all cliques of the original graph.

Both directed and undirected graphical models involve factorized expressions for joint probabilities, and it should come as no surprise that exact inference algorithms treat them in an essentially identical manner. Indeed, to permit a simple unified treatment of inference algorithms, it is convenient to convert directed models to undirected models and to work exclusively within the undirected formalism. We do this by observing that the terms in the directed factorization (2.1) are not necessarily defined on cliques, since the parents of a given vertex are not necessarily connected. We thus transform a directed graph to an undirected *moral graph*, in which all parents of each child are linked, and all edges are converted to undirected edges. In this moral graph, the factors are all defined on cliques, so that the moralized version of any directed factorization (2.1) is a special case of the undirected factorization (2.2). Throughout the rest of the survey, we assume that this transformation has been carried out.

2.5.1   Message-Passing on Trees

We now turn to a description of message-passing algorithms for exact inference on trees. Our treatment is brief; further details can be found in various sources [2, 65, 142, 125, 154]. We begin by observing that the cliques of a tree-structured graph $T = (V, E(T))$ are simply the individual nodes and edges. As a consequence, any tree-structured graphical model has the following factorization:

$$p(x_1, x_2, \ldots, x_m) = \frac{1}{Z} \prod_{s \in V} \psi_s(x_s) \prod_{(s,t) \in E(T)} \psi_{st}(x_s, x_t). \qquad (2.4)$$

Here, we describe how the sum-product algorithm computes the marginal distribution

$$\mu_s(x_s) := \sum_{\{x' \mid x'_s = x_s\}} p(x'_1, x'_2, \ldots, x'_m) \qquad (2.5)$$

for every node of a tree-structured graph. We will focus in detail on the case of discrete random variables, with the understanding that the computations carry over (at least in principle) to the continuous case by replacing sums with integrals.

Sum-product algorithm: The sum-product algorithm is a form of nonserial dynamic programming [19], which generalizes the usual serial form of deterministic dynamic programming [20] to arbitrary tree-structured graphs. The essential principle underlying dynamic programming (DP) is that of divide and conquer: we solve a large problem by breaking it down into a sequence of simpler problems. In the context of graphical models, the tree itself provides a natural way to break down the problem.

For an arbitrary $s \in V$, consider the set of its neighbors

$$N(s) := \{u \in V \mid (s, u) \in E\}. \tag{2.6}$$

For each $u \in N(s)$, let $T_u = (V_u, E_u)$ be the subgraph formed by the set of nodes (and edges joining them) that can be reached from u by paths that *do not* pass through node s. The key property of a tree is that each such subgraph T_u is again a tree, and T_u and T_v are vertex-disjoint for $u \neq v$. In this way, each vertex $u \in N(s)$ can be viewed as the root of a subtree T_u, as illustrated in Figure 2.10.

For each subtree T_t, we define the subvector $x_{V_t} := (x_u, u \in V_t)$ of variables associated with its vertex set. Now consider the collection of terms in Equation (2.4) associated with vertices or edges in T_t. We collect all of these terms into the following product:

$$p(x_{V_t}; T_t) \propto \prod_{u \in V_t} \psi_u(x_u) \prod_{(u,v) \in E_t} \psi_{uv}(x_u, x_v). \tag{2.7}$$

With this notation, the conditional independence properties of a tree allow the computation of the marginal at node s to be broken down

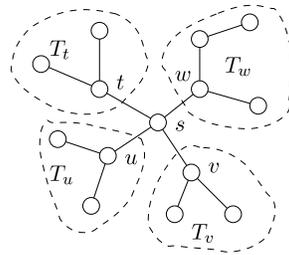Fig. 2.10 Decomposition of a tree, rooted at node s, into subtrees. Each neighbor (e.g., u) of node s is the root of a subtree (e.g., T_u). Subtrees T_u and T_v, for $u \neq v$, are disconnected when node s is removed from the graph.

into a product of subproblems, one for each of the subtrees in the set $\{T_t, t \in N(s)\}$, in the following way:

$$\mu_s(x_s) \; = \; \kappa \, \psi_s(x_s) \prod_{t \in N(s)} M^*_{ts}(x_s) \qquad (2.8\text{a})$$

$$M^*_{ts}(x_s) \; := \; \sum_{x'_{V_t}} \psi_{st}(x_s, x'_t) \; p(x'_{V_t}; T_t) \qquad (2.8\text{b})$$

In these equations, κ denotes a positive constant chosen to ensure that μ_s normalizes properly. For fixed x_s, the subproblem defining $M^*_{ts}(x_s)$ is again a tree-structured summation, albeit involving a subtree T_t *smaller* than the original tree T. Therefore, it too can be broken down recursively in a similar fashion. In this way, the marginal at node s can be computed by a series of recursive updates.

Rather than applying the procedure described above to each node separately, the *sum-product algorithm* computes the marginals for all nodes simultaneously and in parallel. At each iteration, each node t passes a "message" to each of its neighbors $u \in N(t)$. This message, which we denote by $M_{tu}(x_u)$, is a function of the possible states $x_u \in \mathcal{X}_u$ (i.e., a vector of length $|\mathcal{X}_u|$ for discrete random variables). On the full graph, there are a total of $2|E|$ messages, one for each direction of each edge. This full collection of messages is updated, typically in parallel, according to the recursion

$$M_{ts}(x_s) \leftarrow \kappa \sum_{x'_t} \left\{ \psi_{st}(x_s, x'_t) \, \psi_t(x'_t) \prod_{u \in N(t)/s} M_{ut}(x'_t) \right\}, \qquad (2.9)$$

where $\kappa > 0$ again denotes a normalization constant. It can be shown [192] that for tree-structured graphs, iterates generated by the update (2.9) will converge to a unique fixed point $M^* = \{M^*_{st}, M^*_{ts}, \; (s,t) \in E\}$ after a finite number of iterations. Moreover, component M^*_{ts} of this fixed point is precisely equal, up to a normalization constant, to the subproblem defined in Equation (2.8b), which justifies our abuse of notation post hoc. Since the fixed point M^* specifies the solution to all of the subproblems, the marginal μ_s at every node $s \in V$ can be computed easily via Equation (2.8a).

Max-product algorithm: Suppose that the summation in the update (2.9) is replaced by a maximization. The resulting *max-product*

algorithm solves the problem of finding a mode of a tree-structured distribution. In this sense, it represents a generalization of the Viterbi algorithm [82] from chains to arbitrary tree-structured graphs. More specifically, the max-product updates will converge to another unique fixed point M^* — distinct, of course, from the sum-product fixed point. This fixed point can be used to compute the *max-marginal*

$$\nu_s(x_s) := \max_{\{x' \mid x_s' = x_s\}} p(x_1', x_2', \ldots, x_m') \qquad (2.10)$$

at each node of the graph, via the analog of Equation (2.7). Given these max-marginals, a standard back-tracking procedure can be used to compute a mode $\widehat{x} \in \arg\max_{x'} p(x_1', x_2', \ldots, x_m')$ of the distribution; see the papers [65, 244] for further details. More generally, updates of this form apply to arbitrary *commutative semirings* on tree-structured graphs [2, 65, 216, 237]. The pairs "sum-product" and "max-product" are two particular examples of such an algebraic structure.

2.5.2 Junction Tree Representation

We have seen that inference problems on trees can be solved exactly by recursive message-passing algorithms. Given a graph with cycles, a natural idea is to cluster its nodes so as to form a *clique tree* — that is, an acyclic graph whose nodes are formed by the maximal cliques of G. Having done so, it is tempting to apply a standard algorithm for inference on trees. However, the clique tree must satisfy an additional restriction so as to ensure correctness of these computations. In particular, since a given vertex $s \in V$ may appear in multiple cliques (say C_1 and C_2), what is required is a mechanism for enforcing consistency among the different appearances of the variable x_s. It turns out that the following property is necessary and sufficient to enforce such consistency:

Definition 2.1. A clique tree has the *running intersection property* if for any two clique nodes C_1 and C_2, all nodes on the unique path joining them contain the intersection $C_1 \cap C_2$. A clique tree with this property is known as a *junction tree*.

For what type of graphs can one build junction trees? An important result in graph theory establishes a correspondence between junction trees and triangulation. We say that a graph is *triangulated* if every cycle of length four or longer has a chord, meaning an edge joining a pair of nodes that are not adjacent on the cycle. A key theorem is that a graph G has a junction tree if and only if it is triangulated (see Lauritzen [153] for a proof). This result underlies the *junction tree algorithm* [154] for exact inference on arbitrary graphs:

(1) Given a graph with cycles G, triangulate it by adding edges as necessary.

(2) Form a junction tree associated with the triangulated graph $\widetilde{G}$.

(3) Run a tree inference algorithm on the junction tree.

Example 2.1 (Junction Tree). To illustrate the junction tree construction, consider the 3×3 grid shown in Figure 2.11(a). The first step is to form a triangulated version $\widetilde{G}$, as shown in Figure 2.11(b). Note that the graph would not be triangulated if the additional edge joining nodes 2 and 8 were not present. Without this edge, the 4-cycle $(2 - 4 - 8 - 6 - 2)$ would lack a chord. As a result of this additional edge, the junction tree has two 4-cliques in the middle, as shown in Figure 2.11(c).

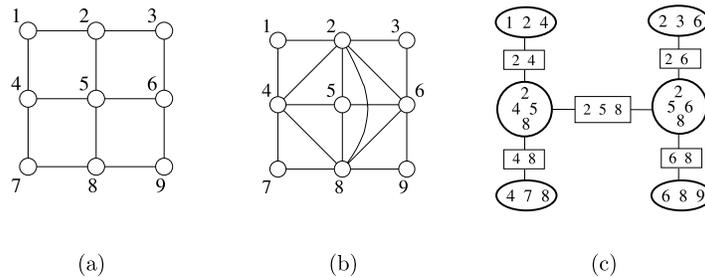(a)   (b)   (c)

Fig. 2.11 Illustration of junction tree construction. (a) Original graph is a 3×3 grid. (b) Triangulated version of original graph. Note the two 4-cliques in the middle. (c) Corresponding junction tree for triangulated graph in (b), with maximal cliques depicted within ellipses, and separator sets within rectangles.

In principle, the inference in the third step of the junction tree algorithm can be performed over an arbitrary commutative semiring (as mentioned in our earlier discussion of tree algorithms). We refer the reader to Dawid [65] for an extensive discussion of the max-product version of the junction tree algorithm. For concreteness, we limit our discussion here to the sum-product version of junction tree updates. There is an elegant way to express the basic algebraic operations in a junction tree inference algorithm that involves introducing potential functions not only on the cliques in the junction tree, but also on the *separators* in the junction tree — the intersections of cliques that are adjacent in the junction tree (the rectangles in Figure 2.11). Let $\phi_C(\cdot)$ denote a potential function defined over the subvector $x_C = (x_t,\ t \in C)$ on clique C, and let $\phi_S(\cdot)$ denote a potential on a separator S. We initialize the clique potentials by assigning each compatibility function in the original graph to (exactly) one clique potential and taking the product over these compatibility functions. The separator potentials are initialized to unity. Given this set-up, the basic message-passing step of the junction tree algorithm are given by

$$\widetilde{\phi}_S(x_S) \leftarrow \sum_{x_{B \setminus S}} \phi_B(x_B), \quad \text{and} \tag{2.11a}$$

$$\phi_C(x_C) \leftarrow \frac{\widetilde{\phi}_S(x_S)}{\phi_S(x_S)} \phi_C(x_C), \tag{2.11b}$$

where in the continuous case the summation is replaced by a suitable integral. We refer to this pair of operations as "passing a message from clique B to clique C" (see Figure 2.12). It can be verified that if a message is passed from B to C, and subsequently from C to B, then the resulting clique potentials are consistent with each other, meaning that they agree with respect to the vertices S.
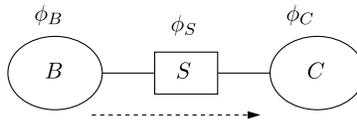

Fig. 2.12 A message-passing operation between cliques B and C via the separator set S.

After a round of message passing on the junction tree, it can be shown that the clique potentials are proportional to marginal probabilities throughout the junction tree. Specifically, letting $\mu_C(x_C)$ denote the marginal probability of x_C, we have $\mu_C(x_C) \propto \widetilde{\phi}_C(x_C)$ for all cliques C. This equivalence can be established by a suitable generalization of the proof of correctness of the sum-product algorithm sketched previously (see also Lauritzen [153]). Note that achieving local consistency between pairs of cliques is obviously a necessary condition if the clique potentials are to be proportional to marginal probabilities. Moreover, the significance of the running intersection property is now apparent; namely, it ensures that local consistency implies global consistency.

An important by-product of the junction tree algorithm is an alternative representation of a distribution p. Let $\mathcal{C}$ denote the set of all maximal cliques in $\widetilde{G}$ (i.e., nodes in the junction tree), and let $\mathcal{S}$ represent the set of all separator sets (i.e., intersections between cliques that are adjacent in the junction tree). Note that a given separator set S may occur multiple times in the junction tree. For each separator set $S \in \mathcal{S}$, let $d(S)$ denote the number of maximal cliques to which it is adjacent. The junction tree framework guarantees that the distribution p factorizes in the form

$$p(x_1, x_2, \ldots, x_m) = \frac{\prod\limits_{C \in \mathcal{C}} \mu_C(x_C)}{\prod\limits_{S \in \mathcal{S}} [\mu_S(x_S)]^{d(S)-1}}, \qquad (2.12)$$

where μ_C and μ_S are the marginal distributions over the cliques and separator sets, respectively. Observe that unlike the factorization (2.2), the decomposition (2.12) is directly in terms of marginal distributions, and does not require a normalization constant (i.e., $Z = 1$).

Example 2.2 (Markov Chain). Consider the Markov chain $p(x_1, x_2, x_3) = p(x_1)\, p(x_2 \,|\, x_1)\, p(x_3 \,|\, x_2)$. The cliques in a graphical model representation are $\{1, 2\}$ and $\{2, 3\}$, with separator $\{2\}$. Clearly the distribution cannot be written as the product of marginals involving only the cliques. It can, however, be written in terms of marginals if we

include the separator:

$$p(x_1, x_2, x_3) = \frac{p(x_1, x_2)\, p(x_2, x_3)}{p(x_2)}.$$

Moreover, it can be easily verified that these marginals result from a single application of the updates (2.11), given the initialization $\phi_{\{1,2\}}(x_1, x_2) = p(x_1)p(x_2 \,|\, x_1)$ and $\phi_{\{2,3\}}(x_2, x_3) = p(x_3 \,|\, x_2)$.

To anticipate part of our development in the sequel, it is helpful to consider the following "inverse" perspective on the junction tree representation. Suppose that we are given a set of functions $\{\tau_C,\ C \in \mathcal{C}\}$ and $\{\tau_S,\ S \in \mathcal{S}\}$ associated with the cliques and separator sets in the junction tree. What conditions are necessary to ensure that these functions are valid marginals for some distribution? Suppose that the functions are *locally consistent* in the following sense:

$$\forall\, S \in \mathcal{S}, \qquad \sum_{x'_S} \tau_S(x'_S) = 1, \qquad \text{and} \quad (2.13a)$$

$$\forall\, C \in \mathcal{C}, S \subseteq C, \qquad \sum_{\{x'_C \,|\, x'_S = x_S\}} \tau_C(x'_C) = \tau_S(x_S). \qquad (2.13b)$$

The essence of the junction tree theory described above is that such local consistency is both necessary and sufficient to ensure that these functions are valid marginals for some distribution. For the sake of future reference, we state this result in the following:

Proposition 2.1. A candidate set of marginal distribution $\{\tau_C, C \in \mathcal{C}\}$ and $\{\tau_C,\ C \in \mathcal{S}\}$ on the cliques and separator sets (respectively) of a junction tree is globally consistent if and only if the normalization condition (2.13a) is satisfied for all separator sets $S \in \mathcal{J}$, and the marginalization condition (2.13b) is satisfied for all cliques $C \in \mathcal{C}$, and separator sets $S \subseteq C$. Moreover, any such locally consistent quantities are the marginals of the probability distribution defined by Equation (2.12).

This particular consequence of the junction tree representation will play a fundamental role in our development in the sequel.

Finally, let us turn to the key issue of the computational complexity of the junction tree algorithm. Inspection of Equation (2.11) reveals that the computational costs grow exponentially in the size of the maximal clique in the junction tree. Clearly then, it is of interest to control the size of this clique. The size of the maximal clique over all possible triangulations of a graph is an important graph-theoretic quantity known as the *treewidth* of the graph.[3] Thus, the complexity of the junction tree algorithm is exponential in the treewidth.

For certain classes of graphs, including chains and trees, the treewidth is small and the junction tree algorithm provides an effective solution to inference problems. Such families include many well-known graphical model architectures, and the junction tree algorithm subsumes the classical recursive algorithms, including the pruning and peeling algorithms from computational genetics [79], the forward–backward algorithms for hidden Markov models [196], and the Kalman filtering-smoothing algorithms for state-space models [126]. On the other hand, there are many graphical models, including several of the examples treated in Section 2.4, for which the treewidth is infeasibly large. Coping with such models requires leaving behind the junction tree framework, and turning to approximate inference algorithms.

2.6   Message-passing Algorithms for Approximate Inference

It is the goal of the remainder of the survey to develop a general theoretical framework for understanding and analyzing *variational methods* for computing approximations to marginal distributions and likelihoods, as well as for solving integer programming problems. Doing so requires mathematical background on convex analysis and exponential families, which we provide starting in Section 3. Historically, many of these algorithms have been developed without this background, but rather via physical intuition or on the basis of analogies to exact or Monte Carlo algorithms. In this section, we give a high-level description of this flavor for two variational inference algorithms, with the goal of highlighting their simple and intuitive nature.

---

[3] To be more precise, the treewidth is one less than the size of this largest clique [33].

The first variational algorithm that we consider is sum-product message-passing applied to a graph with cycles, where it is known as the "loopy" sum-product or belief propagation algorithm. Recall that the sum-product algorithm is an exact inference algorithm for trees. From an algorithmic point of view, however, there is nothing to prevent one from running the procedure on a graph with cycles. More specifically, the message updates (2.9) can be applied at a given node while ignoring the presence of cycles — essentially pretending that any given node is embedded in a tree. Intuitively, such an algorithm might be expected to work well if the graph is sparse, such that the effect of messages propagating around cycles is appropriately diminished, or if suitable symmetries are present. As discussed in Section 2.4, this algorithm is in fact successfully used in various applications. Also, an analogous form of the max-product algorithm can be used for computing approximate modes in graphical models with cycles.

A second variational algorithm is the so-called *naive mean field* algorithm. For concreteness, we describe here this algorithm in application to the Ising model, which is an undirected graphical model or Markov random field involving a binary random vector $X \in \{0,1\}^m$, in which pairs of adjacent nodes are coupled with a weight θ_{st}, and each node has an observation weight θ_s. (See Example 3.1 of Section 3.3 for a more detailed description of this model.) To provide some intuition for the naive mean field updates, we begin by describing the Gibbs sampler, a particular type of Monte Carlo Markov chain algorithm, for this particular model. The basic step of a Gibbs sampler is to choose a node $s \in V$ randomly, and then to update the state of the associated random variable according to the conditional probability with neighboring states fixed. More precisely, denoting by $N(s)$ the neighbors of a node $s \in V$, and letting $X_{N(s)}^{(n)}$ denote the state of the neighbors of s at iteration n, the Gibbs update for vertex s takes the form

$$
X_s^{(n+1)} = \begin{cases} 1 & \text{if } U \le \{1 + \exp[-(\theta_s + \sum_{t \in N(s)} \theta_{st} X_t^{(n)})]\}^{-1} \\ 0 & \text{otherwise} \end{cases}, \quad (2.14)
$$

where U is a sample from a uniform distribution $\mathcal{U}[0,1]$.

In a dense graph, such that the cardinality of the neighborhood set $N(s)$ is large, we might attempt to invoke a law of large numbers

or some other concentration result for $\sum_{t \in N(s)} \theta_{st} X_t^{(n)}$. To the extent that such sums are concentrated, it might make sense to replace sample values with expectations. That is, letting μ_s denote an estimate of the marginal probability $\mathbb{P}[X_s = 1]$ at each vertex $s \in V$, we might consider the following averaged version of Equation (2.14):

$$\mu_s \leftarrow \left\{ 1 + \exp\left[-(\theta_s + \sum_{t \in N(s)} \theta_{st}\mu_t) \right] \right\}^{-1}. \qquad (2.15)$$

Thus, rather than flipping the random variable X_s with a probability that depends on the state of its neighbors, we update a parameter μ_s deterministically that depends on the corresponding parameters at its neighbors. Equation (2.15) defines the naive mean field algorithm for the Ising model. As with the sum-product algorithm, the mean field algorithm can be viewed as a message-passing algorithm, in which the right-hand side of (2.15) represents the "message" arriving at vertex s.

At first sight, message-passing algorithms of this nature might seem rather mysterious, and do raise some questions. Do the updates have fixed points? Do the updates converge? What is the relation between the fixed points and the exact quantities? The goal of the rest of this survey is to shed some light on such issues. Ultimately, we will see that a broad class of message-passing algorithms, including the mean field updates, the sum-product and max-product algorithms, as well as various extensions of these methods, can all be understood as solving either exact or approximate versions of variational problems. Exponential families and convex analysis, which are the subject of the following section, provide the appropriate framework in which to develop these variational principles in an unified manner.

3

Graphical Models as Exponential Families

In this section, we describe how many graphical models are naturally viewed as exponential families, a broad class of distributions that have been extensively studied in the statistics literature [5, 11, 43, 74]. Taking the perspective of exponential families illuminates some fundamental connections between inference algorithms and the theory of convex analysis [38, 112, 203]. More specifically, as we shall see, various types of inference problems in graphical models can be understood in terms of mappings between mean parameters and canonical parameters.

3.1 Exponential Representations via Maximum Entropy

One way in which to motivate exponential family representations of graphical models is through the principle of maximum entropy [123, 264]. Here, so as to provide helpful intuition for our subsequent development, we describe a particularly simple version for a scalar random variable X. Suppose that given n independent and identically distributed (i.i.d.) observations $X^1, \ldots, X^n$, we compute the empirical expectations of certain functions — namely, the quantities

$$\widehat{\mu}_\alpha := \frac{1}{n} \sum_{i=1}^{n} \phi_\alpha(X^i), \quad \text{for all } \alpha \in \mathcal{I}, \tag{3.1}$$

where each α in some set $\mathcal{I}$ indexes a function $\phi_\alpha : \mathcal{X} \to \mathbb{R}$. For example, if we set $\phi_1(x) = x$ and $\phi_2(x) = x^2$, then the observations (3.1) correspond to empirical versions of the first and second moments of the random variable X. Based on the $|\mathcal{I}|$-dimensional vector of empirical expectations $\widehat{\mu} = (\widehat{\mu}_\alpha, \alpha \in \mathcal{I})$, our goal is to infer a full probability distribution over the random variable X. In particular, we represent the probability distribution as a density p absolutely continuous with respect to some measure ν. This base measure ν might be the counting measure on $\{0, 1, \ldots, r - 1\}$, in which case p is a probability mass function; alternatively, for a continuous random vector, the base measure ν could be the ordinary Lebesgue measure on $\mathbb{R}$.

For a given distribution p, let us consider the expectations $\mathbb{E}_p[\phi_\alpha(X)] := \int_\mathcal{X} \phi_\alpha(x) p(x) \nu(dx)$ for $\alpha \in \mathcal{I}$, assuming that they exist. We say that the distribution p is consistent with the data if

$$\mathbb{E}_p[\phi_\alpha(X)] = \widehat{\mu}_\alpha \quad \text{for all } \alpha \in \mathcal{I}.$$

In words, the expectations $\mathbb{E}_p[\phi_\alpha(X)]$ under the distribution p are matched to the expectations under the empirical distribution. An important observation is that generically, this problem is underdetermined, in that there are many distributions p that are consistent with the observations, so that we need a principle for choosing among them.

In order to develop such a principle, we begin by defining a functional of the density p, known as the *Shannon entropy*, via

$$H(p) := -\int_\mathcal{X} (\log p(x)) \, p(x) \, \nu(dx). \tag{3.2}$$

The principle of maximum entropy is to choose, from among the distributions consistent with the data, the distribution p^* whose Shannon entropy is maximal. More formally, letting $\mathcal{P}$ be the set of all probability distributions over the random variable X, the maximum entropy solution p^* is given by the solution to the following constrained optimization problem:

$$p^* := \arg\max_{p \in \mathcal{P}} H(p) \quad \text{subject to } \mathbb{E}_p[\phi_\alpha(X)] = \widehat{\mu}_\alpha \quad \text{for all } \alpha \in \mathcal{I}.$$
$$\tag{3.3}$$

One interpretation of this principle is as choosing the distribution with maximal uncertainty, as measured by the entropy functional (3.2), while remaining faithful to the data. Presuming that problem (3.3) is feasible (and under certain technical conditions to be explored in the sequel), it can be shown — by calculus of variations in the general continuous case, and by ordinary calculus in the discrete case — that the optimal solution p^* takes the form

$$p_\theta(x) \propto \exp\Big\{\sum_{\alpha \in \mathcal{I}} \theta_\alpha \phi_\alpha(x)\Big\}, \tag{3.4}$$

where $\theta \in \mathbb{R}^d$ represents a parameterization of the distribution in exponential family form. From this maximum entropy perspective, the parameters θ have a very specific interpretation as the Lagrange multipliers associated with the constraints specified by the empirical moments $\widehat{\mu}$. We explore this connection in much more depth in the following sections.

3.2 Basics of Exponential Families

With this motivating example in mind, let us now set up the framework of exponential families in more precise terms and greater generality. At a high level, an exponential family is a parameterized family of densities, taken with respect to some underlying measure ν.

Given a random vector $(X_1, X_2, \ldots, X_m)$ taking values in some space $\mathcal{X}^m = \otimes_{s=1}^m \mathcal{X}_s$, let $\phi = (\phi_\alpha, \ \alpha \in \mathcal{I})$ be a collection of functions $\phi_\alpha : \mathcal{X}^m \to \mathbb{R}$, known either as *potential functions* or *sufficient statistics*. Here $\mathcal{I}$ is an index set with $d = |\mathcal{I}|$ elements to be specified, so that ϕ can be viewed as a vector-valued mapping from $\mathcal{X}^m$ to $\mathbb{R}^d$. For a given vector of sufficient statistics ϕ, let $\theta = (\theta_\alpha, \ \alpha \in \mathcal{I})$ be an associated vector of *canonical* or *exponential* parameters. For each fixed $x \in \mathcal{X}^m$, we use $\langle \theta, \phi(x) \rangle$ to denote the Euclidean inner product in $\mathbb{R}^d$ of the two vectors θ and $\phi(x)$.

With this notation, the *exponential family* associated with ϕ consists of the following parameterized collection of density functions

$$p_\theta(x_1, x_2, \ldots, x_m) = \exp\big\{\langle \theta, \phi(x) \rangle - A(\theta)\big\}, \tag{3.5}$$

taken with respect[1] to $d\nu$. The quantity A, known as the *log partition function* or *cumulant function*, is defined by the integral

$$A(\theta) = \log \int_{\mathcal{X}^m} \exp\langle\theta, \phi(x)\rangle \nu(dx). \qquad (3.6)$$

Presuming that the integral is finite, this definition ensures that p_θ is properly normalized (i.e., $\int_{\mathcal{X}^m} p_\theta(x)\nu(dx) = 1$). With the set of potentials ϕ fixed, each parameter vector θ indexes a particular member p_θ of the family. The canonical parameters θ of interest belong to the set

$$\Omega := \{\theta \in \mathbb{R}^d \mid A(\theta) < +\infty\}. \qquad (3.7)$$

We will see shortly that A is a convex function of θ, which in turn implies that Ω must be a convex set. The log partition function A plays a prominent role in this survey.

The following notions will be important in subsequent development:

Regular families: An exponential family for which the domain Ω is an open set is known as a *regular* family. Although there do exist exponential families for which Ω is closed (for instance, see Brown [43]), herein we restrict our attention to regular exponential families.

Minimal: It is typical to define an exponential family with a vector of sufficient statistics $\phi = (\phi_\alpha, \, \alpha \in \mathcal{I})$ for which there does not exist a nonzero vector $a \in \mathbb{R}^d$ such that the linear combination

$$\langle a, \phi(x)\rangle = \sum_{\alpha \in \mathcal{I}} a_\alpha \phi_\alpha(x)$$

is equal to a constant (ν-almost everywhere). This condition gives rise to a so-called *minimal representation*, in which there is a unique parameter vector θ associated with each distribution.

Overcomplete: Instead of a minimal representation, it can be convenient to use a nonminimal or *overcomplete representation*, in which there exist linear combinations $\langle a, \phi(x)\rangle$ that are equal to a constant (ν-a.e.). In this case, there exists an entire affine subset of parameter vectors θ, each associated with the same distribution. The reader might question the utility of an overcomplete representation. Indeed, from a statistical perspective, it can be undesirable since the identifiability of

---

[1] More precisely, for any measurable set S, we have $\mathbb{P}[X \in S] = \int_S p_\theta(x)\nu(dx)$.

the parameter vector θ is lost. However, this notion of overcompleteness is useful in understanding the sum-product algorithm and its variants (see Section 4).

Table 3.1 provides some examples of well-known scalar exponential families. Observe that all of these families are both regular (since Ω is open), and minimal (since the vector of sufficient statistics ϕ does not satisfy any linear relations).

3.3　Examples of Graphical Models in Exponential Form

The scalar examples in Table 3.1 serve as building blocks for the construction of more complex exponential families for which graphical structure plays a role. Whereas earlier we described graphical models in terms of products of functions, as in Equations (2.1) and (2.2), these products become additive decompositions in the exponential family setting. Here, we discuss a few well-known examples of graphical models as exponential families. Those readers familiar with such formulations may skip directly to Section 3.4, where we continue our general discussion of exponential families.

Example 3.1 (Ising Model). The *Ising model* from statistical physics [12, 119] is a classical example of a graphical model in exponential form. Consider a graph $G = (V, E)$ and suppose that the random variable X_s associated with node $s \in V$ is Bernoulli, say taking the "spin" values $\{-1, +1\}$. In the context of statistical physics, these values might represent the orientations of magnets in a field, or the presence/absence of particles in a gas. The Ising model and variations on it have also been used in image processing and spatial statistics [27, 89, 101], where X_s might correspond to pixel values in a black-and-white image, and for modeling of social networks, for instance the voting patterns of politicians [8].

Components X_s and X_t of the full random vector X are allowed to interact directly only if s and t are joined by an edge in the graph. This set-up leads to an exponential family consisting of the densities

$$p_\theta(x) = \exp\Big\{ \sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in E} \theta_{st} x_s x_t - A(\theta) \Big\}, \qquad (3.8)$$

Table 3.1 Several well-known classes of scalar random variables as exponential families.

Family	$\mathcal{X}$	ν $h(\cdot)$	Sufficient statistics $\langle \theta, \phi(x) \rangle$	$A(\theta)$	Ω
Bernoulli	$\{0,1\}$	Counting	θx	$\log[1 + \exp(\theta)]$	$\mathbb{R}$
Gaussian	$\mathbb{R}$	Lebesgue	θx	$\frac{1}{2}\theta^2$	$\mathbb{R}$
Location family		$h(x) = \frac{\exp(-x^2/2)}{\sqrt{2\pi}}$			
Gaussian Location-scale	$\mathbb{R}$	Lebesgue $h(x) = \frac{1}{\sqrt{2\pi}}$	$\theta_1 x + \theta_2 x^2$	$-\frac{\theta_1^2}{4\theta_2} - \frac{1}{2}\log(-2\theta_2)$	$\{(\theta_1,\theta_2) \in \mathbb{R}^2 \mid \theta_2 < 0\}$
Exponential	$(0, +\infty)$	Lebesgue	θx	$-\log(-\theta)$	$(-\infty, 0)$
Poisson	$\{0,1,2\ldots\}$	Counting $h(x) = 1/x!$	θx	$\exp(\theta)$	$\mathbb{R}$
Beta	$(0,1)$	Lebesgue	$\theta_1 \log x + \theta_2 \log(1-x)$	$\sum_{i=1}^{2} \log\Gamma(\theta_i + 1)$ $-\log\Gamma(\sum_{i=1}^{2}(\theta_i + 1))$	$(-1, +\infty)^2$

Notes: In all cases, the base measure ν is either Lebesgue or counting measure, suitably restricted to the space $\mathcal{X}$, and (in some cases) modulated by a factor $h(\cdot)$. All of these examples are both minimal and regular.

where the base measure ν is the counting measure[2] restricted to $\{0,1\}^m$. Here $\theta_{st} \in \mathbb{R}$ is the strength of edge (s,t), and $\theta_s \in \mathbb{R}$ is a potential for node s, which models an "external field" in statistical physics, or a noisy observation in spatial statistics. Strictly speaking, the family of densities (3.8) is more general than the classical Ising model, in which θ_{st} is constant for all edges.

As an exponential family, the Ising index set $\mathcal{I}$ consists of the union $V \cup E$, and the dimension of the family is $d = m + |E|$. The log partition function is given by the sum

$$A(\theta) = \log \sum_{x \in \{0,1\}^m} \exp\left\{\sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in E} \theta_{st} x_s x_t\right\}. \qquad (3.9)$$

Since this sum is finite for all choices of $\theta \in \mathbb{R}^d$, the domain Ω is the full space $\mathbb{R}^d$, and the family is regular. Moreover, it is a minimal representation, since there is no nontrivial linear combination of the potentials equal to a constant ν-a.e.

The standard Ising model can be generalized in a number of different ways. Although Equation (3.8) includes only pairwise interactions, higher-order interactions among the random variables can also be included. For example, in order to include coupling within the 3-clique $\{s,t,u\}$, we add a monomial of the form $x_s x_t x_u$, with corresponding canonical parameter θ_{stu}, to Equation (3.8). More generally, to incorporate coupling in k-cliques, we can add monomials up to order k, which lead to so-called k-spin models in the statistical physics literature. At the upper extreme, taking $k = m$ amounts to connecting all nodes in the graphical model, which allows one to represent any distribution over a binary random vector $X \in \{0,1\}^m$.

Example 3.2 (Metric Labeling and Potts Model). Here, we consider another generalization of the Ising model: suppose that the random variable X_s at each node $s \in V$ takes values in the discrete space

---

[2] Explicitly, for each singleton set $\{x\}$, this counting measure is defined by $\nu(\{x\}) = 1$ if $x \in \{0,1\}^m$ and $\nu(\{x\}) = 0$ otherwise, and extended to arbitrary sets by sub-additivity.

$\mathcal{X} := \{0, 1, \ldots, r - 1\}$, for some integer $r > 2$. One interpretation of a state $j \in \mathcal{X}$ is as a label, for instance defining membership in an image segmentation problem. Each pairing of a node $s \in V$ and a state $j \in \mathcal{X}$ yields a sufficient statistic

$$\mathbb{I}_{s;j}(x_s) = \begin{cases} 1 & \text{if } x_s = j \\ 0 & \text{otherwise,} \end{cases} \tag{3.10}$$

with an associated vector $\theta_s = \{\theta_{s;j}, \ j = 0, \ldots, r - 1\}$ of canonical parameters. Moreover, for each edge (s, t) and pair of values $(j, k) \in \mathcal{X} \times \mathcal{X}$, define the sufficient statistics

$$\mathbb{I}_{st;jk}(x_s, x_t) = \begin{cases} 1 & \text{if } x_s = j \quad \text{and} \quad x_t = k, \\ 0 & \text{otherwise,} \end{cases} \tag{3.11}$$

as well as the associated parameter $\theta_{st;jk} \in \mathbb{R}$. Viewed as an exponential family, the chosen collection of sufficient statistics defines an exponential family with dimension $d = r|V| + r^2|E|$. Like the Ising model, the log partition function is everywhere finite, so that the family is regular. In contrast to the Ising model, however, the family is overcomplete: indeed, the sufficient statistics satisfy various linear relations — for instance, $\sum_{j \in \mathcal{X}} \mathbb{I}_{s;j}(x_s) = 1$ for all $x_s \in \mathcal{X}$.

A special case of this model is the *metric labeling problem*, in which a metric $\rho : \mathcal{X} \times \mathcal{X} \to [0, \infty)$ specifies the parameterization — that is, $\theta_{st;jk} = -\rho(j, k)$ for all $(j, k) \in \mathcal{X} \times \mathcal{X}$. Consequently, the canonical parameters satisfy the relations $\theta_{st;kk} = 0$ for all $k \in \mathcal{X}$, $\theta_{st;jk} < 0$ for all $j \neq k$, and satisfy the reversed triangle inequality (that is, $\theta_{st;j\ell} \geq \theta_{st;jk} + \theta_{st;k\ell}$ for all triples (j, k, ℓ)). Another special case is the *Potts model* from statistical physics, in which case $\theta_{st;kk} = \alpha$ for all $k \in \mathcal{X}$, and $\theta_{st;jk} = \beta$ for all $j \neq k$.

We now turn to an important class of graphical models based on continuous random variables:

Example 3.3 (Gaussian MRF). Given an undirected graph G with vertex set $V = \{1, \ldots, m\}$, a Gaussian Markov random field [220] consists of a multivariate Gaussian random vector $(X_1, \ldots, X_m)$ that
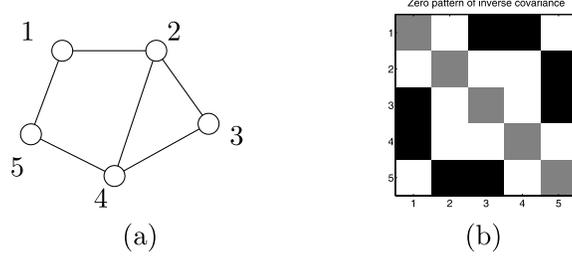
Fig. 3.1 (a) Undirected graphical model on five nodes. (b) For a Gaussian Markov random field, the zero pattern of the inverse covariance or precision matrix respects the graph structure: for any pair $i \neq j$, if $(i,j) \notin E$, then $\Theta_{ij} = 0$.

respects the Markov properties of G (see Section 2.2). It can be represented in exponential form using the collection of sufficient statistics $(x_s, x_s^2, s \in V; \ x_s x_t, \ (s,t) \in E)$. We define an m-vector θ of parameters associated with the vector of sufficient statistics $x = (x_1, \ldots, x_m)$, and a symmetric matrix $\Theta \in \mathbb{R}^{m \times m}$ associated with the matrix xx^T. Concretely, the matrix Θ is the negative of the inverse covariance or precision matrix, and by the Hammersley–Clifford theorem [25, 102, 153], it has the property that $\Theta_{st} = 0$ if $(s,t) \notin E$, as illustrated in Figure 3.1. Consequently, the dimension of the resulting exponential family is $d = 2m + |E|$.

With this set-up, the multivariate Gaussian is an exponential family[3] of the form:

$$p_\theta(x) = \exp \left\{ \langle \theta, x \rangle + \frac{1}{2} \langle\!\langle \Theta, \ xx^T \rangle\!\rangle - A(\theta, \Theta) \right\}, \qquad (3.12)$$

where $\langle \theta, x \rangle := \sum_{i=1}^m \theta_i x_i$ is the Euclidean inner product on $\mathbb{R}^m$, and

$$\langle\!\langle \Theta, \ xx^T \rangle\!\rangle := \mathrm{trace}(\Theta xx^T) = \sum_{i=1}^m \sum_{j=1}^m \Theta_{ij} x_i x_j \qquad (3.13)$$

is the Frobenius inner product on symmetric matrices. The integral defining $A(\theta, \Theta)$ is finite only if $\Theta \prec 0$, so that

$$\Omega = \{ (\theta, \Theta) \in \mathbb{R}^m \times \mathbb{R}^{m \times m} \mid \Theta \prec 0, \ \Theta = \Theta^T \}. \qquad (3.14)$$

---

[3] Our inclusion of the $\frac{1}{2}$-factor in the term $\frac{1}{2} \langle\!\langle \Theta, \ xx^T \rangle\!\rangle$ is for later technical convenience.
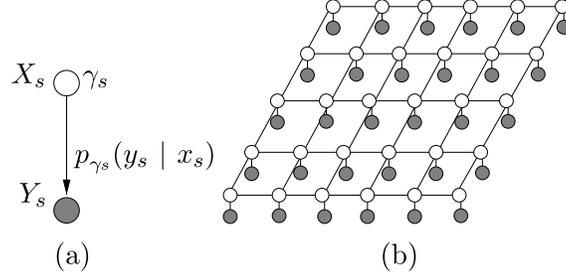
Fig. 3.2 (a) Graphical representation of a finite mixture of Gaussians: X_s is a multinomial label for the mixture component, whereas Y_s is conditionally Gaussian given $X_s = j$, with Gaussian density $p_{\gamma_s}(y_s \mid x_s)$ in exponential form. (b) Coupled mixture-of-Gaussian graphical model, in which the vector $X = (X_1, \dots, X_m)$ are a Markov random field on an undirected graph, and the elements of $(Y_1, \dots, Y_m)$ are conditionally independent given X.

Graphical models are not limited to cases in which the random variables at each node belong to the same exponential family. More generally, we can consider heterogeneous combinations of exponential family members, as illustrated by the following examples.

Example 3.4 (Mixture Models). As shown in Figure 3.2(a), a scalar mixture model has a very simple graphical interpretation. Concretely, in order to form a finite mixture of Gaussians, let X_s be a multinomial variable, taking values in $\{0, 1, \dots, r-1\}$. The role of X_s is to specify the choice of mixture component, so that our mixture model has r components in total. As in the Potts model described in Example 3.2, the distribution of this random variable is exponential family with sufficient statistics $\{\mathbb{I}_j[x_s], j = 0, \dots, r-1\}$, and associated canonical parameters $\{\alpha_{s;0}, \dots, \alpha_{s;r-1}\}$.

In order to form the finite mixture, conditioned on $X_s = j$, we now let Y_s be conditionally Gaussian with mean and variance (μ_j, σ_j^2). Each such conditional distribution can be written in exponential family form in terms of the sufficient statistics $\{y_s, y_s^2\}$, with an associated pair of canonical parameters $(\gamma_{s;j}, \gamma'_{s;j}) := (\frac{\mu_j}{\sigma_j^2}, -\frac{1}{2\sigma_j^2})$. Overall, we obtain the exponential family with the canonical parameters

$$\theta_s := \Big(\underbrace{\alpha_{s;0}, \dots, \alpha_{s;r-1}}_{\alpha_s}, \underbrace{\gamma_{s;0}, \dots, \gamma_{s;r-1}}_{\gamma_s}, \underbrace{\gamma'_{s;0}, \dots, \gamma'_{s;r-1}}_{\gamma'_s} \Big),$$

and a density of the form:

$$p_{\theta_s}(y_s, x_s) = p_\alpha(x_s)\, p_{\gamma_s}(y_s \,|\, x_s)$$

$$\propto \exp\left\{ \sum_{j=0}^{r-1} \alpha_{s;j} \mathbb{I}_j(x_s) + \sum_{j=0}^{r-1} \mathbb{I}_j(x_s) \left[\gamma_{s;j} y_s + \gamma'_{s;j} y_s^2 \right] \right\}.$$

(3.15)

The pair (X_s, Y_s) serves a basic block for building more sophisticated graphical models, suitable for describing multivariate data $((X_1, Y_1), \ldots, (X_m, Y_m))$. For instance, suppose that the vector $X = (X_1, \ldots, X_m)$ of multinomial variates is modeled as a Markov random field (see Example 3.2 on the Potts model) with respect to some underlying graph $G = (V, E)$, and the variables Y_s are conditionally independent given $\{X = x\}$. These assumptions lead to an exponential family $p_\theta(y, x)$ with parameters $\theta = (\alpha, \gamma)$, and density

$$p_\alpha(x) p_\gamma(y \,|\, x) \propto \exp\left\{ \sum_{s \in V} \alpha_s(x_s) + \sum_{(s,t) \in E} \alpha_{st}(x_s, x_t) \right\} \prod_{s \in V} p_{\gamma_s}(y_s \,|\, x_s),$$

(3.16)

corresponding to a product of the Potts-type distribution $p_\alpha(x)$ over X, and the local conditional distributions $p_{\gamma_s}(y_s \,|\, x_s)$. Here, we have used $\alpha_s(x_s)$ as a shorthand for the exponential family representation $\sum_{j=0}^{r-1} \alpha_{s;j} \mathbb{I}_j(x_s)$, and similarly for the quantity $\alpha_{st}(x_s, x_t)$; see Example 3.2 where this notation was used.

Whereas the mixture model just described is a two-level hierarchy, the following example involves three distinct levels:

Example 3.5 (Latent Dirichlet Allocation). The *latent Dirichlet allocation* model [32] is a particular type of hierarchical Bayes model for capturing the statistical dependencies among words in a corpus of documents. It involves three different types of random variables: "documents" U, "topics" Z, and "words" W. Words are multinomial random variables ranging over some vocabulary. Topics are also multinomial random variables. Associated to each value of the topic variable there is a distribution over words. A document is a distribution over topics.

Finally, a corpus is defined by placing a distribution on the documents. For the latent Dirichlet allocation model, this latter distribution is a Dirichlet distribution.

More formally, words W are drawn from a multinomial distribution, $\mathbb{P}(W = j \,|\, Z = i; \gamma) = \exp(\gamma_{ij})$, for $j = 0, 1, \ldots, k - 1$, where γ_{ij} is a parameter encoding the probability of the jth word under the ith topic. This conditional distribution can be expressed as an exponential family in terms of indicator functions as follows:

$$p_\gamma(w \mid z) \propto \exp\Big(\sum_{i=0}^{r-1}\sum_{j=0}^{k-1}\gamma_{ij}\mathbb{I}_i(z)\,\mathbb{I}_j(w)\Big), \qquad (3.17)$$

where $\mathbb{I}_i(z)$ is an $\{0, 1\}$-valued indicator for the event $\{Z = i\}$, and similarly for $\mathbb{I}_j(w)$. At the next level of the hierarchy (see Figure 3.3), the topic variable Z also follows a multinomial distribution whose parameters are determined by the Dirichlet variable as follows:

$$p(z \mid u) \propto \exp\Big\{\sum_{i=0}^{r-1}\mathbb{I}_i[z]\log u_i\Big\}. \qquad (3.18)$$

Finally, at the top level of the hierarchy, the Dirichlet variable U has a density with respect to Lebesgue measure of the form $p_\alpha(u) \propto \exp\{\sum_{i=0}^{r-1}\alpha_i \log u_i\}$. Overall then, for a single triplet $X := (U, Z, W)$, the LDA model is an exponential family with parameter vector $\theta := (\alpha, \gamma)$, and an associated density of the form:

$$p_\alpha(u)p(z \mid u)\,p_\gamma(w \mid z) \propto \exp\Big\{\sum_{i=0}^{r-1}\alpha_i \log u_i + \sum_{i=0}^{r-1}\mathbb{I}_i[z]\log u_i\Big\}$$

$$\times \exp\Big\{\sum_{i=0}^{r-1}\sum_{j=0}^{k-1}\gamma_{ij}\mathbb{I}_i[z]\mathbb{I}_j[w]\Big\}. \qquad (3.19)$$

The sufficient statistics ϕ consist of the collections of functions $\{\log u_i\}$, $\{\mathbb{I}_i[z]\log u_i\}$, and $\{\mathbb{I}_i[z]\mathbb{I}_j[w]\}$. As illustrated in Figure 3.3, the full LDA model entails replicating these types of local structures many times.

Many graphical models include what are known as "hard-core" constraints, meaning that a subset of configurations are forbidden.
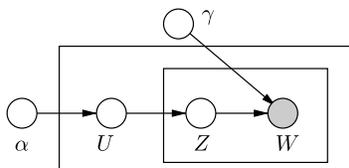
Fig. 3.3 Graphical illustration of the Latent Dirichlet allocation (LDA) model. The word variable W is multinomial conditioned on the underlying topic Z, where γ specifies the topic distributions. The topics Z are also modeled as multinomial variables, with distributions parameterized by a probability vector U that follows a Dirichlet distribution with parameter α. This model is an example of a hierarchical Bayesian model. The rectangles, known as plates, denote replication of the random variables.

Instances of such problems include decoding of binary linear codes, and other combinatorial optimization problems (e.g., graph matching, covering, packing, etc.). At first glance, it might seem that such families of distributions cannot be represented as exponential families, since the density p_θ in any exponential family is strictly positive — that is, $p_\theta(x) > 0$ for all $x \in \mathcal{X}^m$. In the following example, we show that these hard-core constraints can be incorporated within the exponential family framework by making appropriate use of the underlying base measure ν.

Example 3.6 (Models with Hard Constraints). One important domain in which hard-core constraints arise is communication theory, and in particular the problem of error-control coding [87, 166, 200]. To motivate the coding problem, suppose that two people — following an old convention, let us call them Alice and Bob — wish to communicate. We assume that they can communicate by transmitting a sequence of $0s$ and $1s$, but make the problem interesting by assuming that the communication channel linking them behaves in a random manner. Concretely, in a "bit-flipping" channel, any binary digit transmitted by Alice is received correctly by Bob only with probability $1 - \epsilon$. As a probabilistic model, this channel can be modeled by the conditional distribution

$$p(y \mid x) := \begin{cases} 1 - \epsilon & \text{if } x = y \\ \epsilon & \text{if } x \neq y, \end{cases}$$

where $x \in \{0,1\}$ represents the bit transmitted by Alice, and $y \in \{0,1\}$ represents the bit received by Bob.

In order to mitigate the "noisiness" of the channel, Alice and Bob agree on the following *block coding* scheme: instead of transmitting a single bit, they communicate using strings $(x_1, x_2, \ldots, x_m)$ of bits, and moreover, they agree to use only a subset of the total number 2^m of length m binary strings. These privileged strings, which are known as *codewords*, can be defined using a particular type of graphical model. Concretely, suppose that Alice decides to choose from among strings $x \in \{0,1\}^m$ that satisfy a set of parity checks — say of the form $x_1 \oplus x_2 \oplus x_3 = 0$, where $\oplus$ denotes addition in modulo two arithmetic. Let us consider a collection F of such parity checks; each $a \in F$ enforces a parity check constraint on some subset $N(a) \subset \{1, \ldots, m\}$ of bits. If we define the indicator function

$$\psi_a(x_{N(a)}) := \begin{cases} 1 & \text{if } \bigoplus_{i \in N(a)} x_i = 0 \\ 0 & \text{otherwise,} \end{cases} \tag{3.20}$$

then a *binary linear code* consists of all bit strings $(x_1, x_2, \ldots, x_m) \in \{0,1\}^m$ for which $\prod_{a \in F} \psi_a(x_{N(a)}) = 1$. These constraints are known as "hard-core," since they take the value 0 for some settings of x, and hence eliminate such configurations from the support of the distribution. Letting $(y_1, \ldots, y_m) \in \{0,1\}^m$ denote the string of bits received by Bob, his goal is to use these observations to infer which codeword was transmitted by Alice. Depending on the error metric used, this decoding problem corresponds to either computing marginals or modes of the posterior distribution

$$p(x_1, \ldots, x_m \mid y_1, \ldots, y_m) \propto \prod_{i=1}^{m} p(y_i \mid x_i) \prod_{a \in F} \psi_a(x_{N(a)}). \tag{3.21}$$

This distribution can be described by a factor graph, with the bits x_i represented as unshaded circular variable nodes, the observed values y_i as shaded circular variable nodes, and the parity check indicator functions represented at the square factor nodes (see Figure 2.9).

The code can also be viewed as an exponential family on this graph, if we take densities with respect to an *appropriately chosen* base measure. In particular, let us use the parity checks to define the counting

measure restricted to valid codewords

$$\nu(\{x\}) := \prod_{a \in F} \psi_a(x_{N(a)}). \tag{3.22}$$

Moreover, since the quantities y_i are observed, they may be viewed as fixed, so that the conditional distribution $p(y_i \mid x_i)$ can be viewed as some function $q(\cdot)$ of x_i only. A little calculation shows that we can write this function in exponential form as

$$q(x_i; \theta_i) = \exp\{\theta_i x_i - \log(1 + \exp(\theta_i))\}, \tag{3.23}$$

where the exponential parameter θ_i is defined by the observation y_i and conditional distribution $p(y_i \mid x_i)$ via the relation $\theta_i = \log p(y_i \mid 1)/p(y_i \mid 0)$. In the particular case of the binary symmetric channel, these canonical parameters take the form

$$\theta_i = (2y_i - 1) \log \frac{1 - \epsilon}{\epsilon}. \tag{3.24}$$

Note that we are using the fact that the vector $(y_1, y_2, \ldots, y_m)$ is observed, and hence can be viewed as fixed. With this set-up, the distribution (3.21) can be cast as an exponential family, where the density is taken with respect to the restricted counting measure (3.22), and has the form $p_\theta(x \mid y) \propto \exp\{\sum_{i=1}^m \theta_i x_i\}$.

3.4 Mean Parameterization and Inference Problems

Thus far, we have characterized any exponential family member p_θ by its vector of canonical parameters $\theta \in \Omega$. As we discuss in this section, it turns out that any exponential family has an alternative parameterization in terms of a vector of *mean parameters*. Moreover, various statistical computations, among them marginalization and maximum likelihood estimation, can be understood as transforming from one parameterization to the other.

We digress momentarily to make an important remark regarding the role of observed variables or conditioning for the marginalization problem. As discussed earlier, applications of graphical models frequently involve conditioning on a subset of random variables — say

Y — that represent observed quantities, and computing marginals under the posterior distribution $p_\theta(x \mid y)$. The application to error-correcting coding, just discussed in Example 3.6, is one such example. So as to simplify notation in our development to follow, it is convenient to no longer make direct reference to observed variables, but rather discuss marginalization and other computational problems only in reference to unconditional forms of exponential families. For such computational purposes, there is no loss of generality in doing so, since the effects of observations can always be absorbed by modifying the canonical parameters θ and/or the sufficient statistics ϕ appropriately. (For instance, in Example 3.6, the noisy observation y_i at node i contributes a new factor $\exp(\theta_i x_i)$ to the exponential family factorization, as described in Equation (3.23).)

3.4.1 Mean Parameter Spaces and Marginal Polytopes

Let p be a given density defined with respect to the underlying base measure ν; for the moment, we do not assume that p is a member of an exponential family defined with respect to ν. The mean parameter μ_α associated with a sufficient statistic $\phi_\alpha : \mathcal{X}^m \to \mathbb{R}$ is defined by the expectation

$$\mu_\alpha = \mathbb{E}_p[\phi_\alpha(X)] = \int \phi_\alpha(x) p(x) \nu(dx), \quad \text{for } \alpha \in \mathcal{I}. \qquad (3.25)$$

In this way, we define a vector of *mean parameters* $(\mu_1, \ldots, \mu_d)$, one for each of the $|\mathcal{I}| = d$ sufficient statistics ϕ_α, with respect to an arbitrary density p. An interesting object is the set of all such vectors $\mu \in \mathbb{R}^d$ traced out as the underlying density p is varied. More formally, we define the set

$$\mathcal{M} := \{\, \mu \in \mathbb{R}^d \mid \exists\, p \text{ s.t. } \mathbb{E}_p[\phi_\alpha(X)] = \mu_\alpha \,\forall \alpha \in \mathcal{I}\}, \qquad (3.26)$$

corresponding to all realizable mean parameters. It is important to note that in this definition, we have *not* restricted the density p to the exponential family associated with the sufficient statistics ϕ and base measure ν. However, it turns out that under suitable technical conditions, this vector provides an alternative parameterization of this exponential family.

We illustrate these concepts with a continuation of a previous example:

Example 3.7 (Gaussian MRF Mean Parameters). Using the canonical parameterization of the Gaussian Markov random field provided in Example 3.3, the mean parameters for a Gaussian Markov random field are the second-order moment matrix $\Sigma := \mathbb{E}[XX^T] \in \mathbb{R}^{m \times m}$, and the mean vector $\mu = \mathbb{E}[X] \in \mathbb{R}^m$. For this particular model, it is straightforward to characterize the set $\mathcal{M}$ of globally realizable mean parameters (μ, Σ). We begin by recognizing that if (μ, Σ) are realized by some distribution (not necessarily Gaussian), then $\Sigma - \mu\mu^T$ must be a valid covariance matrix of the random vector X, implying that the positive semidefiniteness (PSD) condition $\Sigma - \mu\mu^T \succeq 0$ must hold. Conversely, any pair (μ, Σ) for which the PSD constraint holds, we may construct a multivariate Gaussian distribution with mean μ, and (possibly degenerate) covariance $\Sigma - \mu\mu^T$, which by construction realizes (μ, Σ). Thus, we have established that for a Gaussian Markov random field, the set $\mathcal{M}$ has the form

$$\mathcal{M} = \{(\mu, \Sigma) \in \mathbb{R}^m \times \mathcal{S}_+^m \mid \Sigma - \mu\mu^T \succeq 0\}, \qquad (3.27)$$

where $\mathcal{S}_+^m$ denotes the set of $m \times m$ symmetric positive semidefinite matrices. Figure 3.4 illustrates this set in the scalar case $(m = 1)$,
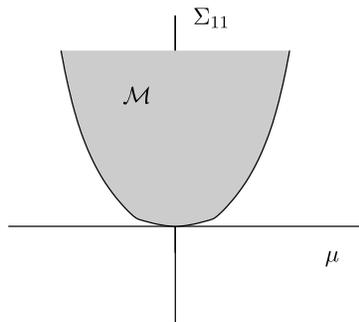

Fig. 3.4 Illustration of the set $\mathcal{M}$ for a scalar Gaussian: the model has two mean parameters $\mu = \mathbb{E}[X]$ and $\Sigma_{11} = \mathbb{E}[X^2]$, which must satisfy the quadratic constraint $\Sigma_{11} - \mu^2 \geq 0$. Note that the set $\mathcal{M}$ is convex, which is a general property.

where the mean parameters $\mu = \mathbb{E}[X]$ and $\Sigma_{11} = \mathbb{E}[X^2]$ must satisfy the quadratic constraint $\Sigma_{11} - \mu^2 \geq 0$.

The set $\mathcal{M}$ is always a convex subset of $\mathbb{R}^d$. Indeed, if μ and μ' are both elements of $\mathcal{M}$, then there must exist distributions p and p' that realize them, meaning that $\mu = \mathbb{E}_p[\phi(X)]$ and $\mu' = \mathbb{E}_{p'}[\phi(X)]$. For any $\lambda \in [0,1]$, the convex combination $\mu(\lambda) := \lambda\mu + (1 - \lambda)\mu'$ is realized by the mixture distribution $\lambda p + (1 - \lambda)p'$, so that $\mu(\lambda)$ also belongs to $\mathcal{M}$. In Appendix B.3, we summarize further properties of $\mathcal{M}$ that hold for general exponential families.

The case of discrete random variables yields a set $\mathcal{M}$ with some special properties. More specifically, for any random vector $(X_1, X_2, \ldots X_m)$ such that the associated state space $\mathcal{X}^m$ is finite, we have the representation

$$
\begin{aligned}
\mathcal{M} = \Bigg\{ \mu \in \mathbb{R}^d \ \mid \ \mu = \sum_{x \in \mathcal{X}^m} \phi(x)p(x) \quad \text{for some } p(x) \geq 0, \\
\sum_{x \in \mathcal{X}^m} p(x) = 1 \Bigg\} \\
= \text{conv}\{\phi(x), x \in \mathcal{X}^m\}, \hspace{3cm} (3.28)
\end{aligned}
$$

where conv denotes the convex hull operation (see Appendix A.2). Consequently, when $|\mathcal{X}^m|$ is finite, the set $\mathcal{M}$ is — by definition — a *convex polytope*.

The Minkowski–Weyl theorem [203], stated in Appendix A.2, provides an alternative description of a convex polytope. As opposed to the convex hull of a finite collection of vectors, any polytope $\mathcal{M}$ can also be characterized by a finite collection of linear inequality constraints. Explicitly, for any polytope $\mathcal{M}$, there exists a collection $\{(a_j, b_j) \in \mathbb{R}^d \times \mathbb{R} \mid j \in \mathcal{J}\}$ with $|\mathcal{J}|$ finite such that

$$
\mathcal{M} = \{\mu \in \mathbb{R}^d \mid \langle a_j, \mu \rangle \geq b_j \ \forall j \in \mathcal{J}\}. \hspace{2cm} (3.29)
$$

In geometric terms, this representation shows that $\mathcal{M}$ is equal to the intersection of a finite collection of half-spaces, as illustrated in Figure 3.5. Let us show the distinction between the convex hull (3.28) and linear inequality (3.29) representations using the Ising model.
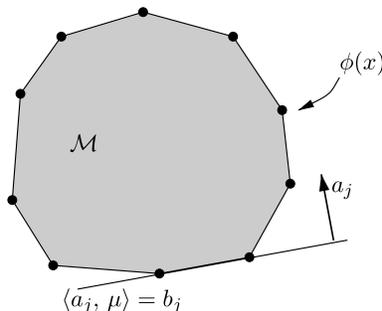
Fig. 3.5 Generic illustration of $\mathcal{M}$ for a discrete random variable with $|\mathcal{X}^m|$ finite. In this case, the set $\mathcal{M}$ is a convex polytope, corresponding to the convex hull of $\{\phi(x) \mid x \in \mathcal{X}^m\}$. By the Minkowski–Weyl theorem, this polytope can also be written as the intersection of a finite number of half-spaces, each of the form $\{\mu \in \mathbb{R}^d \mid \langle a_j, \mu \rangle \geq b_j\}$ for some pair $(a_j, b_j) \in \mathbb{R}^d \times \mathbb{R}$.

Example 3.8 (Ising Mean Parameters). Continuing from Example 3.1, the sufficient statistics for the Ising model are the singleton functions $(x_s,\ s \in V)$ and the pairwise functions $(x_s x_t,\ (s,t) \in E)$. The vector of sufficient statistics takes the form:

$$\phi(x) := \big(x_s, s \in V;\ x_s x_t,\ (s,t) \in E\big) \in \mathbb{R}^{|V|+|E|}. \qquad (3.30)$$

The associated mean parameters correspond to particular marginal probabilities, associated with nodes and edges of the graph G as

$$\mu_s = \mathbb{E}_p[X_s] = \mathbb{P}[X_s = 1] \quad \text{for all } s \in V, \text{ and} \qquad (3.31a)$$

$$\mu_{st} = \mathbb{E}_p[X_s X_t] = \mathbb{P}[(X_s, X_t) = (1,1)] \quad \text{for all } (s,t) \in E. \qquad (3.31b)$$

Consequently, the mean parameter vector $\mu \in \mathbb{R}^{|V|+|E|}$ consists of marginal probabilities over singletons (μ_s), and pairwise marginals over variable pairs on graph edges (μ_{st}). The set $\mathcal{M}$ consists of the convex hull of $\{\phi(x), x \in \{0,1\}^m\}$, where ϕ is given in Equation (3.30). In probabilistic terms, the set $\mathcal{M}$ corresponds to the set of all singleton and pairwise marginal probabilities that can be realized by some distribution over $(X_1, \ldots, X_m) \in \{0,1\}^m$. In the polyhedral combinatorics literature, this set is known as the *correlation polytope*, or the *cut polytope* [69, 187].

To make these ideas more concrete, consider the simplest nontrivial case: namely, a pair of variables (X_1, X_2), and the graph consisting of the single edge joining them. In this case, the set $\mathcal{M}$ is a polytope in three dimensions (two nodes plus one edge): it is the convex hull of the vectors $\{(x_1, x_2, x_1 x_2) \mid (x_1, x_2) \in \{0,1\}^2\}$, or more explicitly

$$\text{conv}\{(0,0,0), (1,0,0), (0,1,0), (1,1,1)\},$$

as illustrated in Figure 3.6.

Let us also consider the half-space representation (3.29) for this case. Elementary probability theory and a little calculation shows that the three mean parameters (μ_1, μ_2, μ_{12}) must satisfy the constraints $0 \leq \mu_{12} \leq \mu_i$ for $i = 1, 2$ and $1 + \mu_{12} - \mu_1 - \mu_2 \geq 0$. We can write these constraints in matrix-vector form as

$$
\begin{bmatrix}
0 & 0 & 1 \\
1 & 0 & -1 \\
0 & 1 & -1 \\
-1 & -1 & 1
\end{bmatrix}
\begin{bmatrix}
\mu_1 \\
\mu_2 \\
\mu_{12}
\end{bmatrix}
\geq
\begin{bmatrix}
0 \\
0 \\
0 \\
-1
\end{bmatrix}.
$$

These four constraints provide an alternative characterization of the 3D polytope illustrated in Figure 3.6.
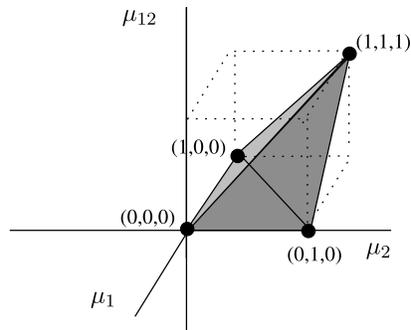
Fig. 3.6 Illustration of $\mathcal{M}$ for the special case of an Ising model with two variables $(X_1, X_2) \in \{0,1\}^2$. The four mean parameters $\mu_1 = \mathbb{E}[X_1]$, $\mu_2 = \mathbb{E}[X_2]$ and $\mu_{12} = \mathbb{E}[X_1 X_2]$ must satisfy the constraints $0 \leq \mu_{12} \leq \mu_i$ for $i = 1, 2$, and $1 + \mu_{12} - \mu_1 - \mu_2 \geq 0$. These constraints carve out a polytope with four facets, contained within the unit hypercube $[0,1]^3$.

Our next example deals with a interesting family of polytopes that arises in error-control coding and binary matroid theory:

Example 3.9 (Codeword Polytopes and Binary Matroids). Recall the definition of a binary linear code from Example 3.6: it corresponds to the subset of binary strings $x \in \{0,1\}^m$ that satisfy a set of parity check relations

$$\mathbb{C} := \big\{ x \in \{0,1\}^m \mid \prod_{a \in F} \psi_a(x_{N(a)}) = 1 \big\},$$

where each parity check function operates on the subset $N(a) \subset \{1,2,\ldots,m\}$ of bits. Viewed as an exponential family, the sufficient statistics are simply $\phi(x) = (x_1, x_2, \ldots, x_m)$, the base measure ν is counting measure restricted to the set $\mathbb{C}$. Consequently, the set $\mathcal{M}$ for this problem corresponds to the *codeword polytope* — namely, the convex hull of all possible codewords

$$\mathcal{M} = \mathrm{conv}\bigg\{ x \in \{0,1\}^m \mid \prod_{a \in F} \psi_a(x_{N(a)}) = 1 \bigg\}$$
$$= \mathrm{conv}\{ x \in \mathbb{C} \}. \tag{3.32}$$

To provide a concrete illustration, consider the code on $m = 3$ bits, defined by the single parity check relation $x_1 \oplus x_2 \oplus x_3 = 0$. Figure 3.7(a) shows the factor graph representation of this toy code. This parity check eliminates half of the $2^3 = 8$ possible binary sequences of length 3. The codeword polytope is simply the convex hull of $\{(0,0,0), (0,1,1), (1,0,1), (1,1,0)\}$, as illustrated in Figure 3.7(b), Equivalently, we can represent this codeword polytope in terms of half-space constraints. For this single parity check code, the codeword polytope is defined by the four inequalities

$$(1 - \mu_1) + (1 - \mu_2) + (1 - \mu_3) \geq 1, \tag{3.33a}$$

$$(1 - \mu_1) + \mu_2 + \mu_3 \geq 1, \tag{3.33b}$$

$$\mu_1 + (1 - \mu_2) + \mu_3 \geq 1, \quad \text{and} \tag{3.33c}$$

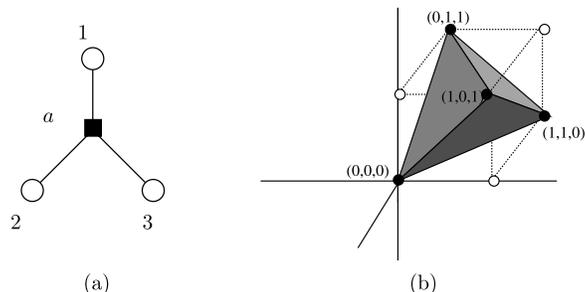$$\mu_1 + \mu_2 + (1 - \mu_3) \geq 1. \tag{3.33d}$$

(a)   (b)

Fig. 3.7 Illustration of $\mathcal{M}$ for the special case of the binary linear code $\mathbb{C} = \{(0,0,0),(0,1,1),(1,1,0),(1,1,0)\}$. This code is defined by a single parity check.

As we discuss at more length in Section 8.4.4, these inequalities can be understood as requiring that (μ_1, μ_2, μ_3) is at least distance 1 from each of the forbidden odd-parity vertices of the hypercube $\{0,1\}^3$.

   Of course, for larger code lengths m and many parity checks, the associated codeword polytope has a much more complicated structure. It plays a key role in the error-control decoding problem [78], studied intensively in the coding and information theory communities. Any binary linear code can be identified with a binary matroid [186], in which context the codeword polytope is referred to as the *cycle polytope* of the binary matroid. There is a rich literature in combinatorics on the structure of these codeword or cycle polytopes [9, 69, 104, 215].

   Examples 3.8 and 3.9 are specific instances of a more general object that we refer to as the *marginal polytope* for a discrete graphical model. The marginal polytope is defined for any graphical model with multinomial random variables $X_s \in \mathcal{X}_s := \{0, 1, \ldots, r_s - 1\}$ at each vertex $s \in V$; note that the cardinality $|\mathcal{X}_s| = r_s$ can differ from node to node. Consider the exponential family defined in terms of $\{0,1\}$-valued indicator functions

$$\forall s \in V, j \in \mathcal{X}_s, \qquad \mathbb{I}_{s;j}(x_s) := \begin{cases} 1 & \text{if } x_s = j \\ 0 & \text{otherwise.} \end{cases}$$

$$\forall (s,t) \in E, (j,k) \qquad \mathbb{I}_{st;jk}(x_s, x_t) := \begin{cases} 1 & \text{if } (x_s, x_t) = (j,k) \\ 0 & \text{otherwise.} \end{cases}$$

$$(3.34)$$

We refer to the sufficient statistics (3.34) as the *standard overcomplete representation*. Its overcompleteness was discussed previously in Example 3.2.

With this choice of sufficient statistics, the mean parameters take a very intuitive form: in particular, for each node $s \in V$

$$\mu_{s;j} = \mathbb{E}_p[\mathbb{I}_j(X_s)] = \mathbb{P}[X_s = j] \quad \forall j \in \mathcal{X}_s, \tag{3.35}$$

and for each edge $(s,t) \in E$, we have

$$\mu_{st;jk} = \mathbb{E}_p[\mathbb{I}_{st;jk}(X_s, X_t)] = \mathbb{P}[X_s = j, X_t = k] \quad \forall(j,k) \in \mathcal{X}_s \in \mathcal{X}_t. \tag{3.36}$$

Thus, the mean parameters correspond to singleton marginal distributions μ_s and pairwise marginal distributions μ_{st} associated with the nodes and edges of the graph. In this case, we refer to the set $\mathcal{M}$ as the *marginal polytope* associated with the graph, and denote it by $\mathbb{M}(G)$. Explicitly, it is given by

$$\mathbb{M}(G) := \{\mu \in \mathbb{R}^d \mid \exists p \text{ such that (3.35) holds } \forall(s;j), \text{ and}$$
$$(3.36) \text{ holds } \forall(st;jk\}. \tag{3.37}$$

Note that the correlation polytope for the Ising model presented in Example 3.8 is a special case of a marginal polytope, obtained for $X_s \in \{0,1\}$ for all nodes s. The only difference is we have defined marginal polytopes with respect to the standard overcomplete basis of indicator functions, whereas the Ising model is usually parameterized as a minimal exponential family. The codeword polytope of Example 3.9 is another special case of a marginal polytope. In this case, the reduction requires two steps: first, we convert the factor graph representation of the code — for instance, as shown in Figure 3.7(a) — to an equivalent pairwise Markov random field, involving binary variables at each bit node, and higher-order discrete variables at each factor node. (See Appendix E.3 for details of this procedure for converting from factor graphs to pairwise MRFs.) The marginal polytope associated with this pairwise MRF is simply a lifted version of the codeword polytope. We discuss these and other examples of marginal polytopes in more detail in later sections.

For the toy models considered explicitly in Examples 3.8 and 3.9, the number of half-space constraints $|\mathcal{J}|$ required to characterize the marginal polytopes was very small ($|\mathcal{J}| = 4$ in both cases). It is natural to ask how the number of constraints required grows as a function of the graph size. Interestingly, we will see later that this so-called facet complexity depends critically on the graph structure. For trees, any marginal polytope is characterized by local constraints — involving only pairs of random variables on edges — with the total number growing only linearly in the graph size. In sharp contrast, for general graphs with cycles, the constraints are very nonlocal and the growth in their number is astonishingly fast. For the special case of the Ising model, the book by Deza and Laurent [69] contains a wealth of information about the correlation/cut polytope. The intractability of representing marginal polytopes in a compact manner is one underlying cause of the complexity in performing statistical computation.

3.4.2   Role of Mean Parameters in Inference Problems

The preceding examples suggest that mean parameters have a central role to play in the marginalization problem. For the multivariate Gaussian (Example 3.7), an efficient algorithm for computing the mean parameterization provides us with both the Gaussian mean vector, as well as the associated covariance matrix. For the Ising model (see Example 3.8), the mean parameters completely specify the singleton and pairwise marginals of the probability distribution; the same statement holds more generally for the multinomial graphical model defined by the standard overcomplete parameterization (3.34). Even more broadly, the computation of the *forward mapping*, from the canonical parameters $\theta \in \Omega$ to the mean parameters $\mu \in \mathcal{M}$, can be viewed as a fundamental class of inference problems in exponential family models. Although computing the mapping is straightforward for most low-dimensional exponential families, the computation of this forward mapping is extremely difficult for many high-dimensional exponential families.

The *backward mapping*, namely from mean parameters $\mu \in \mathcal{M}$ to canonical parameters $\theta \in \Omega$, also has a natural statistical interpretation. In particular, suppose that we are given a set of samples $X_1^n := \{X^1, \ldots, X^n\}$, drawn independently from an exponential family

member $p_\theta(x)$, where the parameter θ is unknown. If the goal is to estimate θ, the classical principle of maximum likelihood dictates obtaining an estimate $\widehat{\theta}$ by maximizing the likelihood of the data, or equivalently (after taking logarithms and rescaling), maximizing the quantity

$$\ell(\theta; X_1^n) := \frac{1}{n} \sum_{i=1}^n \log p_\theta(X^i) = \langle \theta, \widehat{\mu} \rangle - A(\theta), \qquad (3.38)$$

where $\widehat{\mu} := \widehat{\mathbb{E}}[\phi(X)] = \frac{1}{n} \sum_{i=1}^n \phi(X^i)$ is the vector of *empirical mean parameters* defined by the data X_1^n. The maximum likelihood estimate $\widehat{\theta}$ is chosen to achieve the maximum of this objective function. Note that computing $\widehat{\theta}$ is, in general, another challenging problem, since the objective function involves the log partition function A. As will be demonstrated by the development to follow, under suitable conditions, the maximum likelihood estimate is unique, and specified by the stationarity condition $\mathbb{E}_{\widehat{\theta}}[\phi(X)] = \widehat{\mu}$. Finding the unique solution to this equation is equivalent to computing the backward mapping $\mu \mapsto \theta$, from mean parameters to canonical parameters. In general, computing this inverse mapping is also computationally intensive.

3.5   Properties of A

With this background in mind, we now turn to a deeper exploration of the properties of the cumulant function A. Perhaps the most important property of A is its convexity, which brings us into convex analysis, and more specifically leads us to conjugate duality. Under suitable conditions, the function A and its conjugate dual A^* — or more precisely, their derivatives — turn out to define a one-to-one and surjective mapping between the canonical and mean parameters. As discussed above, the mapping between canonical and mean parameters is a core challenge that underlies various statistical computations in high-dimensional graphical models.

3.5.1   Derivatives and Convexity

Recall that a real-valued function g is *convex* if, for any two x, y belonging to the domain of g and any scalar $\lambda \in (0, 1)$, the inequality

$$g(\lambda x + (1 - \lambda)y) \leq \lambda g(x) + (1 - \lambda)g(y) \qquad (3.39)$$

holds. In geometric terms, this inequality means that the line connecting the function values $g(x)$ and $g(y)$ lies above the graph of the function itself. The function is *strictly convex* if the inequality (3.39) is strict for all $x \neq y$. (See Appendix A.2.5 for some additional properties of convex functions.) We begin by establishing that the log partition function is smooth, and convex in terms of θ.

Proposition 3.1. The cumulant function

$$A(\theta) \; := \; \log \int_{\mathcal{X}^m} \exp\langle \theta, \phi(x) \rangle \, \nu(dx) \tag{3.40}$$

associated with any regular exponential family has the following properties:

(a) It has derivatives of all orders on its domain Ω. The first two derivatives yield the cumulants of the random vector $\phi(X)$ as follows:

$$\frac{\partial A}{\partial \theta_\alpha}(\theta) \; = \; \mathbb{E}_\theta[\phi_\alpha(X)] := \int \phi_\alpha(x) p_\theta(x) \nu(dx). \tag{3.41a}$$

$$\frac{\partial^2 A}{\partial \theta_\alpha \partial \theta_\beta}(\theta) \; = \; \mathbb{E}_\theta[\phi_\alpha(X)\phi_\beta(X)] - \mathbb{E}_\theta[\phi_\alpha(X)]\mathbb{E}_\theta[\phi_\beta(X)]. \tag{3.41b}$$

(b) Moreover, A is a convex function of θ on its domain Ω, and strictly so if the representation is minimal.

Proof. Let us assume that differentiating through the integral (3.40) defining A is permitted; verifying the validity of this assumption is a standard argument using the dominated convergence theorem (e.g., Brown [43]). Under this assumption, we have

$$\frac{\partial A}{\partial \theta_\alpha}(\theta) \; = \; \frac{\partial}{\partial \theta_\alpha}\left\{ \log \int_{\mathcal{X}^m} \exp\langle \theta, \phi(x)\rangle \, \nu(dx) \right\}$$

$$= \; \frac{\left[\int_{\mathcal{X}^m} \frac{\partial}{\partial \theta_\alpha} \exp\langle \theta, \phi(x)\rangle \, \nu(dx) \right]}{\left[\int_{\mathcal{X}^m} \exp\langle \theta, \phi(u)\rangle \, \nu(du) \right]}$$

$$= \int_{\mathcal{X}^m} \phi_\alpha(x) \frac{\exp\langle\theta, \phi(x)\rangle \, \nu(dx)}{\int_{\mathcal{X}^m} \exp\langle\theta, \phi(u)\rangle \, \nu(du)}$$

$$= \mathbb{E}_\theta[\phi_\alpha(X)],$$

which establishes Equation (3.41a). The formula for the higher-order derivatives can be proven in an entirely analogous manner.

Observe from Equation (3.41b) that the second-order partial derivative $\frac{\partial^2 A}{\partial\theta_\alpha\theta_\beta^2}$ is equal to the covariance element $\mathrm{cov}\{\phi_\alpha(X), \phi_\beta(X)\}$. Therefore, the full Hessian $\nabla^2 A(\theta)$ is the covariance matrix of the random vector $\phi(X)$, and so is positive semidefinite on the open set Ω, which ensures convexity (see Theorem 4.3.1 of Hiriart-Urruty and Lemaréchal [112]). If the representation is minimal, there is no nonzero vector $a \in \mathbb{R}^d$ and constant $b \in \mathbb{R}$ such that $\langle a, \phi(x)\rangle = b$ holds ν-a.e. This condition implies $\mathrm{var}_\theta[\langle a, \phi(x)\rangle] = a^T \nabla^2 A(\theta) a > 0$ for all $a \in \mathbb{R}^d$ and $\theta \in \Omega$; this strict positive definiteness of the Hessian on the open set Ω implies strict convexity [112]. $\square$

3.5.2   Forward Mapping to Mean Parameters

We now turn to an in-depth consideration of the forward mapping $\theta \mapsto \mu$, from the canonical parameters $\theta \in \Omega$ defining a distribution p_θ to its associated vector of mean parameters $\mu \in \mathbb{R}^d$. Note that the gradient ∇A can be viewed as mapping from Ω to $\mathbb{R}^d$. Indeed, Proposition 3.1 demonstrates that the range of this mapping is contained within the set $\mathcal{M}$ of realizable mean parameters, defined previously as

$$\mathcal{M} := \{\mu \in \mathbb{R}^d \mid \exists\, p \text{ s.t. } \mathbb{E}_p[\phi(X)] = \mu\}.$$

We will see that a great deal hinges on the answers to the following two questions:

   (a) when does ∇A define a one-to-one mapping?
   (b) when does the image of Ω under the mapping ∇A — that is, the set $\nabla A(\Omega)$ — fully cover the set $\mathcal{M}$?

The answer to the first question is relatively straightforward, essentially depending on whether or not the exponential family is minimal. The second question is somewhat more delicate: to begin, note that our

definition of $\mathcal{M}$ allows for mean parameters $\mu \in \mathbb{R}^d$ generated by *any* possible distribution, not just distributions p_θ in the exponential family defined by the sufficient statistics ϕ. It turns out that this extra freedom does not really enlarge the set $\mathcal{M}$; as Theorem 3.3 makes precise, under suitable conditions, all mean parameters in $\mathcal{M}$ can be realized by an exponential family distribution (or, for boundary points, by a limiting sequence of such distributions).

We begin with a result addressing the first question:

Proposition 3.2. The gradient mapping $\nabla A : \Omega \to \mathcal{M}$ is one-to-one if and only if the exponential representation is minimal.

Proof. If the representation is not minimal, then there must exist a nonzero vector $\gamma \in \mathbb{R}^d$ for which $\langle \gamma, \phi(x) \rangle$ is a constant (almost surely with respect to ν). Given any parameter $\theta^1 \in \Omega$, let us define another parameter vector $\theta^2 = \theta^1 + t\gamma$, where $t \in \mathbb{R}$. Since Ω is open, choosing t sufficiently small ensures that $\theta^2 \in \Omega$ as well. By the condition on the vector γ, the densities p_{θ^1} and p_{θ^2} induce the same probability distribution (only their normalization constants differ). For this pair, we have $\nabla A(\theta^1) = \nabla A(\theta^2)$, so that ∇A is not one-to-one.

Conversely, if the representation is minimal, then A is strictly convex by Proposition 3.1. For any strictly convex and differentiable function, we have $A(\theta^2) > A(\theta^1) + \langle \nabla A(\theta^1), \theta^2 - \theta^1 \rangle$, for all $\theta^1 \neq \theta^2$ in the domain Ω. The same inequality also holds with the roles of θ^1 and θ^2 reversed; adding together these inequalities yields that

$$\langle \nabla A(\theta^1) - \nabla A(\theta^2), \theta^1 - \theta^2 \rangle > 0$$

for all distinct $\theta^1, \theta^2 \in \Omega$, which shows that ∇A is one-to-one. $\qquad \square$

In general, although the gradient mapping ∇A is not one-to-one for an overcomplete representation, there is still a one-to-one correspondence between each element of $\nabla A(\Omega)$ and an affine subset of Ω. In particular, this affine subset contains all those canonical parameters θ that are mapped to the same mean parameter. For either a minimal or an overcomplete representation, we say that a pair (θ, μ) is *dually*

coupled if $\mu = \nabla A(\theta)$. This notion of dual coupling plays an important role in the sequel.

We now turn to the second issue regarding the image $\nabla A(\Omega)$ of the domain of valid canonical parameters Ω under the gradient mapping ∇A. Specifically, the goal is to determine for which mean parameter vectors $\mu \in \mathcal{M}$ does there exist a vector $\theta = \theta(\mu) \in \Omega$ such that $\mathbb{E}_\theta[\phi(X)] = \mu$. The solution turns out to be rather simple: the image $\nabla A(\Omega)$ is simply the interior $\mathcal{M}^\circ$. This fact is remarkable: it means that (disregarding boundary points) *all mean parameters $\mathcal{M}$ that are realizable by some distribution can be realized by a member of the exponential family.* To provide some intuition into this fact, consider the maximum entropy problem (3.3) for a given mean parameter μ in the interior of $\mathcal{M}$. As discussed earlier, when a solution to this problem exists, it necessarily takes the form of an exponential family member, say $p_{\theta(\mu)}$. Moreover, from the optimality conditions for the maximum entropy problem, this exponential family member must satisfy the moment-matching conditions $\mathbb{E}_{\theta(\mu)}[\phi(X)] = \mu$. Note that these moment-matching conditions are identical to those defining the maximum likelihood problem (3.38) — as we discuss in the following section, this fact is not coincidental, but rather a consequence of the primal–dual relationship between maximum entropy and maximum likelihood.

Theorem 3.3. In a minimal exponential family, the gradient map ∇A is onto the interior of $\mathcal{M}$, denoted by $\mathcal{M}^\circ$. Consequently, for each $\mu \in \mathcal{M}^\circ$, there exists some $\theta = \theta(\mu) \in \Omega$ such that $\mathbb{E}_\theta[\phi(X)] = \mu$.

We provide the proof of this result in Appendix B. In conjunction with Proposition 3.2, Theorem 3.3 guarantees that for minimal exponential families, each mean parameter $\mu \in \mathcal{M}^\circ$ is *uniquely realized* by some density $p_{\theta(\mu)}$ in the exponential family. However, a typical exponential family $\{p_\theta \mid \theta \in \Omega\}$ describes only a strict subset of all possible densities (with respect to the given base measure ν). In this case, there must exist at least some other density p — albeit not a member of an exponential family — that also realizes μ. The distinguishing property of the exponential distribution $p_{\theta(\mu)}$ is that, among the set of all distributions that realize μ, it has the maximum entropy. The connection

between A and the maximum entropy principle is specified precisely in terms of the conjugate dual function A^*, to which we now turn.

3.6   Conjugate Duality: Maximum Likelihood and Maximum Entropy

Conjugate duality is a cornerstone of convex analysis [112, 203], and is a natural source for variational representations. In this section, we explore the relationship between the log partition function A and its conjugate dual function A^*. This conjugate relationship is defined by a variational principle that is central to the remainder of this survey, in that it underlies a wide variety of known algorithms, both of an exact nature (e.g., the junction tree algorithm and its special cases of Kalman filtering, the forward–backward algorithm, peeling algorithms) and an approximate nature (e.g., sum-product on graphs with cycles, mean field, expectation-propagation, Kikuchi methods, linear programming, and semidefinite relaxations).

3.6.1   General Form of Conjugate Dual

Given a function A, the *conjugate dual function* to A, which we denote by A^*, is defined as follows:

$$A^*(\mu) := \sup_{\theta \in \Omega} \{\langle \mu, \theta \rangle - A(\theta)\}. \tag{3.42}$$

Here $\mu \in \mathbb{R}^d$ is a fixed vector of so-called dual variables of the same dimension as θ. Our choice of notation — i.e., using μ again — is deliberately suggestive, in that these dual variables turn out to have a natural interpretation as mean parameters. Indeed, we have already mentioned one statistical interpretation of this variational problem (3.42); in particular, the right-hand side is the optimized value of the rescaled log likelihood (3.38). Of course, this maximum likelihood problem only makes sense when the vector μ belongs to the set $\mathcal{M}$; an example is the vector of empirical moments $\widehat{\mu} = \frac{1}{n} \sum_{i=1}^{n} \phi(X^i)$ induced by a set of data $X_1^n = \{X^1, \ldots, X^n\}$. In our development, we consider the optimization problem (3.42) more broadly for any vector $\mu \in \mathbb{R}^d$. In this context, it is necessary to view A^* as a function taking values in the

extended real line $\mathbb{R}^* = \mathbb{R} \cup \{+\infty\}$, as is standard in convex analysis (see Appendix A.2.5 for more details).

As we have previously intimated, the conjugate dual function (3.42) is very closely connected to entropy. Recall the definition (3.2) of the Shannon entropy. The main result of the following theorem is that when $\mu \in \mathcal{M}^\circ$, the value of the dual function $A^*(\mu)$ is precisely the negative entropy of the exponential family distribution $p_{\theta(\mu)}$, where $\theta(\mu)$ is the unique vector of canonical parameters satisfying the relation

$$\mathbb{E}_{\theta(\mu)}[\phi(X)] = \nabla A(\theta(\mu)) = \mu. \tag{3.43}$$

We will also find it essential to consider $\mu \notin \mathcal{M}^\circ$, in which case it is impossible to find canonical parameters satisfying the relation (3.43). In this case, the behavior of the supremum defining $A^*(\mu)$ requires a more delicate analysis. In fact, denoting by $\overline{\mathcal{M}}$ the closure of $\mathcal{M}$, it turns out that whenever $\mu \notin \overline{\mathcal{M}}$, then $A^*(\mu) = +\infty$. This fact is essential in the use of variational methods: it guarantees that any optimization problem involving the dual function can be reduced to an optimization problem over $\mathcal{M}$. Accordingly, a great deal of our discussion in the sequel will be on the structure of $\mathcal{M}$ for various graphical models, and various approximations to $\mathcal{M}$ for models in which its structure is overly complex.

More formally, the following theorem, proved in Appendix B.2, provides a precise characterization of the relation between A and its conjugate dual A^*:

Theorem 3.4.

(a) For any $\mu \in \mathcal{M}^\circ$, denote by $\theta(\mu)$ the unique canonical parameter satisfying the dual matching condition (3.43). The conjugate dual function A^* takes the form

$$A^*(\mu) = \begin{cases} -H(p_{\theta(\mu)}) & \text{if } \mu \in \mathcal{M}^\circ \\ +\infty & \text{if } \mu \notin \overline{\mathcal{M}}. \end{cases} \tag{3.44}$$

For any boundary point $\mu \in \overline{\mathcal{M}} \backslash \mathcal{M}^\circ$ we have $A^*(\mu) = \lim_{n \to +\infty} A^*(\mu^n)$ taken over any sequence $\{\mu^n\} \subset \mathcal{M}^\circ$ converging to μ.

(b) In terms of this dual, the log partition function has the variational representation

$$A(\theta) = \sup_{\mu \in \mathcal{M}} \big\{ \langle \theta, \mu \rangle - A^*(\mu) \big\}. \qquad (3.45)$$

(c) For all $\theta \in \Omega$, the supremum in Equation (3.45) is attained uniquely at the vector $\mu \in \mathcal{M}^\circ$ specified by the moment-matching conditions

$$\mu = \int_{\mathcal{X}^m} \phi(x) p_\theta(x) \nu(dx) = \mathbb{E}_\theta[\phi(X)]. \qquad (3.46)$$

Theorem 3.4(a) provides a precise statement of the duality between the cumulant function A and entropy. A few comments on this relationship are in order. First, it is important to recognize that A^* is a slightly different object than the usual entropy (3.2): whereas the entropy maps density functions to real numbers (and so is a functional), the dual function A^* is an extended real-valued function on $\mathbb{R}^d$, finite only for valid mean parameters $\mu \in \mathcal{M}$.

Second, the value $-A^*(\mu)$ corresponds to the optimum of the maximum entropy problem (3.3), where $\mu \in \mathbb{R}^d$ parameterizes the constraint set. The event $A^*(\mu) = +\infty$ corresponds to infeasibility of the maximum entropy problem. This is an important point. Constrained optimization problems are defined both by the set being optimized over and the function being optimized. Given that the variational representation of the cumulant function in (3.45) takes the form of a maximization problem, we see that vectors μ for which $-A^*(\mu) = -\infty$ can certainly not be optima. Thus, it suffices to maximize over the set $\mathcal{M}$ instead of $\mathbb{R}^d$, as expressed in the variational representation (3.45). This fact implies that the nature of the set $\mathcal{M}$ plays a critical role in determining the complexity of computing the cumulant function.

Third, Theorem 3.4 also clarifies the precise nature of the bijection between the sets Ω and $\mathcal{M}^\circ$, which holds for any minimal exponential family. In particular, the gradient mapping ∇A maps Ω in a one-to-one manner onto $\mathcal{M}^\circ$, whereas the inverse mapping from $\mathcal{M}^\circ$ to Ω is given by the gradient ∇A^* of the dual function (see Appendix B.3 for more details). Figure 3.8 provides an idealized illustration of this bijective correspondence based on the gradient mappings $(\nabla A, \nabla A^*)$.
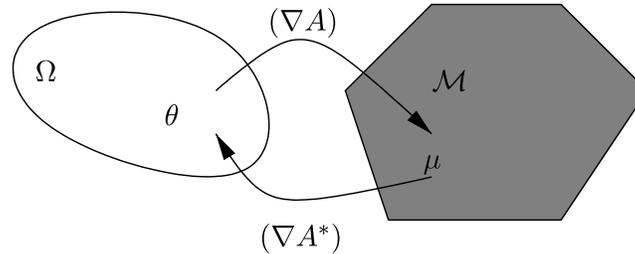
Fig. 3.8 Idealized illustration of the relation between the set Ω of valid canonical parameters, and the set $\mathcal{M}$ of valid mean parameters. The gradient mappings ∇A and ∇A^* associated with the conjugate dual pair (A, A^*) provide a bijective mapping between Ω and the interior $\mathcal{M}^\circ$.

3.6.2 Some Simple Examples

Theorem 3.4 is best understood by working through some simple examples. Table 3.2 provides the conjugate dual pair (A, A^*) for several well-known exponential families of scalar random variables. For each family, the table also lists $\Omega := \operatorname{dom} A$, as well as the set $\mathcal{M}$, which contains the effective domain of A^*, corresponding to the set of values for which A^* is finite.

In the rest of this section, we illustrate the basic ideas by working through two simple scalar examples in detail. To be clear, neither of these examples is interesting from a computational perspective — indeed, for most scalar exponential families, it is trivial to compute the mapping between canonical and mean parameters by direct methods. Nonetheless, they are useful in building intuition for the consequences of Theorem 3.4. The reader interested only in the main thread may skip ahead to Section 3.7, where we resume our discussion of the role of Theorem 3.4 in the derivation of approximate inference algorithms for multivariate exponential families.

Example 3.10 (Conjugate Duality for Bernoulli). Consider a Bernoulli variable $X \in \{0, 1\}$: its distribution can be written as an exponential family with $\phi(x) = x$, $A(\theta) = \log(1 + \exp(\theta))$, and $\Omega = \mathbb{R}$. In order to verify the claim in Theorem 3.4(a), let us compute the conjugate dual function A^* by direct methods. By the definition of conjugate

Table 3.2 Conjugate dual relations of Theorem 3.4 for several well-known exponential families of scalar variables.

Family	Ω	$A(\theta)$	$\mathcal{M}$	$A^*(\mu)$
Bernoulli	$\mathbb{R}$	$\log[1 + \exp(\theta)]$	$[0,1]$	$\mu\log\mu + (1-\mu)\log(1-\mu)$
Gaussian Location family	$\mathbb{R}$	$\dfrac{1}{2}\theta^2$	$\mathbb{R}$	$\dfrac{1}{2}\mu^2$
Gaussian Location-scale	$\{(\theta_1,\theta_2)\mid\theta_2<0\}$	$-\dfrac{\theta_1^2}{4\theta_2} - \log(-2\theta_2)$	$\{(\mu_1,\mu_2)\mid\mu_2 - (\mu_1)^2 > 0\}$	$-\dfrac{1}{2}\log[\mu_2 - \mu_1^2]$
Exponential	$(-\infty,0)$	$-\log(-\theta)$	$(0,+\infty)$	$-1-\log(\mu)$
Poisson	$\mathbb{R}$	$\exp(\theta)$	$(0,+\infty)$	$\mu\log\mu - \mu$

duality (3.42), for any fixed $\mu \in \mathbb{R}$, we have

$$A^*(\mu) = \sup_{\theta \in \mathbb{R}} \{\theta \cdot \mu - \log(1 + \exp(\theta))\}. \qquad (3.47)$$

Taking derivatives yields the stationary condition $\mu = \exp(\theta)/[1 + \exp(\theta)]$, which is simply the moment matching condition (3.43) specialized to the Bernoulli model. When is there a solution to this stationary condition? If $\mu \in (0,1)$, then some simple algebra shows that we may rearrange to find the unique solution $\theta(\mu) := \log[\mu/(1 - \mu)]$. Since $\mathcal{M}^\circ = (0,1)$ for the Bernoulli model, the existence and uniqueness of this solution are particular consequences of Proposition 3.2 and Theorem 3.3. Since the objective function (3.47) is strictly convex, the solution $\theta(\mu)$ specifies the optimum; substituting $\theta(\mu)$ into the objective Equation (3.47) and simplifying yields that

$$A^*(\mu) = \mu \log[\mu/(1 - \mu)] - \log\left[1 + \frac{\mu}{1 - \mu}\right]$$

$$= \mu \log \mu + (1 - \mu) \log(1 - \mu),$$

which is the negative entropy of Bernoulli variate with mean parameter μ. We have thus verified Theorem 3.4(a) in the case that $\mu \in (0,1) = \mathcal{M}^\circ$.

Now, let us consider the case $\mu \notin \overline{\mathcal{M}} = [0,1]$; concretely, let us suppose that $\mu > 1$. In this case, there is no gradient stationary point in the optimization problem (3.47). Therefore, the supremum is specified by the limiting behavior as $\theta \to \pm\infty$. For $\mu > 1$, we claim that the objective function grows unboundedly as $\theta \to +\infty$. Indeed, by the convexity of A, we have $A(0) = \log 2 \geq A(\theta) + A'(\theta)(-\theta)$. Moreover $A'(\theta) \leq 1$ for all $\theta \in \mathbb{R}$, from which we obtain the upper bound $A(\theta) \leq \theta + \log 2$, valid for $\theta > 0$. Consequently, for $\theta > 0$, we have

$$\mu \cdot \theta - \log[1 + \exp(\theta)] \geq (\mu - 1)\theta - \log 2,$$

showing that the objective function diverges as $\theta \to +\infty$, whenever $\mu > 1$. A similar calculation establishes the same claim for $\mu < 0$, showing that $A^*(\mu) = +\infty$ for $\mu \notin \overline{\mathcal{M}}$ and thus verifying the second part of Theorem 3.4(a). Finally, for the boundary points $\mu = 0$ and $\mu = 1$, it can be verified by taking limits that $A^*(0) = A^*(1) = 0$ for the Bernoulli model.

Turning to the verification of Theorem 3.4(b), since $A^*(\mu) = +\infty$ unless $\mu \in [0,1]$, the optimization problem (3.45) reduces to

$$\max_{\mu \in [0,1]} \{\mu \cdot \theta - \mu \log \mu - (1 - \mu) \log(1 - \mu)\}.$$

Explicitly solving this concave maximization problem yields that its optimal value is $\log[1 + \exp(\theta)]$, which verifies the claim (3.45). Moreover, this same calculation shows that the optimum is attained uniquely at $\mu(\theta) = \exp(\theta)/[1 + \exp(\theta)]$, which verifies Theorem 3.4(c) for the Bernoulli model.

Example 3.11 (Conjugate Duality for Exponential). Consider the family of exponential distributions, represented as a regular exponential family with $\phi(x) = x$, $\Omega = (-\infty, 0)$, and $A(\theta) = -\log(-\theta)$. From the conjugate dual definition (3.42), we have

$$A^*(\mu) = \sup_{\theta < 0} \{\mu \cdot \theta + \log(-\theta)\}. \tag{3.48}$$

In order to find the optimum, we take derivatives with respect to θ. We thus obtain the stationary condition $\mu = -1/\theta$, which corresponds to the moment matching condition (3.43) specialized to this family. Hence, for all $\mu \in \mathcal{M}^\circ = (0, \infty)$, we have $A^*(\mu) = -1 - \log(\mu)$. It can be verified that the entropy of an exponential distribution with mean $\mu > 0$ is given by $-A^*(\mu)$. This result confirms Theorem 3.4 for $\mu \in \mathcal{M}^\circ$. For $\mu \notin \overline{\mathcal{M}}$ — that is, for $\mu < 0$ — we see by inspection that the objective function $\mu\theta + \log(-\theta)$ grows unboundedly as $\theta \to -\infty$, thereby demonstrating that $A^*(\mu) = +\infty$ for all $\mu \notin \overline{\mathcal{M}}$. The remaining case is the boundary point $\mu = 0$, for which we have $A^*(0) = +\infty$ from the definition (3.48). Note also that the negative entropy of the exponential distribution $-1 - \log(\mu^n)$ tends to infinity for all sequence $\mu^n \to 0^+$, consistent with Theorem 3.4(a). Having computed the dual A^*, straightforward algebra shows that

$$A(\theta) = -\log(-\theta) = \sup_{\mu > 0} \{\mu \cdot \theta + 1 + \log(\mu)\},$$

with the optimum uniquely attained at $\mu = -1/\theta$. This calculation verifies parts (b) and (c) of Theorem 3.4 for the exponential variate.

3.7 Computational Challenges with High-Dimensional Models

From a computational perspective, the essential features of Theorem 3.4 are the representation (3.45) of the cumulant function, and the assertion (3.46) that the optimum is uniquely achieved at the mean parameters $\mu = \mathbb{E}_\theta[\phi(X)]$. It thus illuminates a key property of computing the cumulant function A, as well as the mean parameters μ: in principle, we can *compute both of these quantities by solving the variational problem* (3.45). Even more encouragingly, at least from a superficial perspective, the optimization problem appears to be rather simple: it is a finite-dimensional optimization problem over a convex set, and the objective function is strictly concave and differentiable. Thus, the optimization lacks local optima or other unpleasant features. It is tempting, then, to assert that the problem of computing the log partition function and the associated mean parameters is now solved, since we have "reduced" it to a convex optimization problem.

In this context, the simple scalar examples of Table 3.2, for which the fundamental variational problem (3.45) had an explicit form and could be solved easily, are very misleading. For general multivariate exponential families, in contrast, there are two primary challenges associated with the variational representation:

(a) In many cases, the constraint set $\mathcal{M}$ of realizable mean parameters is extremely difficult to characterize in an explicit manner.

(b) The negative entropy function A^* is defined indirectly — in a variational manner — so that it too typically lacks an explicit form.

For instance, to illustrate issue (a) concerning the nature of $\mathcal{M}$, for Markov random fields involving discrete random variables $X \in \{0, 1, \ldots, r-1\}^m$, the set $\mathcal{M}$ is always a polytope, which we have referred to as a *marginal polytope*. In this case, at least in principle, the set $\mathcal{M}$ can be characterized by some finite number of linear inequalities. However, for general graphs, the number of such inequalities grows

$$\mu \longrightarrow \boxed{(\nabla A)^{-1}} \xrightarrow{\theta(\mu)} \boxed{-H(p_{\theta(\mu)})} \longrightarrow A^*(\mu)$$
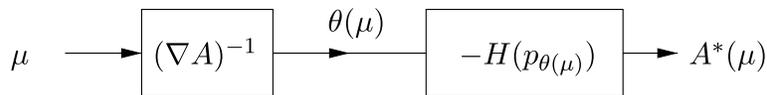
Fig. 3.9 A block diagram decomposition of A^* as the composition of two functions. Any mean parameter $\mu \in \mathcal{M}^\circ$ is first mapped back to a canonical parameter $\theta(\mu)$ in the inverse image $(\nabla A)^{-1}(\mu)$. The value of $A^*(\mu)$ corresponds to the negative entropy $-H(p_{\theta(\mu)})$ of the associated exponential family density $p_{\theta(\mu)}$.

rapidly with the graph size. Indeed, unless fundamental conjectures in complexity theory turn out to be false, it is not even possible to optimize a linear function over $\mathcal{M}$ for a general discrete MRF. In addition to the complexity of the constraint set, issue (b) highlights that even evaluating the cost function at a single point $\mu \in \mathcal{M}$, let alone optimizing it over $\mathcal{M}$, is extremely difficult.

To understand the complexity inherent in evaluating the dual value $A^*(\mu)$, note that Theorem 3.4 provides only an implicit characterization of A^* as the composition of mappings: first, the inverse mapping $(\nabla A)^{-1} : \mathcal{M}^\circ \to \Omega$, in which μ maps to $\theta(\mu)$, corresponding to the exponential family member with mean parameters μ; and second, the mapping from $\theta(\mu)$ to the negative entropy $-H(p_{\theta(\mu)})$ of the associated exponential family density. This decomposition of the value $A^*(\mu)$ is illustrated in Figure 3.9. Consequently, computing the dual value $A^*(\mu)$ at some point $\mu \in \mathcal{M}^\circ$ requires computing the inverse mapping $(\nabla A)^{-1}(\mu)$, in itself a nontrivial problem, and then evaluating the entropy, which requires high-dimensional integration for general graphical models. These difficulties motivate the use of approximations to $\mathcal{M}$ and A^*. Indeed, as shown in the sections to follow, a broad class of methods for approximate marginalization are based on this strategy of finding an approximation to the exact variational principle, which is then often solved using some form of message-passing algorithm.

4

Sum-Product, Bethe–Kikuchi, and Expectation-Propagation

In this section, we begin our exploration of the variational interpretation of message-passing algorithms. We first discuss the sum-product algorithm, also known as the belief propagation algorithm. As discussed in Section 2.5, sum-product is an exact algorithm for tree-structured graphs, for which it can be derived as a divide-and-conquer algorithm. However, given the local form of the sum-product updates, there is no barrier to applying it to a graph with cycles, which yields the "loopy" form of the sum-product or belief propagation algorithm. In the presence of cycles, there are no general convergence or correctness guarantees associated with the sum-product algorithm, but it is nonetheless widely used to compute approximate marginals. The first part of this section describes the variational formulation of the sum-product updates in terms of the Bethe approximation. Although the approximation itself dates back to the work of Bethe [29], the connection to the sum-product algorithm was first elucidated by Yedidia et al. [268, 269]. We then describe various natural generalizations of the Bethe approximation, including Kikuchi clustering and other hypergraph-based methods [133, 269]. Finally, we describe expectation-propagation algorithms [175, 184] and related

moment-matching methods [64, 115, 185]; these are also variational methods based on Bethe-like approximations.

4.1   Sum-Product and Bethe Approximation

The simplest instantiation of the Bethe approximation applies to an undirected graphical model $G = (V, E)$ with potential functions involving at most pairs of variables; we refer to any such model as a *pairwise Markov random field.*[1] The sum-product algorithm is most widely used for the cases of discrete random variables, or Gaussian random variables. The bulk of our discussion focuses on the former case, where for each node $s \in V$, the associated variable X_s takes values in a discrete space $\mathcal{X}_s = \{0, 1, \ldots, r_s - 1\}$. *(log normalizer as a sup over means)*

In the discrete case, the general variational principle (3.45) takes different forms, depending on the chosen form of sufficient statistics ϕ. Recall from Section 3.4.1 our definition of the canonical overcomplete representation (3.34), using indicator functions for events $\{X_s = j\}$ and $\{X_s = j, X_t = k\}$. Using these sufficient statistics, we define an exponential family of the form

$$p_\theta(x) \propto \exp\Big\{ \sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t) \Big\}, \tag{4.1}$$

where we have introduced the convenient shorthand notation

$$\theta_s(x_s) := \sum_j \theta_{s;j} \mathbb{I}_{s;j}(x_s), \quad \text{and} \tag{4.2a}$$

$$\theta_{st}(x_s, x_t) := \sum_{(j,k)} \theta_{st;jk} \mathbb{I}_{st;jk}(x_s, x_t). \tag{4.2b}$$

As previously discussed in Example 3.2, this particular parameterization is overcomplete or nonidentifiable, because there exist entire affine subsets of canonical parameters that induce the same probability

---

[1] In principle, by selectively introducing auxiliary variables, any undirected graphical model can be converted into an equivalent pairwise form to which the Bethe approximation can be applied; see Appendix E.3 for details of this procedure. It can also be useful to treat higher order interactions directly, which can be done using the approximations discussed in Section 4.2.

distribution[2] over the random vector X. Nonetheless, this overcompleteness is useful because the associated mean parameters are easily interpretable, since they correspond to singleton and pairwise marginal probabilities (see Equations (3.35) and (3.36)). Using these mean parameters, it is convenient for various calculations to define the shorthand notations

$$\mathbb{P}(X_s = x_s) \qquad \mu_s(x_s) := \sum_{j \in \mathcal{X}_s} \mu_{s;j} \mathbb{I}_{s;j}(x_s), \quad \text{and} \tag{4.3a}$$

$$\mathbb{P}(X_s = x_s, X_t = x_t) \quad \mu_{st}(x_s, x_t) := \sum_{(j,k) \in \mathcal{X}_s \times \mathcal{X}_t} \mu_{st;jk} \mathbb{I}_{st;jk}(x_s, x_t). \tag{4.3b}$$

Note that μ_s is a $|\mathcal{X}_s|$-dimensional marginal distribution over X_s, whereas μ_{st} is a $|\mathcal{X}_s| \times |\mathcal{X}_t|$ matrix, representing a joint marginal over (X_s, X_t). The *marginal polytope* $\mathbb{M}(G)$ corresponds to the set of all singleton and pairwise marginals that are jointly realizable by some distribution p—namely

$$\mathbb{M}(G) := \{\mu \in \mathbb{R}^d \mid \exists\, p \text{ with marginals } \mu_s(x_s),\, \mu_{st}(x_s, x_t)\}. \tag{4.4}$$

In the case of discrete (multinomial) random variables, this set plays a central role in the general variational principle from Theorem 3.4.

4.1.1   A Tree-Based Outer Bound to $\mathbb{M}(G)$

As previously discussed in Section 3.4.1, the polytope $\mathbb{M}(G)$ can either be written as the convex hull of a finite number of vectors, one associated with each configuration $x \in \mathcal{X}^m$, or alternatively, as the intersection of a finite number of half-spaces (see Figure 3.5 for an illustration). But how to provide an explicit listing of these half-space constraints, also known as facets? In general, this problem is extremely difficult, so that we resort to listing only subsets of the constraints, thereby obtaining a polyhedral outer bound on $\mathbb{M}(G)$.

---

[2] A little calculation shows that the exponential family (4.1) has dimension $d = \sum_{s \in V} r_s + \sum_{(s,t) \in E} r_s r_t$. Given some fixed parameter vector $\theta \in \mathbb{R}^d$, suppose that we form a new canonical parameter $\theta' \in \mathbb{R}^d$ by setting $\theta'_{1;j} = \theta_{1;j} + C$ for all $j \in \mathcal{X}$, where $C \in \mathbb{R}$ is some fixed constant, and $\theta'_\alpha = \theta_\alpha$ for all other indices α. It is then straightforward to verify that $p_\theta(x) = p_{\theta'}(x)$ for all $x \in \mathcal{X}^m$, so that θ and θ' describe the same probability distribution.

More specifically, consider a trial set of single node functions $\{\tau_s, s \in V\}$ and edge-based functions $\{\tau_{st}, (s,t) \in E\}$. If these trial marginal distributions are to be globally realizable, they must of course be nonnegative, and in addition, each singleton quantity τ_s must satisfy the *normalization condition*

$$\sum_{x_s} \tau_s(x_s) = 1. \tag{4.5}$$

Moreover, for each edge $(s,t) \in E$, the singleton $\{\tau_s, \tau_t\}$ and pairwise quantities τ_{st} must satisfy the *marginalization constraints*

$$\sum_{x_t'} \tau_{st}(x_s, x_t') = \tau_s(x_s), \quad \forall\, x_s \in \mathcal{X}_s, \quad \text{and} \tag{4.6a}$$

$$\sum_{x_s'} \tau_{st}(x_s', x_t) = \tau_t(x_t), \quad \forall\, x_t \in \mathcal{X}_t. \tag{4.6b}$$

These constraints define the set

$$\mathbb{L}(G) := \{\tau \geq 0 \mid \text{condition (4.5) holds for } s \in V, \quad \text{and}$$

$$\text{condition (4.6) holds for } (s,t) \in E\}, \tag{4.7}$$

of *locally consistent* marginal distributions. Note that $\mathbb{L}(G)$ is also a polytope, in fact a very simple one, since it is defined by $\mathcal{O}(|V| + |E|)$ constraints in total.

What is the relation between the set $\mathbb{L}(G)$ of locally consistent marginals and the set $\mathbb{M}(G)$ of globally realizable marginals? On one hand, it is clear that $\mathbb{M}(G)$ is always a subset of $\mathbb{L}(G)$, since any set of globally realizable marginals must satisfy the normalization (4.5) and marginalization (4.6) constraints. Apart from this inclusion relation, it turns out that for a graph with cycles, these two sets are rather different. However, for a tree T, the junction tree theorem, in the form of Proposition 2.1, guarantees that they are equivalent, as summarized in the following proposition.

Proposition 4.1. The inclusion $\mathbb{M}(G) \subseteq \mathbb{L}(G)$ holds for any graph. For a tree-structured graph T, the marginal polytope $\mathbb{M}(T)$ is equal to $\mathbb{L}(T)$.

Proof. Consider an element μ of the full marginal polytope $\mathbb{M}(G)$: clearly, any such vector must satisfy the normalization and pairwise marginalization conditions defining the set $\mathbb{L}(G)$, from which we conclude that $\mathbb{M}(G) \subseteq \mathbb{L}(G)$. In order to demonstrate the reverse inclusion for a tree-structured graph T, let μ be an arbitrary element of $\mathbb{L}(T)$; we need to show that $\mu \in \mathbb{M}(T)$. By definition of $\mathbb{L}(T)$, the vector μ specifies a set of locally consistent singleton marginals μ_s for vertices $s \in V$ and pairwise marginals μ_{st} for edges $(s,t) \in E$. By the junction tree theorem, we may use them to form a distribution, Markov with respect to the tree, as follows:

$$p_\mu(x) := \prod_{s \in V} \mu_s(x_s) \prod_{(s,t) \in E} \frac{\mu_{st}(x_s, x_t)}{\mu_s(x_s)\mu_t(x_t)}. \tag{4.8}$$

(We take $0/0 := 0$ in cases of zeros in the elements of μ.) It is a consequence of the junction tree theorem or can be verified directly via an inductive "leaf-stripping" argument that with this choice of p_μ, we have $\mathbb{E}_{p_\mu}[\mathbb{I}_j(X_s)] = \mu_s(x_s)$ for all $s \in V$ and $j \in \mathcal{X}_s$, as well as $\mathbb{E}_{p_\mu}[\mathbb{I}_{jk}(X_s, X_t)] = \mu_{st}(x_s, x_t)$ for all $(s,t) \in E$, and $(j,k) \in \mathcal{X}_s \times \mathcal{X}_t$. Therefore, the distribution (4.8) provides a constructive certificate of the membership $\mu \in \mathbb{M}(T)$, which establishes that $\mathbb{L}(T) = \mathbb{M}(T)$. $\qquad\square$

For a graph G with cycles, in sharp contrast to the tree case, the set $\mathbb{L}(G)$ is a strict outer bound on $\mathbb{M}(G)$, in that there exist vectors $\tau \in \mathbb{L}(G)$ that do *not* belong to $\mathbb{M}(G)$, for which reason we refer to members τ of $\mathbb{L}(G)$ as *pseudomarginals*. The following example illustrates the distinction between globally realizable marginals and pseudomarginals.

Example 4.1 ($\mathbb{L}(G)$ versus $\mathbb{M}(G)$). Let us explore the relation between the two sets on the simplest graph for which they fail to be equivalent — namely, the single cycle on three vertices, denoted by C_3. Considering the binary random vector $X \in \{0,1\}^3$, note that each singleton pseudomarginal τ_s, for $s = 1,2,3$, can be viewed as a 1×2 vector, whereas each pairwise pseudomarginal τ_{st}, for edges $(s,t) \in \{(12), (13), (23)\}$ can be viewed as a 2×2 matrix. We define

the family of pseudomarginals

$$\tau_s(x_s) := \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}, \quad \text{and} \tag{4.9a}$$

$$\tau_{st}(x_s, x_t) := \begin{bmatrix} \beta_{st} & 0.5 - \beta_{st} \\ 0.5 - \beta_{st} & \beta_{st} \end{bmatrix}, \tag{4.9b}$$

where for each edge $(s,t) \in E$, the quantity $\beta_{st} \in \mathbb{R}$ is a parameter to be specified.

We first observe that for any $\beta_{st} \in [0, 0.5]$, these pseudomarginals satisfy the normalization (4.5) and marginalization constraints (4.6), so the associated pseudomarginals (4.9) belong to $\mathbb{L}(C_3)$. As a particular choice, consider the collection τ of pseudomarginals generated by setting $\beta_{12} = \beta_{23} = 0.4$, and $\beta_{13} = 0.1$, as illustrated in Figure 4.1(a). With these settings, the vector τ is an element of $\mathbb{L}(C_3)$; however, as a candidate set of global marginal distributions, certain features of the collection τ should be suspicious. In particular, according to the putative marginals τ, the events $\{X_1 = X_2\}$ and $\{X_2 = X_3\}$ should each hold with probability 0.8, whereas the event $\{X_1 = X_3\}$ should only hold with probability 0.2. At least intuitively, this set-up appears likely to violate some type of global constraint.

In order to prove the global invalidity of τ, we first specify the constraints that actually define the marginal polytope $\mathbb{M}(G)$. For ease of
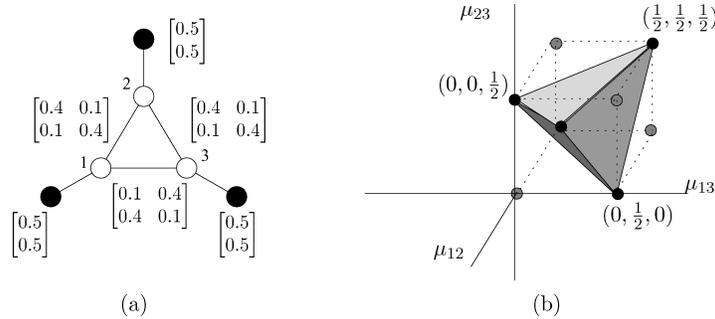
Fig. 4.1 (a) A set of pseudomarginals associated with the nodes and edges of the graph: setting $\beta_{12} = \beta_{23} = 0.4$ and $\beta_{13} = 0.1$ in Equation (4.9) yields a pseudomarginal vector τ which, though locally consistent, is not globally consistent. (b) Marginal polytope $\mathbb{M}(C_3)$ for the three node cycle; in a minimal exponential representation, it is a 6D object. Illustrated here is the slice $\{\mu_1 = \mu_2 = \mu_3 = \frac{1}{2}\}$, as well as the outer bound $\mathbb{L}(C_3)$, also for this particular slice.

illustration, let us consider the slice of the marginal polytope defined by the constraints $\mu_1 = \mu_2 = \mu_3 = \frac{1}{2}$. Viewed in this slice, the constraints defining $\mathbb{L}(G)$ reduce to $0 \leq \beta_{st} \leq \frac{1}{2}$ for all edges (s,t), so that the set is simply a 3D cube, as drawn with dotted lines in Figure 4.1(b). It can be shown that the sliced version of $\mathbb{M}(G)$ is defined by these box constraints, and in addition the following cycle inequalities

$$\mu_{12} + \mu_{23} - \mu_{13} \leq \frac{1}{2}, \quad \mu_{12} - \mu_{23} + \mu_{13} \leq \frac{1}{2}, \tag{4.10a}$$

$$-\mu_{12} + \mu_{23} + \mu_{13} \leq \frac{1}{2}, \quad \text{and } \mu_{12} + \mu_{23} + \mu_{13} \geq \frac{1}{2}. \tag{4.10b}$$

(See Example 8.7 in the sequel for the derivation of these cycle inequalities.) As illustrated by the shaded region in Figure 4.1(b), the marginal polytope $\mathbb{M}(C_3)$ is strictly contained within the scaled cube $[0, \frac{1}{2}]^3$.

To conclude the discussion of our example, note that the pseudo-marginal vector τ specified by $\beta_{12} = \beta_{23} = 0.4$ and $\beta_{13} = 0.1$ fails to satisfy the cycle inequalities (4.10), since in particular, we have the violation

$$\beta_{12} + \beta_{23} - \beta_{13} = 0.4 + 0.4 - 0.1 > \frac{1}{2}.$$

Therefore, the vector τ is an instance of a set of pseudomarginals valid under the Bethe approximation, but which could never arise from a true probability distribution.

4.1.2 Bethe Entropy Approximation

We now turn to the second variational ingredient that underlies the sum-product algorithm, namely an approximation to the dual function (or negative entropy). As with the outer bound $\mathbb{L}(G)$ on the marginal polytope, the Bethe entropy approximation is also tree-based.

For a general MRF based on a graph with cycles, the negative entropy A^* — as a function of *only* the mean parameters μ — typically lacks a closed form expression. An important exception to this rule is the case of a tree-structured Markov random field, for which the entropy decomposes in terms of local entropies associated with the edges and nodes of the graph. In order to derive this decomposition,

recall from the proof of Proposition 4.1 the factorization (4.8) of any tree-structured MRF distribution in terms of marginal distributions $\{\mu_s, s \in V\}$ and $\{\mu_{st}, (s,t) \in E\}$ on the node and edges, respectively, of the tree. These marginal distributions correspond to the mean parameters under the canonical overcomplete sufficient statistics (3.34). Thus, for a tree-structured MRF, we can compute the (negative) dual value $-A^*(\mu)$ directly, simply by computing the entropy $H(p_\mu)$ of the distribution (4.8). Denoting by $\mathbb{E}_\mu$ the expectation under the distribution (4.8), we obtain

$$
\begin{aligned}
H(p_\mu) = -A^*(\mu) &= \mathbb{E}_\mu[-\log p_\mu(X)] \\
&= \sum_{s \in V} H_s(\mu_s) - \sum_{(s,t) \in E} I_{st}(\mu_{st}).
\end{aligned} \tag{4.11}
$$

The different terms in this expansion are the singleton entropy

$$
H_s(\mu_s) := - \sum_{x_s \in \mathcal{X}_s} \mu_s(x_s) \log \mu_s(x_s) \tag{4.12}
$$

for each node $s \in V$, and the mutual information

$$
I_{st}(\mu_{st}) := \sum_{(x_s, x_t) \in \mathcal{X}_s \times \mathcal{X}_t} \mu_{st}(x_s, x_t) \log \frac{\mu_{st}(x_s, x_t)}{\mu_s(x_s) \mu_t(x_t)} \tag{4.13}
$$

for each edge $(s,t) \in E$. Consequently, for a tree-structured graph, the dual function A^* can be expressed as an explicit and easily computable function of the mean parameters μ.

With this background, the Bethe approximation to the entropy of an MRF with cycles is easily described: it simply assumes that decomposition (4.11) is approximately valid for a graph with cycles. This assumption yields the *Bethe entropy approximation*

$$
-A^*(\tau) \approx H_{\text{Bethe}}(\tau) := \sum_{s \in V} H_s(\tau_s) - \sum_{(s,t) \in E} I_{st}(\tau_{st}). \tag{4.14}
$$

An important fact, central in the derivation of the sum-product algorithm, is that this approximation (4.14) can be evaluated for any set of pseudomarginals $\{\tau_s, s \in V\}$ and $\{\tau_{st}, (s,t) \in E\}$ that belong to $\mathbb{L}(G)$. For this reason, our change in notation — from μ for exact marginals to τ for pseudomarginals — is deliberate.

We note in passing that Yedidia et al. [268, 269] used an alternative form of the Bethe entropy approximation (4.14), one which can be obtained via the relation $I_{st}(\tau_{st}) = H_s(\tau_s) + H_t(\tau_t) - H_{st}(\tau_{st})$, where H_{st} is the joint entropy defined by the pseudomarginal τ_{st}. Doing so and performing some algebraic manipulation yields

$$H_{\text{Bethe}}(\tau) = -\sum_{s \in V} (d_s - 1) H_s(\tau_s) + \sum_{(s,t) \in E} H_{st}(\tau_{st}), \quad (4.15)$$

where d_s corresponds to the number of neighbors of node s (i.e., the degree of node s). However, the symmetric form (4.14) turns out to be most natural for our development in the sequel.

4.1.3 Bethe Variational Problem and Sum-Product

We now have the two ingredients needed to construct the Bethe approximation to the exact variational principle (3.45) from Theorem 3.4:

- the set $\mathbb{L}(G)$ of locally consistent pseudomarginals (4.7) is a convex (polyhedral) outer bound on the marginal polytope $\mathbb{M}(G)$; and
- the Bethe entropy (4.14) is an approximation of the exact dual function $A^*(\tau)$.

By combining these two ingredients, we obtain the *Bethe variational problem* (BVP):

$$\max_{\tau \in \mathbb{L}(G)} \left\{ \langle \theta, \tau \rangle + \sum_{s \in V} H_s(\tau_s) - \sum_{(s,t) \in E} I_{st}(\tau_{st}) \right\}. \quad (4.16)$$

Note that this problem has a very simple structure: the cost function is given in closed form, it is differentiable, and the constraint set $\mathbb{L}(G)$ is a polytope specified by a small number of constraints. Given this special structure, one might suspect that there should exist a relatively simple algorithm for solving this optimization problem (4.16). Indeed, the sum-product algorithm turns out to be exactly such a method.

In order to develop this connection between the variational problem (4.16) and the sum-product algorithm, let λ_{ss} be a Lagrange

multiplier associated with the normalization constraint $C_{ss}(\tau) = 0$, where

$$C_{ss}(\tau) := 1 - \sum_{x_s} \tau_s(x_s). \qquad (4.17)$$

Moreover, for each direction $t \to s$ of each edge and each $x_s \in \mathcal{X}_s$, define the constraint function

$$C_{ts}(x_s; \tau) := \tau_s(x_s) - \sum_{x_t} \tau_{st}(x_s, x_t), \qquad (4.18)$$

and let $\lambda_{ts}(x_s)$ be a Lagrange multiplier associated with the constraint $C_{ts}(x_s; \tau) = 0$. These Lagrange multipliers turn out to be closely related to the sum-product messages, in particular via the relation $M_{ts}(x_s) \propto \exp(\lambda_{ts}(x_s))$.

We then consider the Lagrangian corresponding to the Bethe variational problem (4.16):

$$\mathcal{L}(\tau, \lambda; \theta) := \langle \theta, \tau \rangle + H_{\mathrm{Bethe}}(\tau) + \sum_{s \in V} \lambda_{ss} C_{ss}(\tau) \qquad (4.19)$$

$$+ \sum_{(s,t) \in E} \left[\sum_{x_s} \lambda_{ts}(x_s) C_{ts}(x_s; \tau) + \sum_{x_t} \lambda_{st}(x_t) C_{st}(x_t; \tau) \right].$$

With these definitions, the connection between sum-product and the BVP is made precise by the following result, due to Yedidia et al. [269].

Theorem 4.2(Sum-Product and the Bethe Problem). The sum-product updates are a Lagrangian method for attempting to solve the Bethe variational problem:

(a) For any graph G, any fixed point of the sum-product updates specifies a pair (τ^*, λ^*) such that

$$\nabla_\tau \mathcal{L}(\tau^*, \lambda^*; \theta) = 0, \quad \text{and} \quad \nabla_\lambda \mathcal{L}(\tau^*, \lambda^*; \theta) = 0. \qquad (4.20)$$

(b) For a tree-structured Markov random field (MRF), the Lagrangian Equations (4.20) have a unique solution (τ^*, λ^*), where the elements of τ^* correspond to the exact singleton and pairwise marginal distributions of the MRF. Moreover, the optimal value of the BVP is equal to the cumulant function $A(\theta)$.

Remark 4.1. It should be noted that the Lagrangian formulation (4.19) is a partial one, because it assigns Lagrange multipliers to the normalization C_{ss} and marginalization C_{ts} constraints, but deals with the nonnegativity constraints implicitly. However, any optimum τ^* of the Bethe variational principle with strictly positive elements must satisfy the Lagrangian conditions (4.20) from Theorem 4.2(a). To see this fact, note that if λ^* is Lagrange multiplier vector for the BVP, then any optimal solution τ^* must maximize the Lagrangian $\mathcal{L}(\tau, \lambda^*)$ over the positive orthant $\tau \geq 0$ (see Bertsekas [21]). If the optimal solution τ^* has strictly positive elements, then a necessary condition for Lagrangian optimality is the zero-gradient condition $\nabla_\tau \mathcal{L}(\tau^*, \lambda^*) = 0$, as claimed. Moreover, for graphical models where all configurations are given strictly positive mass (such as graphical models in exponential form with finite θ), the sum-product messages stay bounded strictly away from zero [204], so that there is always an optimum τ^* with strictly positive elements. In this case, Theorem 4.2(a) guarantees that any sum-product fixed point satisfies the Lagrangian conditions necessary to be an optimum of the Bethe variational principle. For graphical models in which some configurations are assigned zero mass, further care is required in connecting fixed points to local optima; we refer the reader to Yedidia et al. [269] for details.

Proof. Proof of part (a): Computing the partial derivative $\nabla_\tau \mathcal{L}(\tau; \lambda)$ and setting it to zero yields the relations

$$\log \tau_s(x_s) = \lambda_{ss} + \theta_s(x_s) + \sum_{t \in N(s)} \lambda_{ts}(x_s), \quad \text{and} \quad (4.21\text{a})$$

$$\log \frac{\tau_{st}(x_s, x_t)}{\widetilde{\tau}_s(x_s)\widetilde{\tau}_t(x_t)} = \theta_{st}(x_s, x_t) - \lambda_{ts}(x_s) - \lambda_{st}(x_t), \quad (4.21\text{b})$$

where we have used the shorthand $\widetilde{\tau}_s(x_s) := \sum_{x_t} \tau(x_s, x_t)$. The condition $\nabla_\lambda \mathcal{L}(\tau; \lambda) = 0$ is equivalent to $C_{ts}(x_s; \tau) = 0$ and $C_{ss}(\tau) = 0$. Using Equation (4.21a) and the fact that the marginalization condition $C_{ts}(x_s; \tau) = 0$ hold at the optimum (so that $\sum_{x_t} \tau(x_s, x_t) = \tau_s(x_s)$), we

can rearrange Equation (4.21b) to obtain:

$$
\begin{aligned}
\log \tau_{st}(x_s, x_t) = {} & \lambda_{ss} + \lambda_{tt} + \theta_{st}(x_s, x_t) + \theta_s(x_s) + \theta_t(x_t) \\
& + \sum_{u \in N(s) \setminus t} \lambda_{us}(x_s) + \sum_{u \in N(t) \setminus s} \lambda_{ut}(x_t).
\end{aligned}
\tag{4.22}
$$

So as to make explicit the connection to the sum-product algorithm, let us define, for each directed edge $t \to s$, an r_s-vector of "messages"

$$
M_{ts}(x_s) = \exp(\lambda_{ts}(x_s)), \quad \text{for all } x_s \in \{0, 1, \ldots, r_s - 1\}.
$$

With this notation, we can then write an equivalent form of Equation (4.21a) as follows:

$$
\tau_s(x_s) = \kappa \exp(\theta_s(x_s)) \prod_{t \in N(s)} M_{ts}(x_s).
\tag{4.23}
$$

Similarly, we have an equivalent form of Equation (4.22):

$$
\begin{aligned}
\tau_{st}(x_s, x_t) = {} & \kappa' \exp\big(\theta_{st}(x_s, x_t) + \theta_s(x_s) + \theta_t(x_t)\big) \\
& \times \prod_{u \in N(s) \setminus t} M_{us}(x_s) \prod_{u \in N(t) \setminus s} M_{ut}(x_t).
\end{aligned}
\tag{4.24}
$$

Here κ, κ' are positive constants dependent on λ_{ss} and λ_{tt}, chosen so that the pseudomarginals satisfy normalization conditions. Note that τ_s and τ_{st} so defined are nonnegative.

To conclude, we need to adjust the Lagrange multipliers or messages so that the constraint $\sum_{x_t} \tau_{st}(x_s, x_t) = \tau_s(x_s)$ is satisfied for every edge. Using the relations (4.23) and (4.24) and performing some algebra, the end result is

$$
M_{ts}(x_s) \propto \sum_{x_t} \left[\exp\big\{\theta_{st}(x_s, x_t) + \theta_t(x_t)\big\} \prod_{u \in N(t) \setminus s} M_{ut}(x_t) \right], \tag{4.25}
$$

which is equivalent to the familiar sum-product update (2.9). By construction, any fixed point M^* of these updates (4.25) specifies a pair (τ^*, λ^*) that satisfies the stationary conditions (4.20).

Proof of part (b): See Appendix B.4 for the proof of this claim.   □

The connection between the sum-product algorithm and the Bethe variational principle has a number of important consequences. First, it provides a principled basis for applying the sum-product algorithm for graphs with cycles, namely as a particular type of iterative method for attempting to satisfy Lagrangian conditions. It should be noted, however, that this connection between sum-product and the Bethe problem in itself provides no guarantees on the convergence of the sum-product updates on graphs with cycles. Indeed, whether or not the algorithm converges depends both on the potential strengths and the topology of the graph. In the standard scheduling of the messages, each node applies Equation (4.25) in parallel. Other more global schemes for message-passing are possible, and commonly used in certain applications like error-control coding [e.g., 166]; some recent work has also studied adaptive schedules for message-passing [75, 226]. Tatikonda and Jordan [230] established an elegant connection between convergence of parallel updates and Gibbs measures on the infinitely unwrapped computation tree, thereby showing that sufficient conditions for convergence can be obtained from classical conditions for uniqueness of Gibbs measures (e.g., Dobrushin's condition or Simon's condition [90]). In subsequent work, other researchers [109, 118, 178, 204] have used various types of contraction arguments to obtain sharper conditions for convergence, and/or uniqueness of fixed points. For suitably weak potentials, Dobrushin-type conditions and related contraction arguments guarantee both convergence of the updates, and as a consequence, uniqueness of the associated fixed point. A parallel line of work [111, 254, 270] has explored alternatives to sum-product that are guaranteed to converge, albeit at the price of increased computational cost. However, with the exception of trees and other special cases [110, 167, 188], the Bethe variational problem is usually a non-convex problem, in that H_{Bethe} fails to be concave. As a consequence, there are frequently local optima, so that even when using a convergent algorithm, there are no guarantees that it will find the global optimum.

For each $\theta \in \mathbb{R}^d$, let $A_{\text{Bethe}}(\theta)$ denote the optimal value of the Bethe variational problem (4.16). Theorem 4.2(b) states for any

tree-structured problem, we have the equality $A_{\text{Bethe}}(\theta) = A(\theta)$ for all $\theta \in \mathbb{R}^d$. Given this equivalence, it is natural to consider the relation between $A_{\text{Bethe}}(\theta)$ and the cumulant function $A(\theta)$ for general graphs. In general, the Bethe value $A_{\text{Bethe}}(\theta)$ is simply an approximation to the cumulant function value $A(\theta)$. Unlike the mean field methods to be discussed in Section 5, it is not guaranteed to provide a lower bound on the cumulant function. As will be discussed at more length in Section 7, Wainwright et al. [246] derived "convexified" forms of the Bethe variational principle that are guaranteed to yield upper bounds on the cumulant function for any graphical model. On the other hand, Sudderth et al. [224] show that $A_{\text{Bethe}}(\theta)$ is a lower bound on the cumulant function $A(\theta)$ for certain classes of attractive graphical models. Such models, in which the interactions encourage random variables to agree with one another, are common in computer vision and other applications in spatial statistics. This lower-bounding property is closely related to the connection between the Bethe approximation and loop series expansions [51], discussed in Section 4.1.6.

Another important consequence of the Bethe/sum-product connection is in suggesting a number of avenues for improving upon the ordinary sum-product algorithm, via progressively better approximations to the entropy function and outer bounds on the marginal polytope. We turn to discussion of a class of such generalized sum-product algorithms beginning in Section 4.2.

4.1.4   Inexactness of Bethe and Sum-Product

In this section, we explore some aspects of the inexactness of the sum-product algorithm. From a variational perspective, the inexactness stems from the two approximations made in setting up Bethe variational principle:

(a) Replacing the marginal polytope $\mathbb{M}(G)$ by the polyhedral outer bound $\mathbb{L}(G)$ and

(b) The Bethe entropy H_{Bethe} as an approximation to the exact entropy as a function of the mean parameters.

We begin by considering the Bethe entropy approximation, and its potential inexactness:

Example 4.2 (Inexactness of $H_{\mathbf{Bethe}}$). Consider the fully connected graph K_4 on four vertices, and the collection of singleton and pairwise marginal distributions given by

$$\mu_s(x_s) = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix} \quad \text{for } s = 1, 2, 3, 4 \qquad (4.26\text{a})$$

$$\mu_{st}(x_s, x_t) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \quad \forall \, (s, t) \in E. \qquad (4.26\text{b})$$

It can be verified that these marginals are globally valid, generated in particular by the distribution that places mass 0.5 on each of the configurations $(0, 0, 0, 0)$ and $(1, 1, 1, 1)$. Let us calculate the Bethe entropy approximation. Each of the four singleton entropies are given by $H_s(\mu_s) = \log 2$, and each of the six (one for each edge) mutual information terms are given by $I_{st}(\mu_{st}) = \log 2$, so that the Bethe entropy is given by

$$H_{\text{Bethe}}(\mu) = 4 \log 2 - 6 \log 2 = -2 \log 2 < 0,$$

which cannot be a true entropy. In fact, for this example, the true entropy (or value of the negative dual function) is given by $-A^*(\mu) = \log 2 > 0$.

In addition to the inexactness of H_{Bethe} as an approximation to the negative dual function, the Bethe variational principle also involves relaxing the marginal polytope $\mathbb{M}(G)$ to the first-order constraint set $\mathbb{L}(G)$. As illustrated in Example 4.1, the inclusion $\mathbb{M}(C_3) \subseteq \mathbb{L}(C_3)$ holds *strictly* for the 3-node cycle C_3. The constructive procedure of Example 4.1 can be substantially generalized to show that the inclusion $\mathbb{M}(G) \subset \mathbb{L}(G)$ holds strictly for any graph G with cycles. Figure 4.2 provides a highly idealized illustration[3] of the relation between $\mathbb{M}(G)$ and $\mathbb{L}(G)$: both sets are polytopes, and for a graph with cycles, $\mathbb{M}(G)$ is always strictly contained within the outer bound $\mathbb{L}(G)$.

---

[3] In particular, this picture is misleading in that it suggests that $\mathbb{L}(G)$ has more facets and more vertices than $\mathbb{M}(G)$; in fact, the polytope $\mathbb{L}(G)$ has fewer facets and more vertices, but this is difficult to convey in a 2D representation.
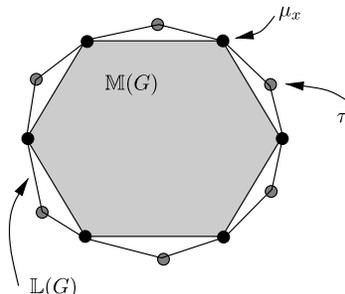
Fig. 4.2 Highly idealized illustration of the relation between the marginal polytope $\mathbb{M}(G)$ and the outer bound $\mathbb{L}(G)$. The set $\mathbb{L}(G)$ is always an outer bound on $\mathbb{M}(G)$, and the inclusion $\mathbb{M}(G) \subset \mathbb{L}(G)$ is strict whenever G has cycles. Both sets are polytopes and so can be represented either as the convex hull of a finite number of extreme points, or as the intersection of a finite number of half-spaces, known as facets.

Both sets are polytopes, and consequently can be represented either as the convex hull of a finite number of extreme points, or as the intersection of a finite number of half-spaces, known as facets. Letting ϕ be a shorthand for the full vector of indicator functions in the standard overcomplete representation (3.34), the marginal polytope has the convex hull representation $\mathbb{M}(G) = \operatorname{conv}\{\phi(x) \mid x \in \mathcal{X}\}$. Since the indicator functions are $\{0,1\}$-valued, all of its extreme points consist of $\{0,1\}$ elements, of the form $\mu_x := \phi(x)$ for some $x \in \mathcal{X}^m$; there are a total of $|\mathcal{X}^m|$ such extreme points. However, with the exception of tree-structured graphs, the number of facets for $\mathbb{M}(G)$ is not known in general, even for relatively simple cases like the Ising model; see the book [69] for background on the cut or correlation polytope, which is equivalent to the marginal polytope for an Ising model. However, the growth must be super-polynomial in the graph size, unless certain widely believed conjectures in computational complexity are false.

On the other hand, the polytope $\mathbb{L}(G)$ has a polynomial number of facets, upper bounded by any graph by $\mathcal{O}(rm + r^2|E|)$. It has more extreme points than $\mathbb{M}(G)$, since in addition to all the integral extreme points $\{\mu_x, x \in \mathcal{X}^m\}$, it includes other extreme points $\tau \in \mathbb{L}(G) \backslash \mathbb{M}(G)$ that contain fractional elements; see Section 8.4 for further discussion of integral versus fractional extreme points. With the exception of trees and small instances, the total number of extreme points of $\mathbb{L}(G)$ is not known in general.

The strict inclusion of $\mathbb{M}(G)$ within $\mathbb{L}(G)$ — and the fundamental role of the latter set in the Bethe variational problem (4.16) — leads to the following question: *do solutions to the Bethe variational problem ever fall into the gap* $\mathbb{M}(G)\backslash\mathbb{L}(G)$*?* The optimistically inclined would hope that these points would somehow be excluded as optima of the Bethe variational problem. Unfortunately, such hope turns out to be misguided. In fact, for every element τ of $\mathbb{L}(G)$, it is possible to construct a distribution p_θ such that if the sum-product algorithm is run on the problem, then τ arises from the messages defined by a sum-product fixed point. In order to understand this fact, we need to describe the reparameterization interpretation of the sum-product algorithm [242], to which we now turn.

4.1.5 Bethe Optima and Reparameterization

One view of the junction tree algorithm, as described in Section 2.5.2, is as follows: taking as input a set of potential functions on the cliques of some graph, it returns as output an *alternative factorization* of the same distribution in terms of local marginal distributions on the cliques and separator sets of a junction tree. In the special case of an ordinary tree, the alternative factorization is a product of local marginals at single nodes and edges of the tree, as in Equation (4.8). Indeed, the sum-product algorithm for trees can be understood as an efficient method for computing this alternative parameterization.

It turns out that the same interpretation applies to arbitrary graphs with cycles: more precisely, any fixed point of the sum-product algorithm — and even more generally, any local optimum of the Bethe variational principle — specifies a reparameterization of the original distribution p_θ. We summarize in the following proposition [242]:

Proposition 4.3 (Reparameterization Properties of Bethe Approximation). Let $\tau^* = (\tau^*_s, s \in V; \tau^*_{st}, (s,t) \in E)$ denote any optimum of the Bethe variational principle defined by the distribution p_θ. Then the distribution defined by the fixed point as

$$p_{\tau^*}(x) := \frac{1}{Z(\tau^*)} \prod_{s \in V} \tau^*_s(x_s) \prod_{(s,t) \in E} \frac{\tau^*_{st}(x_s, x_t)}{\tau^*_s(x_s)\tau^*_t(x_t)}, \qquad (4.27)$$

is a reparameterization of the original distribution p_θ — that is, $p_{\tau^*}(x) = p_\theta(x)$ for all $x \in \mathcal{X}^m$.

Note that this type of reparameterization is possible only because the exponential family is defined by an overcomplete set of sufficient statistics, involving the indicator functions (3.34). The reparameterization (4.27) is the analog of the tree-structured factorization (4.8), but as applied to a graph with cycles. In contrast to the tree case, the normalization constant $Z(\tau^*)$ is not, in general, equal to one. Whereas the junction tree theorem guarantees this reparameterization for any tree, it is not immediately obvious that even one reparameterization need exist for a general graph. In fact, the result establishes that every graph has at least one such reparameterization, and some graphs may have multiple reparameterizations of the form (4.27); in particular, this is the case for any problem for which the BVP has multiple optima. Moreover, the reparameterization viewpoint provides some insight into the *approximation error*: that is, the difference between the exact marginals μ_s of $p_\theta(x)$ and the approximations τ_s^* computed by the sum-product algorithm. Indeed, using Equation (4.27), it is possible to derive an exact expression for the error in the sum-product algorithm, as well as computable error bounds, as described in more detail in Wainwright et al. [242].

We now show how the reparameterization characterization (4.27) enables us to specify, for *any* pseudomarginal τ in the interior of $\mathbb{L}(G)$, a distribution p_θ for which τ is a fixed point of the sum-product algorithm. The following example illustrates this construction.

Example 4.3 (Fooling the Sum-Product Algorithm). Let us return to the simplest graph for which sum-product is not exact — namely, a single cycle with three nodes (see Example 4.1). Consider candidate marginal distributions $(\tau_s,\ s \in V)$ and $(\tau_{st},\ (s,t) \in E)$ of the form illustrated in Figure 4.1(a), with $\beta_{12} = \beta_{23} = 0.4$ and $\beta_{13} = 0.1$. As discussed in Example 4.1, this setting yields a set of pseudomarginals τ that lie in $\mathbb{L}(G)$ but *not* in $\mathbb{M}(G)$, and therefore could not possibly arise from any global probability distribution.

Let us now demonstrate how, for an appropriately chosen distribution p_θ on the graph, the sum-product algorithm can be "fooled" into converging to this pseudomarginal vector τ. Using the canonical overcomplete representation (3.34), consider a set of canonical parameters of the form:

$$\theta_s(x_s) := \log \tau_s(x_s) = \log \begin{bmatrix} 0.5 & 0.5 \end{bmatrix} \quad \forall\, s \in V, \text{ and} \quad (4.28\text{a})$$

$$\theta_{st}(x_s, x_t) := \log \frac{\tau_{st}(x_s, x_t)}{\tau_s(x_s)\tau_t(x_t)}$$

$$= \log 4 \begin{bmatrix} \beta_{st} & 0.5 - \beta_{st} \\ 0.5 - \beta_{st} & \beta_{st} \end{bmatrix} \quad \forall\, (s,t) \in E, \quad (4.28\text{b})$$

where we have adopted the short-hand notation from Equation (4.2). With these canonical parameters, suppose that we apply the sum-product algorithm to the Markov random field p_θ, using the uniform message initialization $M_{ts}(x_s) \propto \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}$. A little bit of algebra using the sum-product update (4.25) shows that for this parameter choice, the uniform messages M already define a fixed point of the sum-product algorithm. Moreover, if we compute the associated pseudomarginals specified by M and θ, they are equal to the previously specified τ_s, τ_{st}. In summary, the sum-product algorithm — when applied to the distribution p_θ defined by the canonical parameters (4.28) — produces as its output the pseudomarginal τ as its estimate of the true marginals.

The reader might object to the fact that the problem construction ensured the sum-product algorithm was already at this particular fixed point, and so obviates the possibility of the updates converging to some other fixed point if initialized in a different way. However, it is known [110, 249] that for any discrete Markov random field in exponential family form with at most a single cycle, sum-product has a unique fixed point, and always converges to it. Therefore, the sum-product fixed point that we have constructed (4.28) is the unique fixed point for this problem, and the algorithm converges to it from any initialization of the messages.

More generally, the constructive approach illustrated in the preceding example applies to an arbitrary member of the interior[4] of $\mathbb{L}(G)$. Therefore, for all pseudomarginals $\tau \in \mathbb{L}(G)$, including those that are not globally valid, there exists a distribution p_θ for which τ arises from a sum-product fixed point.

4.1.6 Bethe and Loop Series Expansions

In this section, we discuss the loop series expansions of Chertkov and Chernyak [51]. These expansions provide exact representation of the cumulant function as a sum of terms, with the first term corresponding to the Bethe approximation $A_{\mathrm{Bethe}}(\theta)$, and higher-order terms obtained by adding in so-called loop corrections. They provided two derivations of their loop series: one applies a trigonometric identity to a Fourier representation of binary variables, while the second is based upon a saddle point approximation obtained via an auxiliary field of complex variables. In this section, we describe a more direct derivation of the loop expansion based on the reparameterization characterization of sum-product fixed points given in Proposition 4.3. Although the loop series expansion can be developed for general factor graphs, considering the case of a pairwise Markov random field with binary variables — that is, the Ising model from Example 3.1 — suffices to illustrate the basic ideas. (See Sudderth et al. [224] for a derivation for more general factor graphs.)

Before stating the result, we require a few preliminary definitions. Given an undirected graph $G = (V, E)$ and some subset $\widetilde{E} \subseteq E$ of the edge set E, we let $G(\widetilde{E})$ denote the induced subgraph associated with $\widetilde{E}$ — that is, the graph with edge set $\widetilde{E}$, and vertex set

$$V(\widetilde{E}) := \{t \in V \mid (t, u) \in \widetilde{E} \text{ for some } u\}. \qquad (4.29)$$

For any vertex $s \in V$, we define its degree with respect to $\widetilde{E}$ as

$$d_s(\widetilde{E}) := \{t \in V \mid (s, t) \in \widetilde{E}\}. \qquad (4.30)$$

---

[4] Strictly speaking, it applies to members of the relative interior since, as described in the overcomplete representation (3.34), the set $\mathbb{L}(G)$ is not full-dimensional and hence has an empty interior.
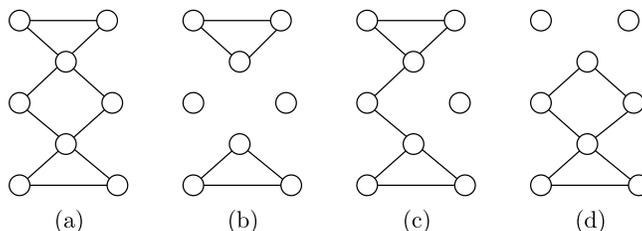
Fig. 4.3 Illustration of generalized loops. (a) Original graph. (b)–(d) Various generalized loops associated with the graph in (a). In this particular case, the original graph is a generalized loop for itself.

Following Chertkov and Chernyak [51], we define a *generalized loop* to be a subgraph $G(\widetilde{E})$ for which all nodes $s \in V$ have degree $d_s(\widetilde{E}) \neq 1$. Otherwise stated, for every node s, it either does not belong to $G(\widetilde{E})$ so that $d_s(\widetilde{E}) = 0$, or it has degree $d_s(\widetilde{E}) \geq 2$. See Figure 4.3 for an illustration of the concept of a generalized loop. Note also that a graph without cycles (i.e., a tree or forest graph) does not have any generalized loops.

Consider a BP fixed point for a pairwise MRF with binary variables — that is, an Ising model (3.8). For binary variables $X_s \in \{0,1\}$, the singleton and edgewise pseudomarginals associated with a BP fixed point can be parameterized as

$$\tau_s(x_s) = \begin{bmatrix} 1 - \tau_s \\ \tau_s \end{bmatrix}, \quad \text{and} \quad \tau_{st}(x_s, x_t) = \begin{bmatrix} 1 - \tau_s - \tau_t + \tau_{st} & \tau_t - \tau_{st} \\ \tau_s - \tau_{st} & \tau_{st} \end{bmatrix} \tag{4.31}$$

Some calculation shows that membership in the set $\mathbb{L}(G)$ is equivalent to imposing the following four inequalities

$$\tau_{st} \geq 0, \quad 1 - \tau_s - \tau_t + \tau_{st} \geq 0, \quad \tau_s - \tau_{st} \geq 0, \quad \text{and} \quad \tau_t - \tau_{st} \geq 0,$$

for each edge $(s,t) \in E$. See also Example 3.8 for some discussion of the mean parameters for an Ising model.

Using this parameterization, for each edge $(s,t) \in E$, we define the *edge weight*

$$\beta_{st} := \frac{\tau_{st} - \tau_s \tau_t}{\tau_s(1 - \tau_s)\tau_t(1 - \tau_t)}, \tag{4.32}$$

which extends naturally to the *subgraph weight* $\beta_{\widetilde{E}} := \prod_{(s,t)\in\widetilde{E}} \beta_{st}$. With these definitions, we have [224]:

Proposition 4.4. Consider a pairwise Markov random field with binary variables (3.8), and let $A_{\text{Bethe}}(\theta)$ be the optimized free energy (4.16) evaluated at a BP fixed point $\tau = (\tau_s, s \in V; \ \tau_{st}, \ (s,t) \in E)$. The cumulant function value $A(\theta)$ is equal to the loop series expansion:

$$
A(\theta) = A_{\text{Bethe}}(\theta) + \log\left\{1 + \sum_{\emptyset \neq \widetilde{E} \subseteq E} \beta_{\widetilde{E}} \prod_{s \in V} \mathbb{E}_{\tau_s}[(X_s - \tau_s)^{d_s(\widetilde{E})}]\right\}.
$$

(4.33)

Before proving Proposition 4.4, we pause to make some remarks. By definition, we have

$$
\begin{aligned}
\mathbb{E}_{\tau_s}[(X_s - \tau_s)^d] &= (1 - \tau_s)(-\tau_s)^d + \tau_s(1 - \tau_s)^d \\
&= \tau_s(1 - \tau_s)\big[(1 - \tau_s)^{d-1} + (-1)^d(\tau_s)^{d-1}\big],
\end{aligned}
$$
(4.34)

corresponding to dth central moments of a Bernoulli variable with parameter $\tau_s \in [0,1]$. Consequently, for any $\widetilde{E} \subseteq E$ such that $d_s(\widetilde{E}) = 1$ for at least one $s \in V$, then the associated term in the expansion (4.33) vanishes. For this reason, only generalized loops $\widetilde{E}$ lead to nonzero terms in the expansion (4.33). The terms associated with these generalized loops effectively define corrections to the Bethe estimate $A_{\text{Bethe}}(\theta)$ of the cumulant function. Tree-structured graphs do not contain any nontrivial generalized loops, which provides an alternative proof of the exactness of the Bethe approximation for trees.

The first term $A_{\text{Bethe}}(\theta)$ in the loop expansion is easily computed from any BP fixed point, since it simply corresponds to the optimized value of the Bethe free energy (4.16). However, explicit computation of the full sequence of loop corrections — and hence exact calculation of the cumulant function — is intractable for general (nontree) models. For instance, any fully connected graph with $n \geq 5$ nodes has more than 2^n generalized loops. In some cases, accounting for a small set of significant loop corrections may lead to improved approximations to

the partition function [100], or more accurate approximations of the marginals for LDPC codes [52].

Proof of Proposition 4.4: Recall that the canonical overcomplete parameterization (3.34) using indicator functions is overcomplete, so that there exist many distinct parameter vectors $\theta \neq \theta'$ such that $p_\theta = p_{\theta'}$. However, the difference between the left-hand and right-hand sides of the loop series expansion (4.33) is identical for any choice of θ, since moving from $\theta \to \theta'$ simply shifts both $A(\theta)$ and $A_{\text{Bethe}}(\theta)$ by some constant. Consequently, it suffices to prove the loop series expansion for a single parameterization; in particular, we prove it for the reparameterization given by the canonical parameters

$$\widetilde{\theta}_s(x_s) = \log \tau_s(x_s), \quad \text{and} \quad \widetilde{\theta}_{st}(x_s, x_t) = \log \frac{\tau_{st}(x_s, x_t)}{\tau_s(x_s)\tau_t(x_t)}, \qquad (4.35)$$

as specified by any BP fixed point (see Proposition 4.3). For this reparameterization, a little calculation shows that $A_{\text{Bethe}}(\widetilde{\theta}) = 0$. Consequently, it suffices to show that for this particular parameterization (4.35), we have the equality

$$A(\widetilde{\theta}) = \log \left\{ 1 + \sum_{\emptyset \neq \widetilde{E} \subseteq E} \beta_{\widetilde{E}} \prod_{s \in V} \mathbb{E}_{\tau_s}[(X_s - \tau_s)^{d_s(\widetilde{E})}] \right\}. \qquad (4.36)$$

Using the representation (4.31), a little calculation shows that

$$\frac{\tau_{st}(x_s, x_t)}{\tau_s(x_s)\tau_t(x_t)} = 1 + \beta_{st}(x_s - \tau_s)(x_t - \tau_t). \qquad (4.37)$$

By definition of $\widetilde{\theta}$, we have

$$\exp(A(\widetilde{\theta})) = \sum_{x \in \{0,1\}^m} \prod_{s \in V} \tau_s(x_s) \prod_{(s,t) \in E} \frac{\tau_{st}(x_s, x_t)}{\tau_s(x_s)\tau_t(x_t)}.$$

Let $\mathbb{E}$ denote expectation taken with respect to the product distribution $\tau_{\text{fact}}(x) := \prod_s \tau_s(x_s)$. With this notation, and applying the

identity (4.37), we have

$$\exp(A(\widetilde{\theta})) = \sum_{x \in \{0,1\}^n} \prod_{s \in V} \tau_s(x_s) \prod_{(s,t) \in E} \frac{\tau_{st}(x_s, x_t)}{\tau_s(x_s)\tau_t(x_t)}$$

$$= \mathbb{E}\left[\prod_{(s,t) \in E} \big(1 + \beta_{st}(X_s - \tau_s)(X_t - \tau_t)\big) \right]. \quad (4.38)$$

Expanding this polynomial and using linearity of expectation, we recover one term for each nonempty subset $\widetilde{E} \subseteq E$ of the graph's edges:

$$\exp(A(\widetilde{\theta})) = 1 + \sum_{\emptyset \neq \widetilde{E} \subseteq E} \mathbb{E}\left[\prod_{(s,t) \in \widetilde{E}} \beta_{st}(X_s - \tau_s)(X_t - \tau_t) \right]. \quad (4.39)$$

The expression (4.36) then follows from the independence structure of $\tau_{\mathrm{fact}}(x)$, and standard formulas for the moments of Bernoulli random variables. To evaluate these terms, note that if $d_s(\widetilde{E}) = 1$, it follows that $\mathbb{E}[X_s - \tau_s] = 0$. There is thus one loop correction for each generalized loop $\widetilde{E}$, in which all connected nodes have degree at least two.   □

Proposition 4.4 has extensions to more general factor graphs; see the papers [51, 224] for more details. Moreover, Sudderth et al. [224] show that the loop series expansion can be exploited to show that for certain types of graphical models with attractive interactions, the Bethe value $A_{\mathrm{Bethe}}(\theta)$ is actually a lower bound on the true cumulant function value $A(\theta)$.

4.2   Kikuchi and Hypertree-based Methods

From our development thus far, we have seen that there are two *distinct* ways in which the Bethe variational principle (4.16) is an approximate version of the exact variational principle (3.45). First, for general graphs, the Bethe entropy (4.14) is only an approximation to the true entropy or negative dual function. Second, the constraint set $\mathbb{L}(G)$ outer bound on the marginal polytope $\mathbb{M}(G)$, as illustrated in Figure 4.2. In principle, the accuracy of the Bethe variational principle could be strengthened by improving either one, or both, of these components. This section is devoted to one natural

generalization of the Bethe approximation, first proposed by Yedidia et al. [268, 269] and further explored by various researchers [111, 167, 188, 240, 270], that improves both components simultaneously. The origins of these methods lie in the statistical physics literature, where they were referred to as cluster variational methods [6, 133, 229].

At the high level, the approximations in the Bethe approach are based on trees, which represent a special case of the junction trees. A natural strategy, then, is to strengthen the approximations by exploiting more complex junction trees. These approximations are most easily understood in terms of hypertrees, which represent an alternative way in which to describe junction trees. Accordingly, we begin with some necessary background on hypergraphs and hypertrees.

4.2.1 Hypergraphs and Hypertrees

A hypergraph $G = (V, E)$ is a generalization of a graph, consisting of a vertex set $V = \{1, \ldots, m\}$, and a set of hyperedges E, where each *hyperedge h* is a particular subset of V. The hyperedges form a partially ordered set or poset [221], where the partial ordering is specified by inclusion. We say that a hyperedge h is *maximal* if it is not contained within any other hyperedge. With these definitions, we see that an ordinary graph is a special case of a hypergraph, in which each maximal hyperedge consists of a pair of vertices or equivalently, an ordinary edge of the graph.[5]

A convenient graphical representation of a hypergraph is in terms of a diagram of its hyperedges, with directed edges representing the inclusion relations; such a representation is known as a *poset diagram* [167, 188, 221]. Figure 4.4 provides some simple graphical illustrations of hypergraphs. Any ordinary graph, as a special case of a hypergraph, can be drawn in terms of a poset diagram; in particular, panel (a) shows the hypergraph representation of a single cycle on

---

[5] There is a minor inconsistency in our definition of the hyperedge set E and previous graph-theoretic terminology; for hypergraphs (unlike graphs), the set of hyperedges can include a individual vertex $\{s\}$ as an element.

four nodes. Panel (b) shows a hypergraph that is not equivalent to an ordinary graph, consisting of two hyperedges of size three joined by
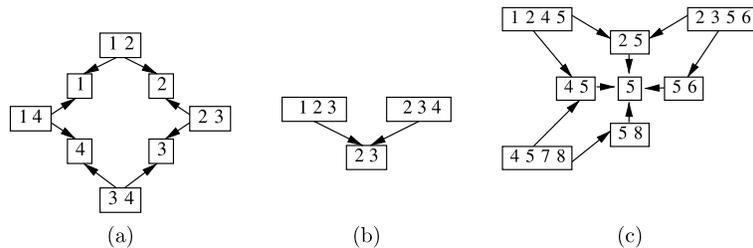

Fig. 4.4 Graphical representations of hypergraphs. Subsets of nodes corresponding to hyperedges are shown in rectangles, whereas the arrows represent inclusion relations among hyperedges. (a) An ordinary single cycle graph represented as a hypergraph. (b) A simple hypertree of width two. (c) A more complex hypertree of width three.

their intersection of size two. Shown in panel (c) is a more complex hypertree, to which we will return in the sequel.

Hypertrees or acyclic hypergraphs provide an alternative way to describe the concept of junction trees, as originally described in Section 2.5.2. In particular, a hypergraph is *acyclic* if it is possible to specify a junction tree using its maximal hyperedges and their intersections. The *width* of an acyclic hypergraph is the size of the largest hyperedge minus one; we use the term *k-hypertree* to mean an acyclic hypergraph of width k consisting of a single connected component. Thus, for example, a spanning tree of an ordinary graph is a 1-hypertree, because its maximal hyperedges, corresponding to ordinary edges in the original graph, all have size two. As a second example, consider the hypergraph shown in Figure 4.4(c). It is clear that this hypergraph is equivalent to the junction tree with maximal cliques $\{(1245), (4578), (2356)\}$ and separator sets $\{(25), (45)\}$. Since the maximal hyperedges have size four, this hypergraph is a hypertree of width three.

4.2.2 Hypertree-based factorization and entropy decomposition

With this background, we now specify an alternative form of the junction tree factorization (2.12), and show how it leads to a local decomposition of the entropy. Recall that we view the set E of hyperedges

as a partially ordered set. As described in Appendix E.1, associated with any poset is a Möbius function $\omega : E \times E \to \mathbb{R}$. Using the set of marginals $\mu = (\mu_h,\ h \in E)$ associated with the hyperedge set, we can define a new set of functions $\varphi := (\varphi_h,\ h \in E)$ as follows:

$$\log \varphi_h(x_h) := \sum_{g \subseteq h} \omega(g,h) \log \mu_g(x_g). \tag{4.40}$$

As a consequence of this definition and the Möbius inversion formula (see Lemma E.1 in Appendix E.1), the marginals can be represented as

$$\log \mu_h(x_h) = \sum_{g \subseteq h} \log \varphi_g(x_g). \tag{4.41}$$

This formula provides a useful recursive method for computing the functions φ_g, as illustrated in the examples to follow.

The significance of the functions (4.40) is that they yield a particularly simple factorization for any hypertree-structured graph. In particular, for a hypertree with an edge set containing all intersections between maximal hyperedges, the underlying distribution is guaranteed to factorize as follows:

$$p_\mu(x) = \prod_{h \in E} \varphi_h(x_h; \mu). \tag{4.42}$$

Here we use the notation $\varphi_h(x_h; \mu)$ to emphasize that φ_h is a function of the marginals μ. Equation (4.42) is an alternative formulation of the well-known junction tree decomposition (2.12). Let us consider some examples to gain some intuition.

Example 4.4 (Hypertree Factorization).

(a) First, suppose that the hypertree is an ordinary tree, in which case the hyperedge set consists of the union of the vertex set with the (ordinary) edge set. Viewing this hyperedge set as a poset ordered by inclusion, it can be shown (see Appendix E.1) that the Möbius function takes the form $\omega(g,g) = 1$ for all $g \in E$; $\omega(\{s\},\{s,t\}) = -1$ for all vertices s and ordinary edges $\{s,t\}$, and $\omega(g,h) = 0$ for all g that are not contained within h. Consequently, for any ordinary edge $\{s,t\}$, we have

$$\varphi_{st}(x_s, x_t) = \frac{\mu_{st}(x_s, x_t)}{\mu_s(x_s)\mu_t(x_t)},$$

whereas for each vertex, we have $\varphi_s(x_s) = \mu_s(x_s)$. Consequently, in this special case, Equation (4.42) reduces to the tree factorization in Equation (4.8).

(b) Turning to a more complex example, consider the acyclic hypergraph specified by the hyperedge set

$$E = \big\{(1245),(2356),(4578),(25),(45),(56),(58),(5)\big\},$$

as illustrated in Figure 4.4(c). Rather than computing the Möbius function for this poset, it is more convenient to do the computation via Equation (4.41). In particular, omitting explicit dependence on x for notational simplicity, we first calculate the singleton function $\varphi_5 = \mu_5$, and the pairwise function $\varphi_{25} = \mu_{25}/\mu_5$, with analogous expressions for the other pairwise terms. Finally, applying the representation (4.41) to $h = (1245)$, we have

$$\varphi_{1245} = \frac{\mu_{1245}}{\varphi_{25}\varphi_{45}\varphi_5} = \frac{\mu_{1245}}{\frac{\mu_{25}}{\mu_5}\frac{\mu_{45}}{\mu_5}\mu_5} = \frac{\mu_{1245}\,\mu_5}{\mu_{25}\mu_{45}}.$$

Similar reasoning yields analogous expressions for φ_{2356} and φ_{4578}. Putting the pieces together yields that the density p_μ factorizes as

$$
\begin{aligned}
p_\mu &= \frac{\mu_{1245}\mu_5}{\mu_{25}\mu_{45}}\,\frac{\mu_{2356}\mu_5}{\mu_{25}\mu_{56}}\,\frac{\mu_{4578}\mu_5}{\mu_{45}\mu_{58}}\,\frac{\mu_{25}}{\mu_5}\,\frac{\mu_{45}}{\mu_5}\,\frac{\mu_{56}}{\mu_5}\,\frac{\mu_{58}}{\mu_5}\,\mu_5 \\
&= \frac{\mu_{1245}\,\mu_{2356}\,\mu_{4578}}{\mu_{25}\,\mu_{45}},
\end{aligned}
$$

which agrees with the expression from the junction tree formula (2.12).

An immediate but important consequence of the factorization (4.42) is a local decomposition of the entropy. This decomposition can be expressed either as a sum of multi-information terms over the hyperedges, or as a weighted sum of entropy terms. More precisely, for each hyperedge $g \in E$, let us define the *hyperedge entropy*

$$H_h(\mu_h) := -\sum_{x_h} \mu_h(x_h) \log \mu_h(x_h), \tag{4.43}$$

as defined by the marginal μ_h, and the *multi-information*

$$I_h(\mu_h) := \sum_{x_h} \mu_h(x_h) \log \varphi_h(x_h), \tag{4.44}$$

defined by the marginal μ_h, and function φ_h.

In terms of these quantities, the entropy of any hypertree-structured distribution has the additive decomposition

$$H_{\text{hyper}}(\mu) = -\sum_{h \in E} I_h(\mu_h), \tag{4.45}$$

which follows immediately from the hypertree factorization (4.42) and the definition of I_h.

It is also possible to derive an alternative decomposition in terms of the hyperedge entropies. Using the Möbius inversion relation (4.40), we have

$$
\begin{aligned}
I_h(\mu_h) &= \sum_{g \subseteq h} \omega(g, h) \Big\{ \sum_{x_h} \mu_h(x_h) \log \mu_g(x_g) \Big\} \\
&= \sum_{f \subseteq h} \sum_{e \supseteq f} \omega(e, f) \Big\{ \sum_{x_f} \mu_h(x_f) \log \mu_f(x_f) \Big\} \\
&= -\sum_{f \subseteq h} c(f) H_f(\mu_f),
\end{aligned}
$$

where we have defined the *overcounting numbers*

$$c(f) := \sum_{e \supseteq f} \omega(f, e). \tag{4.46}$$

Thus, we have the alternative decomposition

$$H_{\text{hyper}}(\mu) = \sum_{h \in E} c(h) H_h(\mu_h). \tag{4.47}$$

We illustrate the decompositions (4.45) and (4.47) by continuing with Example 4.4:

Example 4.5 (Hypertree Entropies).

(a) For an ordinary tree, there are two types of multi-information: for an edge (s, t), I_{st} is equivalent to the ordinary mutual information, whereas for any vertex $s \in V$, the term I_s is equal to the negative entropy $-H_s$. Consequently, in this special case, Equation (4.45) is equivalent to the tree

entropy given in Equation (4.11). The overcounting numbers for a tree are $c(\{(s,t)\}) = 1$ for any edge (s,t), and $c(\{s\}) = 1 - d(s)$ for any vertex s, where $d(s)$ denotes the number of neighbors of s. Consequently, the decomposition (4.47) reduces to the alternative form (4.15) of the Bethe entropy.

(b) Consider again the hypertree in Figure 4.4(c). On the basis of our previous calculations in Example 4.4(c), we calculate $I_{1245} = -\big[H_{1245} - H_{25} - H_{45} + H_5\big]$. The expressions for the other two maximal hyperedges (i.e., I_{2356} and I_{4578}) are analogous. Similarly, we can compute $I_{25} = H_5 - H_{25}$, with analogous expressions for the other hyperedges of size two. Finally, we have $I_5 = -H_5$. Putting the pieces together and doing some algebra yields $H_{\text{hyper}} = H_{1245} + H_{2356} + H_{4578} - H_{25} - H_{45}$.

4.2.3   Kikuchi and Related Approximations

Recall that the core of the Bethe approach of Section 4 consists of a particular tree-based (Bethe) approximation to entropy, and a tree-based outer bound on the marginal polytope. The Kikuchi method and related approximations extend these tree-based approximations to ones based on more general hypertrees, as we now describe. Consider a Markov random field (MRF) defined by some (non-acyclic) hypergraph $G = (V, E)$, giving rise to an exponential family distribution p_θ of the form:

$$p_\theta(x) \propto \exp\left\{\sum_{h \in E} \theta_h(x_h)\right\}. \tag{4.48}$$

Note that this equation reduces to our earlier representation (4.1) of a pairwise MRF when the hypergraph is an ordinary graph.

Let $\tau = \{\tau_h\}$ be a collection of local marginals associated with the hyperedges $h \in E$. These marginals must satisfy the obvious *normalization condition*

$$\sum_{x'_h} \tau_h(x'_h) = 1. \tag{4.49}$$

Similarly, these local marginals must be consistent with one another wherever they overlap; more precisely, for any pair of hyperedges $g \subset h$, the *marginalization condition*

$$\sum_{\{x'_h \mid x'_g = x_g\}} \tau_h(x'_h) = \tau_g(x_g) \tag{4.50}$$

must hold. Imposing these normalization and marginalization conditions leads to the following constraint set:

$$\mathbb{L}_t(G) = \left\{ \tau \geq 0 \mid \text{Conditions (4.49)} \ \forall h, \text{ and (4.50)} \ \forall g \subset h \right\}. \tag{4.51}$$

Note that this constraint set is a natural generalization of the tree-based constraint set defined in Equation (4.7). In particular, definition (4.51) coincides with definition (4.7) when the hypergraph G is an ordinary graph. As before, we refer to members $\mathbb{L}_t(G)$ as *pseudomarginals*. By the junction tree conditions in Proposition 2.1, the local constraints defining $\mathbb{L}_t(G)$ are sufficient to guarantee global consistency whenever G is a hypertree.

In analogy to the Bethe entropy approximation, the entropy decomposition (4.45) motivates the following hypertree-based approximation to the entropy:

$$H_{\text{app}}(\tau) = \sum_{g \in E} c(g) H_g(\tau_g), \tag{4.52}$$

where H_g is the hyperedge-based entropy (4.44), and $c(g) := \sum_{f \supseteq g} \omega(g, f)$ is the overcounting number defined in Equation (4.46). This entropy approximation and the outer bound $\mathbb{L}_t(G)$ on the marginal polytope, in conjunction, lead to the following hypertree-based approximation to the exact variational principle:

$$\max_{\tau \in \mathbb{L}_t(G)} \left\{ \langle \theta, \tau \rangle + H_{\text{app}}(\tau) \right\}. \tag{4.53}$$

This problem is the hypertree-based generalization of the Bethe variational problem (4.16).

Example 4.6 (Kikuchi Approximation). To illustrate the approximate variational principle (4.53), consider the hypergraph shown in
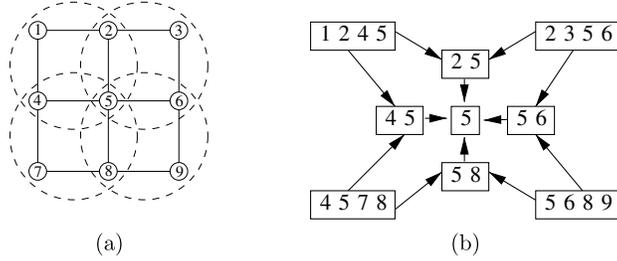
Fig. 4.5 (a) Kikuchi clusters superimposed upon a 3×3 lattice graph. (b) Hypergraph defined by this Kikuchi clustering.

Figure 4.5(b). This hypergraph arises from applying the Kikuchi clustering method [269] to the 3×3 lattice in panel (a); see Appendix D for more details. We determine the form of the entropy approximation H_{app} for this hypergraph by first calculating the overcounting numbers. By definition, $c(h) = 1$ for each of the four maximal hyperedges (e.g., $h = (1245)$). Since each of the 2-hyperedges has two parents, a little calculation shows that $c(g) = -1$ for each 2-hyperedge g. A final calculation shows that $c(\{5\}) = 1$, so that the overall entropy approximation takes the form:

$$H_{\mathrm{app}} = [H_{1245} + H_{2356} + H_{4578} + H_{5689}]$$
$$- [H_{25} + H_{45} + H_{56} + H_{58}] + H_5. \qquad (4.54)$$

Since the hypergraph in panel (b) has treewidth 3, the appropriate constraint set is the polytope $\mathbb{L}_3(G)$, defined over the pseudomarginals $(\tau_h, \ h \in E)$. In particular, it imposes nonnegativity constraints, normalization constraints, and marginalization constraints of the form:

$$\sum_{x_1', x_2'} \tau_{1245}(x_1', x_2', x_4, x_5) = \tau_{45}(x_4, x_5), \quad \text{and} \quad \sum_{x_6'} \tau_{56}(x_5, x_6') = \tau_5(x_6).$$

4.2.4   Generalized Belief Propagation

In principle, the variational problem (4.53) could be solved by a number of methods. Here we describe a Lagrangian-based message-passing algorithm that is a natural generalization of the ordinary sum-product updates for the Bethe approximation. As indicated by its name, the

defining feature of this scheme is that the only messages passed are from parents to children — i.e., along directed edges in the poset representation of a hypergraph.

In the hypertree-based variational problem (4.53), the variables correspond to a pseudomarginal τ_h for each hyperedge $E \in E'$. As with the earlier derivation of the sum-product algorithm, a Lagrangian formulation of this optimization problem leads to a specification of the optimizing pseudomarginals in terms of messages, which represent Lagrange multipliers associated with the constraints. There are various Lagrangian reformulations of the original problem [e.g., 167, 268, 269], which lead to different message-passing algorithms. Here we describe the *parent-to-child* form of message-passing derived by Yedidia et al. [269].

In order to describe the message-passing updates, it is convenient to define for a given hyperedge h, the sets of its *descendants* and *ancestors* in the following way:

$$\mathcal{D}(h) := \{g \in E \mid g \subset h\}, \quad \mathcal{A}(h) := \{g \in E \mid g \supset h\}. \quad (4.55)$$

For example, given the hyperedge $h = (1245)$ in the hypergraph in Figure 4.4(c), we have $\mathcal{A}(h) = \emptyset$ and $\mathcal{D}(h) = \{(25), (45), (5)\}$. We use the notation $\mathcal{D}^+(h)$ and $\mathcal{A}^+(h)$ as shorthand for the sets $\mathcal{D}(h) \cup h$ and $\mathcal{A}(h) \cup h$, respectively.

Given a pair of hyperedges (f, g), we let $M_{f \to g}(x_g)$ denote the "message" passed from hyperedge f to hyperedge g. More precisely, this message is a function over the state space of x_g (e.g., a multi-dimensional array in the case of discrete random variables). In terms of these messages, the pseudomarginal τ_h in parent-to-child message-passing takes the following form:

$$\tau_h(x_h) \propto \left[\prod_{g \in \mathcal{D}^+(h)} \psi_g(x_g; \theta) \right] \left[\prod_{g \in \mathcal{D}^+(h)} \prod_{f \in \mathrm{Par}(g) \backslash \mathcal{D}^+(h)} M_{f \to g}(x_g) \right],$$
$$(4.56)$$

where we have introduced the convenient shorthand $\psi_g(x_g; \theta) = \exp(\theta(x_g))$. In this equation, the pseudomarginal τ_h includes a compatibility function ψ_g for each hyperedge g in the set

$\mathcal{D}^+(h) := \mathcal{D}(h) \cup h$. It also collects a message from each hyperedge $f \notin \mathcal{D}^+(h)$ that is a parent of some hyperedge $g \in \mathcal{D}^+(h)$. We illustrate this construction by following up on Example 4.6.

Example 4.7 (Parent-to-child for Kikuchi). In order to illustrate the parent-to-child message-passing, consider the Kikuchi approximation for a 3×3 grid, illustrated in Figure 4.5. Focusing first on the hyperedge (1245), the first term in Equation (4.56) specifies a product of compatibility functions ψ_g as g ranges over $\mathcal{D}^+(1245)$, which in this case yields the product $\psi_{1245}\psi_{25}\psi_{45}\psi_5$. We then take the product over messages from hyperedges that are parents of hyperedges in $\mathcal{D}^+\{(1245)\}$, excluding hyperedges in $\mathcal{D}^+\{(1245)\}$ itself. Figure 4.6(a) provides an illustration; the set $\mathcal{D}^+\{(1245)\}$ is given by the hyperedges within the dotted ellipses. In this case, the set $\cup_g \mathrm{Par}(g) \backslash \mathcal{D}^+(h)$ is given by (2356) and (4578), corresponding to the parents of (25) and (45), respectively, combined with hyperedges (56) and (58), which are both parents of hyperedge (5). The overall result is an expression of the following form:

$$\tau_{1245} \propto \psi'_{12}\; \psi'_{14}\; \psi'_{25}\; \psi'_{45}\;\; \psi'_1\; \psi'_2\; \psi'_4\; \psi'_5$$
$$\times M_{(2356)\to(25)}M_{(4578)\to(45)}M_{(56)\to5}M_{(58)\to5}.$$

By symmetry, the expressions for the pseudomarginals on the other 4-hyperedges are analogous. By similar arguments, it is straightforward

(a)   (b)
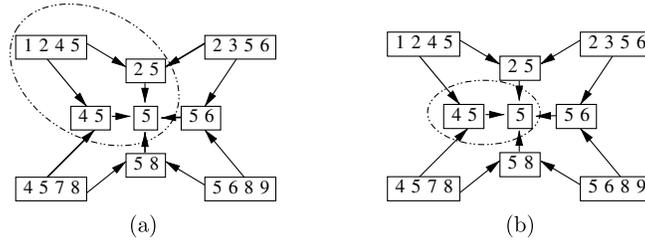
Fig. 4.6 Illustration of relevant regions for parent-to-child message-passing in a Kikuchi approximation. (a) Message-passing for hyperedge (1245). Set of descendants $\mathcal{D}^+\{(1245)\}$ is shown within a dotted ellipse. Relevant parents for τ_{1245} consists of the set $\{(2356), (4578), (56), (58)\}$. (b) Message-passing for hyperedge (45). Dotted ellipse shows descendant set $\mathcal{D}^+\{(45)\}$. In this case, relevant parent hyperedges are $\{(1245), (4578), (25), (56), (58)\}$.

to compute the following expression for τ_{45} and τ_5:

$$\tau_{45} \propto \psi'_{45} \, \psi'_4 \, \psi'_5 \; M_{(1245)\to(45)} M_{(4578)\to(45)} \; M_{(25)\to 5} \; M_{(56)\to 5} \; M_{(58)\to 5}$$
$$\tau_5 \propto \psi'_5 \, M_{(45)\to 5} \, M_{(25)\to 5} \, M_{(56)\to 5} \, M_{(58)\to 5}.$$

Generalized forms of the sum-product updates follow by updating the messages so as to enforce the marginalization constraints defining membership in $\mathbb{L}(G)$; as in the proof of Theorem 4.2, fixed points of these updates satisfy the necessary stationary conditions of the Lagrangian formulation. Further details on different variants of generalized sum-product updates can be found in various papers [268, 269, 188, 167, 128].

4.3 Expectation-Propagation Algorithms

There are a variety of other algorithms in the literature that are message-passing algorithms in the spirit of the sum-product algorithm. Examples of such algorithms include the family of expectation-propagation algorithms due to Minka [175], the related class of assumed density filtering methods [152, 164, 40], expectation-consistent inference [185], structured summary-propagation algorithms [64, 115], and the adaptive TAP method of Opper and Winther [183, 184]. These algorithms are often defined operationally in terms of sequences of local moment-matching updates, with variational principles invoked only to characterize each individual update, not to characterize the overall approximation. In this section, we show that these algorithms are in fact variational inference algorithms, involving a particular kind of approximation to the exact variational principle in Theorem 3.4.

The earliest forms of assumed density filtering [164] were developed for time series applications, in which the underlying graphical model is a hidden Markov model (HMM), as illustrated in Figure 2.4(a). As we have discussed, for discrete random variables, the marginal distributions for an HMM can be computed using the forward–backward algorithm, corresponding to a particular instantiation of the sum-product algorithm. Similarly, for a Gauss–Markov process, the Kalman filter also computes the means and covariances at each node of an HMM. However, given a hidden Markov model involving general

continuous random variables, the message M_{ts} passed from node t to s is a real-valued function, and hence difficult to store and transmit.[6] The purpose of assumed density filtering is to circumvent the computational challenges associated with passing function-valued messages. Instead, assumed density filtering (ADF) operates by passing approximate forms of the messages, in which the true message is approximated by the closest member of some tractable class. For instance, a general continuous message might be approximated with a Gaussian message. This procedure of computing the message approximations, if closeness is measured using the Kullback–Leibler divergence, can be expressed in terms of moment-matching operations.

Minka [175] observed that the basic ideas underlying ADF can be generalized beyond Markov chains to arbitrary graphical models, an insight that forms the basis for the family of expectation-propagation (EP) algorithms. As with assumed density filtering, expectation-propagation [174, 175] and various related algorithms [64, 115, 185] are typically described in terms of moment-matching operations. To date, the close link between these algorithms and the Bethe approximation does not appear to have been widely appreciated. In this section, we show that these algorithms are methods for solving certain relaxations of the exact variational principle from Theorem 3.4, using "Bethe-like" entropy approximations and particular convex outer bounds on the set $\mathcal{M}$. More specifically, we recover the moment-matching updates of expectation-propagation as one particular type of Lagrangian method for solving the resulting optimization problem, thereby showing that expectation-propagation algorithms belong to the same class of variational methods as belief propagation. This section is a refinement of results from the thesis [240].

4.3.1   Entropy Approximations Based on Term Decoupling

We begin by developing a general class of entropy approximations based on decoupling an intractable collection of terms. Given a collection of

---

[6] The Gaussian case is special, in that this functional message can always be parameterized in terms of its "mean" and "variance," and the efficiency of Kalman filtering stems from this fact.

random variables $(X_1, \ldots, X_m) \in \mathbb{R}^m$, consider a collection of sufficient statistics that is partitioned as

$$\underbrace{\phi := \big(\phi_1, \phi_2, \ldots, \phi_{d_T}\big)}_{\text{Tractable component}}, \quad \text{and} \quad \underbrace{\Phi := \big(\Phi^1, \Phi^2, \ldots, \Phi^{d_I}\big)}_{\text{Intractable component.}}, \quad (4.57)$$

where ϕ_i are univariate statistics and the Φ^i are generally multivariate. As will be made explicit in the examples to follow, the partitioning separates the tractable from the intractable components of the associated exponential family distribution.

Let us set up various exponential families associated with sub-collections of (ϕ, Φ). First addressing the tractable component, the vector-valued function $\phi : \mathcal{X}^m \to \mathbb{R}^{d_T}$ has an associated vector of canonical parameters $\theta \in \mathbb{R}^{d_T}$. Turning to the intractable component, for each $i = 1, \ldots, d_I$, the function Φ^i maps from $\mathcal{X}^m$ to $\mathbb{R}^b$, and the vector $\widetilde{\theta}^i \in \mathbb{R}^b$ is the associated set of canonical parameters. Overall, the function $\Phi = (\Phi^1, \ldots, \Phi^{d_I})$ maps from $\mathcal{X}^m$ to $\mathbb{R}^{b \times d_I}$, and has the associated canonical parameter vector $\widetilde{\theta} \in \mathbb{R}^{b \times d_I}$, partitioned as $\widetilde{\theta} = \big(\widetilde{\theta}^1, \widetilde{\theta}^2, \ldots, \widetilde{\theta}^{d_I}\big)$. These families of sufficient statistics define the exponential family

$$p(x; \theta, \widetilde{\theta}) \propto f_0(x) \exp\big(\langle \theta, \phi(x) \rangle\big) \, \exp\big(\langle \widetilde{\theta}, \Phi(x) \rangle\big)$$

$$= f_0(x) \exp\big(\langle \theta, \phi(x) \rangle\big) \prod_{i=1}^{d_I} \exp(\langle \widetilde{\theta}^i, \Phi^i(x) \rangle). \quad (4.58)$$

We say that any density p of the form (4.58) belongs to the (ϕ, Φ)-exponential family.

Next, we define the *base model*

$$p(x; \theta, \vec{0}) \propto f_0(x) \exp\big(\langle \theta, \phi(x) \rangle\big), \quad (4.59)$$

in which the intractable sufficient statistics Φ play no role, since we have set $\widetilde{\theta} = \vec{0}$. We say that any distribution p of the form (4.59) belongs to the ϕ-exponential family. Similarly, for each index $i \in \{1, \ldots, d_I\}$, we define the Φ^i-*augmented distribution*

$$p(x; \theta, \widetilde{\theta}^i) \propto f_0(x) \exp\big(\langle \theta, \phi(x) \rangle\big) \, \exp(\langle \widetilde{\theta}^i, \Phi^i(x) \rangle), \quad (4.60)$$

in which only a single term Φ^i has been introduced, and we say that any p of the form (4.60) belongs to the (ϕ, Φ^i)-exponential family.

The basic premises in the tractable–intractable partitioning between ϕ and Φ are:

- First, it is possible to compute marginals exactly in polynomial time for distributions of the base form (4.59) — that is, for any member of the ϕ-exponential family.
- Second, for each index $i = 1, \ldots, d_I$, exact polynomial-time computation is also possible for any distribution of the Φ^i-augmented form (4.60) — that is, for any member of the (ϕ, Φ^i)-exponential family.
- Third, it is intractable to perform exact computations in the full (ϕ, Φ)-exponential family (4.58), since it simultaneously incorporates *all* of the terms $(\Phi^1, \ldots, \Phi^{d_I})$.

Example 4.8 (Tractable/intractable Partitioning for Mixture Models). Let us illustrate the partitioning scheme (4.58) with the example of a Gaussian mixture model. Suppose that the random vector $X \in \mathbb{R}^m$ has a multivariate Gaussian distribution, $N(0, \Sigma)$. Letting $\varphi(y; \mu, \Lambda)$ denote the density of a random vector with a $N(\mu, \Lambda)$ distribution, consider the two-component Gaussian mixture model

$$p(y \mid X = x) = (1 - \alpha)\, \varphi(y; 0, \sigma_0^2 I) + \alpha\, \varphi(y; x, \sigma_1^2 I), \quad (4.61)$$

where $\alpha \in (0, 1)$ is the mixing weight, σ_0^2 and σ_1^2 are the variances, and I is the $m \times m$ identity matrix.

Given n i.i.d. samples $y^1, \ldots, y^n$ from the mixture density (4.61), it is frequently of interest to compute marginals under the posterior distribution of X conditioned on $(y^1, \ldots, y^n)$. Assuming a multivariate Gaussian prior $X \sim N(0, \Sigma)$, and using Bayes' theorem, we find that the posterior takes the form:

$$p(x \mid y^1 \ldots, y^n) \propto \exp\left(-\frac{1}{2} x^T \Sigma^{-1} x\right) \prod_{i=1}^{n} p(y^i \mid X = x)$$

$$= \exp\left(-\frac{1}{2} x^T \Sigma^{-1} x\right) \exp\left\{\sum_{i=1}^{n} \log p(y^i \mid X = x)\right\}.$$

$$(4.62)$$

To cast this model as a special case of the partitioned exponential family (4.58), we first observe that the term $\exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right)$ can be identified with the base term $f_0(x)\exp(\langle\theta,\phi(x)\rangle)$, so that we have $d_T = m$. On the other hand, suppose that we define $\Phi^i(x) := \log p(y^i \mid X = x)$ for each $i = 1,\dots,n$. Since the observation y^i is a fixed quantity, this definition makes sense. With this definition, the term $\exp\{\sum_{i=1}^n \log p(y^i \mid X = x)\}$ corresponds to the product $\prod_{i=1}^{d_I} \exp(\langle\widetilde{\theta}^i, \Phi^i(x)\rangle)$ in Equation (4.58). Note that we have $d_I = n$ and $b = 1$, with $\widetilde{\theta}^i = 1$ for all $i = 1,\dots,n$.

As a particular case of the (ϕ, Φ)-exponential family in this setting, the base distribution $p(x;\theta,\vec{0}) \propto \exp(-\frac{1}{2}x^T\Sigma^{-1}x)$ corresponds to a multivariate Gaussian, for which exact calculations of marginals is possible in $\mathcal{O}(m^3)$ time. Similarly, for each $i = 1,\dots,n$, the Φ^i-augmented distribution (4.60) is proportional to

$$\exp\left(-\frac{1}{2}x^T\Sigma^{-1}\right)[(1-\alpha)\,\varphi(y^i;0,\sigma_0^2 I) + \alpha\,\varphi(y^i;x,\sigma_1^2 I)], \qquad (4.63)$$

This distribution is a Gaussian mixture with two components, so that it is also possible to compute marginals exactly in cubic time. However, the full distribution (4.62) is a Gaussian mixture with 2^n components, so that the complexity of computing exact marginals is exponential in the problem size.

Returning to the main thread, let us develop a few more features of the distribution of interest (4.48). Since it is an exponential family, it has a partitioned set of mean parameters $(\mu,\widetilde{\mu}) \in \mathbb{R}^{d_T} \times \mathbb{R}^{d_I \times b}$, where

$$\mu = \mathbb{E}[\phi(X)], \quad \text{and} \quad (\widetilde{\mu}^1,\dots,\widetilde{\mu}^{d_I}) = \mathbb{E}[\Phi^1(X),\dots,\Phi^{d_I}(X)].$$

As an exponential family, the general variational principle from Theorem 3.4 is applicable. As usual, the relevant quantities in the variational principle (3.45) are the set

$$\mathcal{M}(\phi,\Phi) := \left\{(\mu,\widetilde{\mu}) \mid (\mu,\widetilde{\mu}) = \mathbb{E}_p[(\phi(X),\Phi(X))] \quad \text{for some } p\right\}, \tag{4.64}$$

and the entropy, or negative dual function $H(\mu,\widetilde{\mu}) = -A^*(\mu,\widetilde{\mu})$. Given our assumption that exact computation under the full distribution (4.48) is intractable, there must be challenges associated with

characterizing the mean parameter space (4.64) and/or the entropy function. Accordingly, we now describe a natural approximation to these quantities, based on the partitioned structure of the distribution (4.48), that leads to the class of expectation-propagation algorithms.

Associated with the base distribution (4.59) is the set

$$\mathcal{M}(\phi) := \left\{ \mu \in \mathbb{R}^{d_T} \mid \mu = \mathbb{E}_p[\phi(X)] \quad \text{for some density } p \right\}, \quad (4.65)$$

corresponding to the globally realizable mean parameters for the base distribution viewed as a d_T-dimensional exponential family. By Theorem 3.3, for any $\mu \in \mathcal{M}^\circ(\phi)$, there exists an exponential family member $p_{\theta(\mu)}$ that realizes it. Moreover, by our assumption that the base distribution is tractable, we can compute the entropy $H(\mu)$ of this distribution. Similarly, for each $i = 1, \ldots, d_I$, the Φ^i-augmented distribution (4.60) is associated with the mean parameter space $\mathcal{M}(\phi, \Phi^i)$

$$\left\{ (\mu, \widetilde{\mu}^i) \in \mathbb{R}^{d_T} \times \mathbb{R}^b \mid (\mu, \widetilde{\mu}^i) = \mathbb{E}_p[(\phi(X), \Phi^i(X))] \quad \text{for some density } p \right\}. \tag{4.66}$$

By similar reasoning as above, for any $(\mu, \widetilde{\mu}^i) \in \mathcal{M}^\circ(\phi, \Phi^i)$, there is a member of the (ϕ, Φ^i)-exponential family with these mean parameters. Furthermore, since we have assumed that the Φ^i-distribution (4.60) is tractable, it is easy to determine membership in the set $\mathcal{M}(\phi; \Phi^i)$, and moreover the entropy $H(\mu, \widetilde{\mu}^i)$ can be computed easily.

With these basic ingredients, we can now define an outer bound on the set $\mathcal{M}(\phi, \Phi)$. Given a candidate set of mean parameters $(\tau, \widetilde{\tau}) \in \mathbb{R}^{d_T} \times \mathbb{R}^{d_I \times b}$, we define for each $i = 1, 2, \ldots, d_I$ the coordinate projection operator $\Pi^i : \mathbb{R}^{d_T} \times \mathbb{R}^{d_I \times b} \to \mathbb{R}^{d_T} \times \mathbb{R}^b$ that operates as

$$(\tau, \widetilde{\tau}) \xrightarrow{\Pi^i} (\tau, \widetilde{\tau}^i) \in \mathbb{R}^{d_T} \times \mathbb{R}^b.$$

We then define the set

$$\mathcal{L}(\phi; \Phi) := \{ (\tau, \widetilde{\tau}) \mid \tau \in \mathcal{M}(\phi), \ \Pi^i(\tau, \widetilde{\tau}) \in \mathcal{M}(\phi, \Phi^i) \quad \forall i = 1, \ldots, d_I \}. \tag{4.67}$$

Note that $\mathcal{L}(\phi; \Phi)$ is a convex set, and moreover it is an outer bound on the true set $\mathcal{M}(\phi; \Phi)$ of mean parameters.

We now develop an entropy approximation that is tailored to the structure of $\mathcal{L}(\phi;\Phi)$. We begin by observing that for any $(\tau,\widetilde{\tau}) \in \mathcal{L}(\phi;\Phi)$ and for each $i = 1,\ldots,d_I$, there is a member of the (ϕ,Φ^i)-exponential family with mean parameters $(\tau,\widetilde{\tau}^i)$. This assertion follows because the projected mean parameters $(\tau,\widetilde{\tau}^i)$ belong to $\mathcal{M}(\phi;\Phi^i)$, so that Theorem 3.3 can be applied. We let $H(\tau,\widetilde{\tau}^i)$ denote the entropy of this exponential family member. Similarly, since $\tau \in \mathcal{M}(\phi)$ by definition of $\mathcal{L}(\phi;\Phi)$, there is a member of the ϕ-exponential family with mean parameter τ; we denote its entropy by $H(\tau)$. With these ingredients, we define the following *term-by-term entropy approximation*

$$H_{\mathrm{ep}}(\tau,\widetilde{\tau}) := H(\tau) + \sum_{\ell=1}^{d_I} \big[H(\tau,\widetilde{\tau}^\ell) - H(\tau)\big]. \qquad (4.68)$$

Combining this entropy approximation with the convex outer bound (4.67) yields the optimization problem

$$\max_{(\tau,\widetilde{\tau})\in\mathcal{L}(\phi;\Phi)} \big\{\langle\tau,\theta\rangle + \langle\widetilde{\tau},\widetilde{\theta}\rangle + H_{\mathrm{ep}}(\tau,\widetilde{\tau})\big\}. \qquad (4.69)$$

This optimization problem is an approximation to the exact variational principle from Theorem 3.4 for the (ϕ,Φ)-exponential family (4.48), and it underlies the family of expectation-propagation algorithms. It is closely related to the Bethe variational principle, as the examples to follow should clarify.

Example 4.9 (Sum-Product and Bethe Approximation). To provide some intuition, let us consider the Bethe approximation from the point of view of the variational principle (4.69). More specifically, we derive the Bethe entropy approximation (4.14) as a particular case of the term-by-term entropy approximation (4.68), and the tree-based outer bound $\mathbb{L}(G)$ from Equation (4.7) as a particular case of the convex outer bound (4.67) on $\mathcal{M}(\phi;\Phi)$.

Consider a pairwise Markov random field based on an undirected graph $G = (V,E)$, involving a discrete variable $X_s \in \{0,1,\ldots,r_s-1\}$ at each vertex $s \in V$; as we have seen, this can be expressed as an exponential family in the form (4.1) using the standard overcomplete parameterization (3.34) with indicator functions. Taking the point of

view of (4.69), we partition the sufficient statistics in the following way. The sufficient statistics associated with nodes are defined to be the tractable set, and those associated with edges are defined to be the intractable set. Thus, using the functions θ_s and θ_{st} defined in Equation (4.2), the base distribution (4.59) takes the form

$$p(x;\theta_1,\ldots,\theta_m,\vec{0}) \propto \prod_{s\in V} \exp(\theta_s(x_s)). \tag{4.70}$$

In this particular case, the terms Φ^i to be added correspond to the functions θ_{st} — that is, the index i runs over the edge set of the graph. For edge (u,v), the Φ^{uv}-augmented distribution (4.60) takes the form:

$$p(x;\theta_1,\ldots,\theta_m,\theta_{uv}) \propto \left[\prod_{s\in V} \exp(\theta_s(x_s)) \right] \exp\left(\theta_{uv}(x_u,x_v)\right). \tag{4.71}$$

The mean parameters associated with the standard overcomplete representation are singleton and pairwise marginal distributions, which we denote by $(\tau_s,\ s \in V)$ and τ_{uv} respectively. The entropy of the base distribution depends only on the singleton marginals, and given the product structure (4.70), takes the simple form

$$H(\tau_1,\ldots,\tau_m) = \sum_{s\in V} H(\tau_s),$$

where $H(\tau_s) = -\sum_{x_s} \tau_s(x_s)\log\tau_s(x_s)$ is the entropy of the marginal distribution. Similarly, since the augmented distribution (4.71) has only a single edge added and factorizes over a cycle-free graph, its entropy has the explicit form

$$H(\tau_1,\ldots,\tau_m,\tau_{uv}) = \sum_{s\in V} H(\tau_s) + \left[H(\tau_{uv}) - H(\tau_u) - H(\tau_v) \right]$$

$$= \sum_{s\in V} H(\tau_s) - I(\tau_{uv}),$$

where $H(\tau_{uv}) := -\sum_{x_u,x_v} \tau_{uv}(x_u,x_v)\log\tau_{uv}(x_u,x_v)$ is the joint entropy, and $I(\tau_{uv}) := H(\tau_u) + H(\tau_v) - H(\tau_{uv})$ is the mutual information. Putting together the pieces, we find that the term-by-term entropy approximation (4.68) — for this particular problem — has the form:

$$H_{\mathrm{ep}}(\tau) = \sum_{s\in V} H(\tau_s) - \sum_{(s,t)\in E} I_{st}(\tau_{st}),$$

which is precisely the Bethe entropy approximation defined previously (4.14).

Next, we show how the outer bound $\mathcal{L}(\phi; \Phi)$ defined in Equation (4.67) specializes to $\mathbb{L}(G)$. Consider a candidate set of local marginal distributions

$$\tau = \big(\tau_s, s \in V; \ \tau_{st}, (s,t) \in E\big).$$

In the current setting, the set $\mathcal{M}(\phi)$ corresponds to the set of all globally realizable marginals $(\tau_s, \ s \in V)$ under a factorized distribution, so that the inclusion $(\tau_s, \ s \in V) \in \mathcal{M}(\phi)$ is equivalent to the nonnegativity constraints $\tau_s(x_s) \geq 0$ for all $x_s \in \mathcal{X}_s$, and the local normalization constraints

$$\sum_{x_s} \tau_s(x_s) = 1 \quad \text{for all } s \in V. \tag{4.72}$$

Recall that the index i runs over edges of the graph; if, for instance $i = (u, v)$, then the projected marginals $\Pi^{uv}(\tau)$ are given by $(\tau_1, \ldots, \tau_m, \tau_{uv})$. The set $\mathcal{M}(\phi; \Phi^{uv})$ is traced out by all globally consistent marginals of this form, and is equivalent to the marginal polytope $\mathbb{M}(G_{uv})$, where G_{uv} denotes the graph with a single edge (u, v). Since this graph is a tree, the inclusion $\Pi^{uv}(\tau) \in \mathcal{M}(\phi; \Phi^{uv})$ is equivalent to having $\Pi^{uv}(\tau)$ satisfy — in addition to the nonnegativity and local normalization constraints (4.72) — the marginalization conditions

$$\sum_{x_v} \tau_{uv}(x_u, x_v) = \tau_u(x_u), \quad \text{and} \quad \sum_{x_u} \tau_{uv}(x_u, x_v) = \tau_v(x_v).$$

Therefore, the full collection of inclusions $\Pi^{uv}(\tau) \in \mathcal{M}(\phi; \Phi^{uv})$, as (u, v) runs over the graph edge set E, specify the same conditions defining the first-order relaxed constraint set $\mathbb{L}(G)$ from Equation (4.7).

4.3.2 Optimality in Terms of Moment-Matching

Returning to the main thread, we now derive a Lagrangian method for attempting to solve the expectation-propagation variational principle (4.69). As we will show, these Lagrangian updates reduce to moment-matching, so that the usual expectation-propagation updates are recovered.

Our Lagrangian formulation is based on the following two steps:

- First, we augment the space of pseudo-mean-parameters over which we optimize, so that the original constraints defining $\mathcal{L}(\phi; \Phi)$ are decoupled.
- Second, we add new constraints — to be penalized with Lagrange multipliers — so as to enforce the remaining constraints required for membership in $\mathcal{L}(\phi; \Phi)$.

Beginning with the augmentation step, let us duplicate the vector $\tau \in \mathbb{R}^{d_T}$ a total of d_I times, defining thereby defining d_I new vectors $\eta^i \in \mathbb{R}^{d_T}$ and imposing the constraint that $\eta^i = \tau$ for each index $i = 1, \ldots, d_I$. This yields a large collection of pseudo-mean-parameters

$$\{\tau, (\eta^i, \widetilde{\tau}^i), i = 1, \ldots, d_I\} \in \mathbb{R}^{d_T} \times (\mathbb{R}^{d_T} \times \mathbb{R}^b)^{d_I},$$

and we recast the variational principle (4.69) using these pseudo-mean-parameters as follows:

$$\max_{\{\tau, (\eta^i, \widetilde{\tau}^i)\}} \left\{ \langle \tau, \theta \rangle + \sum_{i=1}^{d_I} \langle \widetilde{\tau}^i, \widetilde{\theta}^i \rangle + \underbrace{H(\tau) + \sum_{i=1}^{d_I} \left[H(\eta^i, \widetilde{\tau}^i) - H(\eta^i) \right]}_{F(\tau; (\eta^i, \widetilde{\tau}^i))} \right\}, \tag{4.73}$$

subject to the constraints $(\eta^i, \widetilde{\tau}^i) \in \mathcal{M}(\phi; \Phi^i)$ for all $i = 1, \ldots, d_I$, and

$$\tau = \eta^i \qquad \forall i = 1, \ldots, d_I. \tag{4.74}$$

Let us now consider a particular iterative scheme for solving the reformulated problem (4.73). In particular, for each $i = 1, \ldots, d_I$, define a vector of Lagrange multipliers $\lambda^i \in \mathbb{R}^{d_T}$ associated with the constraints $\tau = \eta^i$. We then form the Lagrangian function

$$L(\tau; \lambda) = \langle \tau, \theta \rangle + \sum_{i=1}^{d_I} \langle \widetilde{\tau}^i, \widetilde{\theta}^i \rangle + F(\tau; (\eta^i, \widetilde{\tau}^i)) + \sum_{i=1}^{d_I} \langle \lambda^i, \tau - \eta^i \rangle. \tag{4.75}$$

This Lagrangian is a partial one, since we are enforcing the constraints $\tau \in \mathcal{M}(\phi)$ and $(\eta^i, \widetilde{\tau}^i) \in \mathcal{M}(\phi; \Phi^i)$ explicitly.

Consider an optimum solution $\{\tau, (\eta^i, \widetilde{\tau}^i), i = 1, \ldots, d_I\}$ of the optimization problem (4.73) that satisfies the following properties: (a) the vector τ belongs to the (relative) interior $\mathcal{M}^\circ(\phi)$, and (b) for each $i = 1, \ldots, d_I$, the vector $(\eta^i, \widetilde{\tau}^i)$ belongs to the relative interior $\mathcal{M}^\circ(\phi; \Phi^i)$. Any such solution must satisfy the zero-gradient conditions associated with the partial Lagrangian (4.75) — namely

$$\nabla_\tau L(\tau; \lambda) = 0, \tag{4.76a}$$

$$\nabla_{(\eta^i, \widetilde{\tau}^i)} L(\tau; \lambda) = 0 \quad \text{for } i = 1, \ldots, d_I, \text{ and} \tag{4.76b}$$

$$\nabla_\lambda L(\tau; \lambda) = 0. \tag{4.76c}$$

Since the vector τ belongs to $\mathcal{M}^\circ(\phi)$, it specifies a distribution in the ϕ-exponential family. By explicitly computing the Lagrangian condition (4.76a) and performing some algebra — essentially the same steps as the proof of Theorem 4.2 — we find that this exponential family member can be written, in terms of the original parameter vector θ and the Lagrange multipliers λ, as follows:

$$q(x; \theta, \lambda) \propto f_0(x) \exp\left\{ \left\langle \theta + \sum_{i=1}^{d_I} \lambda^i, \, \phi(x) \right\rangle \right\}. \tag{4.77}$$

Similarly, since for each $i = 1, \ldots, d_I$, the vector $(\tau, \widetilde{\tau}^i)$ belongs to $\mathcal{M}^\circ(\phi; \Phi^i)$, it also specifies a distribution in the (ϕ, Φ^i)-exponential family. Explicitly computing the condition (4.76b) and performing some algebra shows that this distribution can be expressed as

$$q^i(x; \theta, \widetilde{\theta}^i, \lambda) \propto f_0(x) \exp\left(\left\langle \theta + \sum_{\ell \neq i} \lambda^\ell, \, \phi(x) \right\rangle + \langle \widetilde{\theta}^i, \Phi^i(x) \rangle \right), \tag{4.78}$$

The final Lagrangian condition (4.76c) ensures that the constraints (4.74) are satisfied. Noting that $\tau = \mathbb{E}_q[\phi(X)]$ and $\eta^i = \mathbb{E}_{q^i}[\phi(X)]$, these constraints reduce to the moment-matching conditions

$$\int q(x; \theta, \lambda) \phi(x) \nu(dx) = \int q^i(x; \theta, \widetilde{\theta}^i, \lambda) \phi(x) \nu(dx), \tag{4.79}$$

for $i = 1, \ldots, d_I$.

On the basis of Equations (4.77), (4.78), and (4.79), we arrive at the expectation-propagation updates, as summarized in Figure 4.7.

Expectation-propagation (EP) updates:

  (1) At iteration $n = 0$, initialize the Lagrange multiplier vectors $(\lambda^1, \ldots, \lambda^{d_I})$.

  (2) At each iteration, $n = 1, 2, \ldots$, choose some index $i(n) \in \{1, \ldots, d_I\}$, and

  (a) Using Equation (4.78), form the augmented distribution $q^{i(n)}$ and compute the mean parameter

$$\eta^{i(n)} := \int q^{i(n)}(x)\phi(x)\nu(dx) = \mathbb{E}_{q^{i(n)}}[\phi(X)]. \quad (4.80)$$

  (b) Using Equation (4.77), form the base distribution q and adjust $\lambda^{i(n)}$ to satisfy the moment-matching condition

$$\mathbb{E}_q[\phi(X)] = \eta^{i(n)}. \quad (4.81)$$

Fig. 4.7 Steps involved in the expectation-propagation updates. It is a Lagrangian algorithm for attempting to solve the Bethe-like approximation (4.73), or equivalently (4.69).

We note that the algorithm is well defined, since for any realizable mean parameter τ, there is always a solution for $\lambda^{i(n)}$ in Equation (4.81). This fact can be seen by considering the d_T-dimensional exponential family defined by the pair $(\lambda^{i(n)}, \phi)$, and then applying Theorem 3.3. Moreover, it follows that any fixed point of these EP updates satisfies the necessary Lagrangian conditions for optimality in the program (4.73). From this fact, we make an important conclusion — namely, fixed points of the EP updates are guaranteed to exist, assuming that the optimization problem (4.73) has at least one optimum. As with the sum-product algorithm and the Bethe variational principle, there are no guarantees that the EP updates converge in general. However, it would be relatively straightforward to develop convergent algorithms for finding at least a local optimum of the variational problem (4.73), or equivalently (4.69), possibly along the lines of convergent algorithms developed for the ordinary Bethe variational problem or convexified versions thereof [111, 254, 270, 108].

At a high level, the key points to take away are the following: within the variational framework, expectation-propagation algorithms are based on a Bethe-like entropy approximation (4.68), and a particular convex outer bound on the set of mean parameters (4.67). Moreover, the moment-matching steps in the EP algorithm arise from a Lagrangian approach for attempting to solve the relaxed variational principle (4.69).

We conclude by illustrating the specific forms taken by the algorithm in Figure 4.7 for some concrete examples:

Example 4.10 (Sum-Product as Moment-matching). As a continuation of Example 4.9, we now show how the updates (4.80) and (4.81) of Figure 4.7 reduce to the sum-product updates. In this particular example, the index i ranges over edges of the graph. Associated with $i = (u, v)$ are the pair of Lagrange multiplier vectors $(\lambda_{uv}(x_v), \lambda_{vu}(x_u))$. In terms of these quantities, the base distribution (4.77) can be written as

$$q(x; \theta, \lambda) \propto \prod_{s \in V} \exp(\theta_s(x_s)) \prod_{(u,v) \in E} \exp\big(\lambda_{uv}(x_v) + \lambda_{vu}(x_u)\big)$$

$$= \prod_{s \in V} \exp\Big(\theta_s(x_s) + \sum_{t \in N(s)} \lambda_{ts}(x_s)\Big), \qquad (4.82)$$

or more compactly as $q(x; \theta, \lambda) \propto \prod_{s \in V} \tau_s(x_s)$, where we have defined the pseudo-marginals

$$\tau_s(x_s) \propto \exp\Big(\theta_s(x_s) + \sum_{t \in N(s)} \lambda_{ts}(x_s)\Big).$$

It is worthwhile noting the similarity to the sum-product expression (4.23) for the singleton pseudomarginals τ_s, obtained in the proof of Theorem 4.2. Similarly, when $\ell = (u, v)$, the augmented distribution (4.78) is given by

$$q^{(u,v)}(x; \theta; \lambda) \propto q(x; \theta, \lambda) \exp\big(\theta_{uv}(x_u, x_v) - \lambda_{vu}(x_u) - \lambda_{uv}(x_v)\big)$$

$$= \Big[\prod_{s \in V} \tau_s(x_s)\Big] \exp\big(\theta_{uv}(x_u, x_v) - \lambda_{vu}(x_u) - \lambda_{uv}(x_v)\big).$$

$$(4.83)$$

Now consider the updates (4.80) and (4.81). If index $i = (u,v)$ is chosen, the update (4.80) is equivalent to computing the singleton marginals of the distribution (4.83). The update (4.81) dictates that the Lagrange multipliers $\lambda_{uv}(x_v)$ and $\lambda_{vu}(x_u)$ should be adjusted so that the marginals $\{\tau_u, \tau_v\}$ of the distribution (4.82) match these marginals. Following a little bit of algebra, these fixed point conditions reduce to the sum-product updates along edge (u,v), where the messages are given as exponentiated Lagrange multipliers by $M_{uv}(x_v) = \exp(\lambda_{uv}(x_v))$.

From the perspective of term-by-term approximations, the sum-product algorithm uses a product base distribution (see Equation (4.70)). A natural extension, then, is to consider a base distribution with more structure.

Example 4.11 (Tree-structured EP). We illustrate this idea by deriving the tree-structured EP algorithm [174], applied to a pairwise Markov random field on a graph $G = (V, E)$. We use the same set-up and parameterization as in Examples 4.9 and 4.10. Given some fixed spanning tree $T = (V, E(T))$ of the graph, the base distribution (4.59) is given by

$$p(x; \theta, \vec{0}) \propto \prod_{s \in V} \exp(\theta_s(x_s)) \prod_{(s,t) \in E(T)} \exp\big(\theta_{st}(x_s, x_t)\big). \quad (4.84)$$

In this case, the index i runs over edges $(u,v) \in E \backslash E(T)$; given such an edge (u,v), the Φ^{uv}-augmented distribution (4.60) is obtained by adding in the term θ_{uv}:

$$p(x; \theta, \theta_{uv}) \propto p(x; \theta, \vec{0}) \exp\big(\theta_{uv}(x_u, x_v)\big). \quad (4.85)$$

Let us now describe the analogs of the sets $\mathcal{M}(\phi; \Phi)$, $\mathcal{M}(\phi)$, and $\mathcal{M}(\phi; \Phi^i)$ for this setting. First, as we saw in Example 4.9, the subvectro of mean parameters associated with the full model (4.48) is given by

$$\mu := \big(\mu_s, s \in V; \mu_{st}, (s,t) \in E\big).$$

Accordingly, the analog of $\mathcal{M}(\phi; \Phi)$ is the marginal polytope $\mathbb{M}(G)$. Second, if we use the tree-structured distribution (4.84) as the base

distribution, the relevant subvector of mean parameters is

$$\mu(T) := \big(\mu_s, s \in V; \, \mu_{st}, (s,t) \in E(T)\big).$$

The analog of the set $\mathcal{M}(\phi)$ is the marginal polytope $\mathbb{M}(T)$, or equivalently — using Proposition 4.1 — the set $\mathbb{L}(T)$. Finally, if we add in edge $i = (u,v) \notin E(T)$, then the analog of the set $\mathcal{M}(\phi; \Phi^i)$ is the marginal polytope $\mathbb{M}(T \cup (u,v))$.

The analog of the set $\mathcal{L}(\phi; \Phi)$ is an interesting object, which we refer to as the *tree-EP outer bound* on the marginal polytope $\mathbb{M}(G)$. It consists of all vectors $\tau = \big(\tau_s, s \in V; \, \tau_{st}, (s,t) \in E\big)$ such that

- (a) the inclusion $\tau(T) \in \mathbb{M}(T)$ holds, and
- (b) for all $(u,v) \notin T$, the inclusion $(\tau(T), \tau_{uv}) \in \mathbb{M}(T \cup (u,v))$ holds.

Observe that the set thus defined is contained within $\mathbb{L}(G)$: in particular, the condition $\tau(T) \in \mathbb{M}(T)$ ensures nonnegativity, normalization, and marginalization for all mean parameters associated with the tree, and the inclusion $(\tau(T), \tau_{uv}) \in \mathbb{M}(T \cup (u,v))$ ensures that τ_{uv} is nonnegative, and satisfies the marginalization constraints associated with τ_u and τ_v. For graphs with cycles, this tree-EP outer bound is strictly contained within $\mathbb{L}(G)$; for instance, when applied to the single cycle C_3 on three nodes, the tree-EP outer bound is equivalent to the marginal polytope [240]. But the set $\mathbb{M}(C_3)$ is strictly contained within $\mathbb{L}(C_3)$, as shown in Example 4.1.

The entropy approximation H_{ep} associated with tree-EP is easy to define. Given a candidate set of pseudo-mean-parameters τ in the tree-EP outer bound, we define the tree-structured entropy associated with the base distribution (4.84)

$$H(\tau(T)) := \sum_{s \in V} H(\tau_s) - \sum_{(s,t) \in E(T)} I_{st}(\tau_{st}). \qquad (4.86)$$

Similarly, for each edge $(u,v) \notin E(T)$, we define the entropy $H(\tau(T), \tau_{uv})$ associated with the augmented distribution (4.85); unlike the base case, this entropy does not have an explicit form (since the graph is not in junction tree form), but it can be computed easily.

The tree-EP entropy approximation then takes the form:

$$H_{\mathrm{ep}}(\tau) = H(\tau(T)) + \sum_{(u,v)\notin E(T)} \big[H(\tau(T),\tau_{uv}) - H(\tau(T)) \big].$$

Finally, let us derive the moment-matching updates (4.80) and (4.81) for tree-EP. For each edge $(u,v) \notin E(T)$, let $\lambda^{uv}(T)$ denote a vector of Lagrange multipliers, of the same dimension as $\tau(T)$, and let $\phi(x;T)$ denote the subset of sufficient statistics associated with T. The optimized base distribution (4.77) can be represented in terms of the original base distribution (4.84) and these tree-structured Lagrange multipliers as

$$q(x;\theta,\lambda) \propto p(x;\theta,\vec{0}) \prod_{(u,v)\notin E(T)} \exp(\langle \lambda^{uv}(T), \phi(x;T)\rangle).$$

If edge (u,v) is added, the augmented distribution (4.78) is given by

$$q^{uv}(x;\theta,\lambda) \propto p(x;\theta,\vec{0}) \exp(\theta_{uv}(x_u,x_v)) \prod_{(s,t)\neq(u,v)} \exp(\langle \lambda^{uv}(T), \phi(x;T)\rangle).$$

For a given edge $(u,v) \notin E(T)$, the moment-matching step (4.81) corresponds to computing singleton marginals and pairwise marginals for edges $(s,t) \in E(T)$ under the distribution $q^{uv}(x;\theta,\lambda)$. The step (4.81) corresponds to updating the Lagrange multiplier vector $\lambda^{uv}(T)$ so the marginals of $q(x;\theta;\lambda)$ agree with those of $q^{uv}(x;\theta;\lambda)$ for all nodes $s \in V$, and all edges $(s,t) \in E(T)$. See the papers [174, 252, 240] for further details on tree EP.

The previous examples dealt with discrete random variables; we now return to the case of a mixture model with continuous variables, first introduced in Example 4.8.

Example 4.12 (EP for Gaussian Mixture Models). Recall from Equation (4.62) the representation of the mixture distribution as an exponential family with the base potentials $\phi(x) = (x; xx^T)$ and auxiliary terms $\Phi^i(x) = \log p(y^i \mid x)$, for $i = 1,\ldots,n$. Turning to the analogs of the various mean parameter spaces, the set $\mathcal{M}(\phi;\Phi)$ is a subset of

$\mathbb{R}^m \times \mathcal{S}_+^m \times \mathbb{R}^n$, corresponding the space of all globally realizable mean parameters partitioned as

$$\Big(\mathbb{E}[X]; \ \mathbb{E}[XX^T]; \ \mathbb{E}[\log p(y^i \mid X)], \quad i = 1, \ldots, n \Big).$$

Recall that $(y^1, y^2, \ldots, y^n)$ are observed, and hence remain fixed throughout the derivation.

The set $\mathcal{M}(\phi)$ is a subset of $\mathbb{R}^m \times \mathcal{S}_+^m$, corresponding to the mean parameter space for a multivariate Gaussian, as characterized in Example 3.7. For each $i = 1, \ldots, n$, the constraint set $\mathcal{M}(\phi; \Phi^i)$ is a subset of $\mathbb{R}^m \times \mathcal{S}_+^m \times \mathbb{R}$, corresponding to the collection of all globally realizable mean parameters with the partitioned form

$$\Big(\mathbb{E}[X], \ \mathbb{E}[XX^T], \ \mathbb{E}[\log p(y^i \mid X)] \Big).$$

Turning to the various entropies, the base term is the entropy $H(\mu)$ of a multivariate Gaussian where μ denotes the pair $(\mathbb{E}[X], \mathbb{E}[XX^T])$; similarly, the augmented term $H(\mu, \widetilde{\mu}^i)$ is the entropy of the exponential family member with sufficient statistics $(\phi(x), \log p(y^i \mid x))$, and mean parameters $(\mu, \widetilde{\mu}^i)$, where $\widetilde{\mu}^i := \mathbb{E}[\log p(y^i \mid X)]$. Using these quantities, we can define the analog of the variational principle (4.69).

Turning to the moment-matching steps (4.80) and (4.81), the Lagrangian version of Gaussian-mixture EP algorithm can be formulated in terms of a collection of n matrix-vector Lagrange multiplier pairs

$$(\lambda^i, \Lambda^i) \in \mathbb{R}^m \times \mathbb{R}^{m \times m}, \quad i = 1, \ldots, n,$$

one for each augmented distribution. Since $X \sim N(0, \Sigma)$ by definition, the optimized base distribution (4.77) can be written in terms of Σ^{-1} and the Lagrange multipliers

$$q(x; \Sigma; (\lambda, \Lambda)) \propto \exp\bigg\{ \langle \sum_{i=1}^n \lambda^i, x \rangle + \langle\!\langle -\frac{1}{2}\Sigma^{-1} + \sum_{i=1}^n \Lambda^i, \ xx^T \rangle\!\rangle \bigg\}. \ (4.87)$$

Note that this is simply a multivariate Gaussian distribution. The augmented distribution (4.78) associated with any term $i \in \{1, 2, \ldots, d_I\}$,

denoted by q^i, takes the form

$$f_0(x) \exp\left\{ \langle \sum_{\ell \neq i} \lambda^\ell, x \rangle + \langle\!\langle -\frac{1}{2}\Sigma^{-1} + \sum_{\ell \neq i} \Lambda^\ell, \, xx^T \rangle\!\rangle + \langle \widetilde{\theta}_i, \log p(y^i \mid x) \rangle \right\}.$$

(4.88)

With this set-up, the step (4.80) corresponds to computing the mean parameters $(\mathbb{E}_{q^i}[X], \mathbb{E}_{q^i}[XX^T])$ under the distribution (4.88), and step (4.81) corresponds to adjusting the multivariate Gaussian (4.87) so that it has these mean parameters. Fixed points of these moment-matching updates satisfy the necessary Lagrangian conditions to be optima of the associated Bethe-like approximation of the exact variational principle.

We refer the reader to Minka [172, 175] and Seeger [213] for further details of the Gaussian-mixture EP algorithm and some of its properties.

5

Mean Field Methods

This section is devoted to a discussion of mean field methods, which originated in the statistical physics literature [e.g., 12, 47, 190]. From the perspective of this survey, the mean field approach is based on a specific type of approximation to the exact variational principle (3.45). More specifically, as discussed in Section 3.7, there are two fundamental difficulties associated with the variational principle (3.45): the nature of the constraint set $\mathcal{M}$, and the lack of an explicit form for the dual function A^*. The core idea of mean field approaches is simple: let us limit the optimization to a subset of distributions for which both $\mathcal{M}$ and A^* are relatively easy to characterize; e.g., perhaps they correspond to a graph with small treewidth. Throughout this section, we refer to any such distribution as "tractable." The simplest choice is the family of product distributions, which gives rise to the naive mean field method. Higher-order mean field methods are obtained by choosing tractable distributions with more structure.

5.1   Tractable Families

Given a graphical model based on a graph G, we base our description of mean field methods on the notion of a *tractable subgraph*, by which

we mean a subgraph F of the graph G over which it is feasible to perform exact calculations. The simplest example of a tractable subgraph is the fully disconnected subgraph $F_0 = (V, \emptyset)$, which contains all the vertices of G but *none* of the edges. Any distribution that is Markov with respect to F is then a product distribution, for which exact computations are trivial.

A bit more generally, consider an exponential family with a collection $\phi = (\phi_\alpha, \ \alpha \in \mathcal{I})$ of sufficient statistics associated with the cliques of $G = (V, E)$. Given a subgraph F, let $\mathcal{I}(F) \subseteq \mathcal{I}$ be the subset of sufficient statistics associated with cliques of F. The set of all distributions that are Markov with respect to F is a sub-family of the full ϕ-exponential family; it is parameterized by the subspace of canonical parameters

$$\Omega(F) := \{\theta \in \Omega \mid \theta_\alpha = 0 \quad \forall \ \alpha \in \mathcal{I}\backslash\mathcal{I}(F)\}. \tag{5.1}$$

We consider some examples to illustrate:

Example 5.1 (Tractable Subgraphs). Suppose that $\theta \in \Omega$ parameterizes a pairwise Markov random field, with potential functions associated with the vertices and edges of an undirected graph $G = (V, E)$. For each edge $(s, t) \in E$, let $\theta_{(s,t)}$ denote the subvector of parameters associated with sufficient statistics that depend only on (X_s, X_t). Consider the completely disconnected subgraph $F_0 = (V, \emptyset)$. With respect to this subgraph, permissible parameters must belong to the subspace

$$\Omega(F_0) := \{\theta \in \Omega \mid \theta_{(s,t)} = 0 \ \forall \ (s, t) \in E\}. \tag{5.2}$$

The densities in this sub-family are all of the *fully factorized or product form*

$$p_\theta(x) = \prod_{s \in V} p(x_s; \theta_s), \tag{5.3}$$

where θ_s refers to the subvector of canonical parameters associated with vertex s.

To obtain a more structured approximation, one could choose a spanning tree $T = (V, E(T))$. In this case, we are free to choose the canonical parameters corresponding to vertices and edges in the tree

T, but we must set to zero any canonical parameters corresponding to edges not in the tree. Accordingly, the subspace of tree-structured distributions is specified by the subset of canonical parameters

$$\Omega(T) := \big\{ \theta \in \Omega \mid \theta_{(s,t)} = 0 \ \forall \ (s,t) \notin E(T) \big\}. \tag{5.4}$$

Associated with the exponential family defined by ϕ and G is the set $\mathcal{M}(G;\phi)$ of all mean parameters realizable by any distribution, as previously defined in Equation (3.26). (Whereas our previous notation did not make explicit reference to G and ϕ, the definition of $\mathcal{M}$ does depend on these quantities and it is now useful to make this dependence explicit.) For a given tractable subgraph F, mean field methods are based on optimizing over the subset of mean parameters that can be obtained by the subset of exponential family densities $\{p_\theta, \ \theta \in \Omega(F)\}$ — namely

$$\mathcal{M}_F(G;\phi) := \{\mu \in \mathbb{R}^d \mid \mu = \mathbb{E}_\theta[\phi(x)] \quad \text{for some } \theta \in \Omega(F)\}. \tag{5.5}$$

In terms of the moment mapping from Theorem 3.3, a more compact definition of the set $\mathcal{M}_F(G;\phi)$ is as the image $\nabla A(\Omega(F))$. By Theorem 3.3, we have $\mathcal{M}^\circ(G;\phi) = \nabla A(\Omega)$ so that the inclusion

$$\mathcal{M}_F^\circ(G;\phi) \subseteq \mathcal{M}^\circ(G;\phi)$$

holds for any subgraph F. For this reason, we say that $\mathcal{M}_F$ is an *inner approximation* to the set $\mathcal{M}$ of realizable mean parameters.

To lighten the notation in the remainder of this section, we generally drop the ϕ term from $\mathcal{M}(G;\phi)$ and $\mathcal{M}_F(G;\phi)$, writing $\mathcal{M}(G)$ and $\mathcal{M}_F(G)$, respectively. It is important to keep in mind, though, that these sets do depend on the choice of sufficient statistics.

5.2 Optimization and Lower Bounds

We now have the necessary ingredients to develop the mean field approach to approximate inference. Suppose that we are interested in approximating some target distribution p_θ, where $\theta \in \Omega$. Mean field methods generate lower bounds on the value $A(\theta)$ of the cumulant function, as well as approximations to the mean parameters $\mu = \mathbb{E}_\theta[\phi(X)]$ of this target distribution p_θ.

5.2.1   Generic Mean Field Procedure

The key property of any mean field method is the following fact: any valid mean parameter specifies a lower bound on the log partition function.

Proposition 5.1 (Mean Field Lower Bound). Any mean parameter $\mu \in \mathcal{M}^\circ$ yields a lower bound on the cumulant function:

$$A(\theta) \geq \langle \theta, \mu \rangle - A^*(\mu). \tag{5.6}$$

Moreover, equality holds if and only if θ and μ are dually coupled (i.e., $\mu = \mathbb{E}_\theta[\phi(X)]$).

Proof. In convex analysis, this lower bound is known as *Fenchel's inequality*. It is an immediate consequence of the general variational principle (3.45), since the supremum over μ is always greater than the value at any particular μ. Moreover, the supremum is attained when $\mu = \nabla A(\theta)$, which means that θ and μ are dually coupled (by definition).

An alternative proof via Jensen's inequality is also possible, and given its popularity in the literature on mean field methods, we give a version of this proof here. By definition of $\mathcal{M}$, for any mean parameter $\mu \in \mathcal{M}$, there must exist some density q, defined with respect to the base measure ν underlying the exponential family, for which $\mathbb{E}_q[\phi(X)] = \mu$. We then have

$$
\begin{aligned}
A(\theta) &= \log \int_{\mathcal{X}^m} q(x) \frac{\exp\left\{\langle \theta, \phi(x) \rangle\right\}}{q(x)} \, \nu(dx) \\
&\overset{(a)}{\geq} \int_{\mathcal{X}^m} q(x) [\langle \theta, \phi(x) \rangle - \log q(x)] \nu(dx) \\
&\overset{(b)}{=} \langle \theta, \mu \rangle + H(q),
\end{aligned}
\tag{5.7}
$$

where $H(q) = -\mathbb{E}_q[\log q(X)]$ is the entropy of q. In this argument, step (a) follows from Jensen's inequality (see Appendix A.2.5) applied to the negative logarithm, whereas step (b) follows from the moment-matching condition $\mathbb{E}_q[\phi(X)] = \mu$. Since the lower bound (5.7) holds

for any density q satisfying this moment-matching condition, we may optimize over the choice of q: by Theorem 3.4, doing so yields the exponential family density $q^*(x) = p_{\theta(\mu)}(x)$, for which $H(q^*) = -A^*(\mu)$ by construction. $\qquad\square$

Since the dual function A^* typically lacks an explicit form, it is not possible, at least in general, to compute the lower bound (5.6). The mean field approach circumvents this difficulty by restricting the choice of μ to the tractable subset $\mathcal{M}_F(G)$, for which the dual function has an explicit form. For compactness in notation, we define $A_F^* = A^*\big|_{\mathcal{M}_F(G)}$, corresponding to the dual function restricted to the set $\mathcal{M}_F(G)$. As long as μ belongs to $\mathcal{M}_F(G)$, then the lower bound (5.6) involves A_F^*, and hence can be computed easily.

The next step of the mean field method is the natural one: find the best approximation, as measured in terms of the tightness of the lower bound (5.6). More precisely, the best lower bound from within $\mathcal{M}_F(G)$ is given by

$$\max_{\mu \in \mathcal{M}_F(G)} \big\{ \langle \mu, \theta \rangle - A_F^*(\mu) \big\}. \tag{5.8}$$

The corresponding value of μ is defined to be the mean field approximation to the true mean parameters.

In Section 5.3, we illustrate the use of this generic procedure in obtaining lower bounds and approximate mean parameters for various types of graphical models.

5.2.2 Mean Field and Kullback–Leibler Divergence

An important alternative interpretation of the mean field optimization problem (5.8) is as minimizing the Kullback–Leibler (KL) divergence between the approximating (tractable) distribution and the target distribution. In order to make this connection clear, we first digress to discuss various forms of the KL divergence for exponential family models.

The conjugate duality between A and A^*, as characterized in Theorem 3.4, leads to several alternative forms of the KL divergence for exponential family members. Given two distributions with densities q and p with respect to a base measure ν, the standard definition of the

KL divergence [57] is

$$D(q \parallel p) := \int_{\mathcal{X}^m} \left[\log \frac{q(x)}{p(x)} \right] q(x)\, \nu(dx). \tag{5.9}$$

The key result that underlies alternative representations for exponential families is Proposition 5.1.

Consider two canonical parameter vectors $\theta^1, \theta^2 \in \Omega$; with a slight abuse of notation, we use $D(\theta^1 \parallel \theta^2)$ to refer to the KL divergence between p_{θ^1} and p_{θ^2}. We use μ^1 and μ^2 to denote the respective mean parameters (i.e., $\mu^i = \mathbb{E}_{\theta^i}[\phi(X)]$ for $i = 1, 2$). A first alternative form of the KL divergence is obtained by substituting the exponential representations of p_{θ^i} into Equation (5.9), and then expanding and simplifying as follows:

$$\begin{aligned} D(\theta^1 \parallel \theta^2) &= \mathbb{E}_{\theta^1}\left[\log \frac{p_{\theta^1}(x)}{p_{\theta^2}(x)} \right] \\ &= A(\theta^2) - A(\theta^1) - \langle \mu^1, \theta^2 - \theta^1 \rangle. \end{aligned} \tag{5.10}$$

We refer to this representation as the *primal form* of the KL divergence. As illustrated in Figure 5.1, this form of the KL divergence can be interpreted as the difference between $A(\theta^2)$ and the hyperplane tangent to A at θ^1 with normal $\nabla A(\theta^1) = \mu^1$. This interpretation shows that the KL divergence is a particular example of a Bregman distance [42, 45].

A second form of the KL divergence can be obtained by using the fact that equality holds in Proposition 5.1 for dually coupled parameters. In particular, given a pair (θ^1, μ^1) for which $\mu^1 = \mathbb{E}_{\theta^1}[\phi(X)]$,
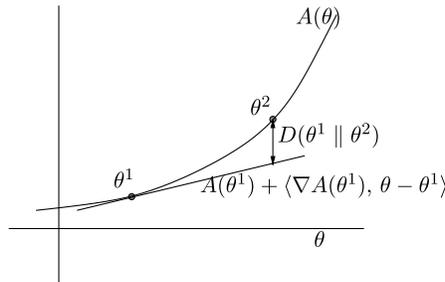

Fig. 5.1 The hyperplane $A(\theta^1) + \langle \nabla A(\theta^1), \theta - \theta^1 \rangle$ supports the epigraph of A at θ^1. The Kullback–Leibler divergence $D(\theta^1 \parallel \theta^2)$ is equal to the difference between $A(\theta^2)$ and this tangent approximation.

we can transform Equation (5.10) into the following *mixed form* of the KL divergence:

$$D(\theta^1 \parallel \theta^2) \equiv D(\mu^1 \parallel \theta^2) = A(\theta^2) + A^*(\mu^1) - \langle \mu^1, \theta^2 \rangle. \qquad (5.11)$$

Note that this mixed form of the divergence corresponds to the slack in the inequality (5.6). It also provides an alternative view of the variational representation given in Theorem 3.4(b). In particular, Equation (3.45) can be rewritten as follows:

$$\min_{\mu \in \mathcal{M}} \left\{ A(\theta) + A^*(\mu) - \langle \theta, \mu \rangle \right\} = 0.$$

Using Equation (5.11), the variational representation in Theorem 3.4(b) is seen to be equivalent to the assertion that $\min_{\mu \in \mathcal{M}} D(\mu \parallel \theta) = 0$.

Finally, by applying Equation (5.6) as an equality once again, this time for the coupled pair (θ^2, μ^2), the mixed form (5.11) can be transformed into a purely *dual form* of the KL divergence:

$$
\begin{aligned}
D(\theta^1 \parallel \theta^2) &\equiv D(\mu^1 \parallel \mu^2) \\
&= A^*(\mu^1) - A^*(\mu^2) - \langle \theta^2, \mu^1 - \mu^2 \rangle. \qquad (5.12)
\end{aligned}
$$

Note the symmetry between representations (5.10) and (5.12). This form of the KL divergence has an interpretation analogous to that of Figure 5.1, but with A replaced by the dual A^*.

With this background on the Kullback-Leibler divergence, let us now return to the consequences for mean field methods. For a given mean parameter $\mu \in \mathcal{M}_F(G)$, the difference between the log partition function $A(\theta)$ and the quantity $\langle \mu, \theta \rangle - A_F^*(\mu)$ to be maximized is equivalent to

$$D(\mu \parallel \theta) = A(\theta) + A_F^*(\mu) - \langle \mu, \theta \rangle,$$

corresponding to the mixed form of the KL divergence defined in Equation (5.11). On the basis of this relation, it can be seen that solving the variational problem (5.8) is equivalent to minimizing the KL divergence $D(\mu \parallel \theta)$, subject to the constraint that $\mu \in \mathcal{M}_F(G)$. Consequently, any mean field method can be understood as obtaining the "best" approximation to p_θ from a family of tractable models, where approximation quality is measured by the KL divergence.

5.3   Naive Mean Field Algorithms

In this section, we illustrate various instances of mean field algorithms for particular graphical models. Our examples in this section focus on the simplest type of approximation, referred to as the naive mean field approach. It is based on choosing a *product distribution*

$$p_\theta(x_1, x_2, \ldots, x_m) := \prod_{s \in V} p(x_s; \theta_s) \qquad (5.13)$$

as the tractable approximation. The naive mean field updates are a particular set of recursions for finding a stationary point of the resulting optimization problem. In the case of the Ising model, the naive mean field updates are a classical set of recursions from statistical physics, typically justified in a heuristic manner in terms of "self-consistency". For the case of a Gaussian Markov random field, the naive mean field updates turn out to be equivalent to the Gauss–Jacobi or Gauss–Seidel algorithm for solving a linear system of equations.

Example 5.2 (Naive Mean Field for Ising Model). As an illustration, we derive the naive mean field updates for the Ising model, first introduced in Example 3.1. Recall that the Ising model is characterized by the sufficient statistics $(x_s, \ s \in V)$ and $(x_s x_t, \ (s,t) \in E)$. The associated mean parameters are the singleton and pairwise marginal probabilities

$$\mu_s = \mathbb{E}[X_s] = \mathbb{P}[X_s = 1], \ \text{and} \ \mu_{st} = \mathbb{E}[X_s X_t] = \mathbb{P}[X_s = 1, X_t = 1],$$

respectively. The full vector μ of mean parameters is an element of $\mathbb{R}^{|V|+|E|}$.

   Letting F_0 denote the fully disconnected graph — that is, without any edges — the tractable set $\mathcal{M}_{F_0}(G)$ consists of all mean parameter vectors $\mu \in \mathbb{R}^{|V|+|E|}$ that arise from the product distribution (5.13). Explicitly, in this binary case, we have

$$\mathcal{M}_{F_0}(G) := \{\mu \in \mathbb{R}^{|V|+|E|} \mid 0 \leq \mu_s \leq 1 \quad \forall \, s \in V, \quad \text{and}$$
$$\mu_{st} = \mu_s \, \mu_t \quad \forall \, (s,t) \in E.\}, \quad (5.14)$$

where the constraints $\mu_{st} = \mu_s \mu_t$ arise from the product nature of any distribution that is Markov with respect to F_0.

For any $\mu \in \mathcal{M}_{F_0}(G)$, the value of the dual function — that is, the negative entropy of a product distribution — has an explicit form in terms of $(\mu_s,\, s \in V)$. In particular, a straightforward computation shows that this entropy takes the form:

$$-A^*_{F_0}(\mu) = -\sum_{s \in V} \left[\mu_s \log \mu_s + (1 - \mu_s) \log(1 - \mu_s) \right]$$

$$= \sum_{s \in V} H_s(\mu_s). \tag{5.15}$$

With these two ingredients, we can derive the specific form of the mean field optimization problem (5.8) for the product distribution and the Ising model. Given the product structure of the tractable family, any mean parameter $\mu \in \mathcal{M}_{F_0}(G)$ satisfies the equality $\mu_{st} = \mu_s \mu_t$ for all $(s,t) \in E$. Combining the expression for the entropy in (5.15) with the characterization of $\mathcal{M}_{F_0}(G)$ in (5.14) yields the naive mean field problem

$$A(\theta) \geq \max_{(\mu_1,\ldots,\mu_m) \in [0,1]^m} \left\{ \sum_{s \in V} \theta_s \mu_s + \sum_{(s,t) \in E} \theta_{st} \mu_s \mu_t + \sum_{s \in V} H_s(\mu_s) \right\}. \tag{5.16}$$

For any $s \in V$, this objective function is strictly concave as a scalar function of μ_s with all other coordinates held fixed. Moreover, the maximum over μ_s with the other variables $\mu_t, t \neq s$ held fixed is attained in the open interval $(0,1)$. Indeed, by taking the derivative with respect to μ, setting it to zero and performing some algebra, we obtain the update

$$\mu_s \leftarrow \sigma \left(\theta_s + \sum_{t \in N(s)} \theta_{st} \mu_t \right), \tag{5.17}$$

where $\sigma(z) := [1 + \exp(-z)]^{-1}$ is the logistic function. Thus, we have derived — from a variational perspective — the naive mean field updates presented earlier (2.15).

What about convergence properties and the nature of the fixed points? Applying Equation (5.17) iteratively to each node in succession amounts to performing coordinate ascent of the mean field variational

problem (5.16). Since the maximum is uniquely attained for every coordinate update, known results on coordinate ascent methods [21] imply that any sequence $\{\mu^0, \mu^1, \ldots\}$ generated by the updates (5.17) is guaranteed to converge to a local optimum of the naive mean field problem (5.16).

Unfortunately, the mean field problem is *nonconvex* in general, so that there may be multiple local optima, and the limit point of the sequence $\{\mu^0, \mu^1, \ldots\}$ can depend strongly on the initialization μ^0. We discuss this nonconvexity and its consequences at greater depth in Section 5.4. Despite these issues, the naive mean field approximation becomes asymptotically exact for certain types of models as the number of nodes m grows to infinity [12, 267]. An example is the ferromagnetic Ising model defined on the complete graph K_m with suitably rescaled parameters $\theta_{st} > 0$ for all $(s,t) \in E$; see Baxter [12] for further discussion of such exact cases.

Similarly, it is straightforward to apply the naive mean field approximation to other types of graphical models, as we illustrate for a multivariate Gaussian.

Example 5.3 (Gaussian Mean Field). Recall the Gaussian Markov random field, first discussed as an exponential family in Example 3.3. Its mean parameters consist of the vector $\mu = \mathbb{E}[X] \in \mathbb{R}^m$, and the symmetric[1] matrix $\Sigma = \mathbb{E}[XX^T] \in \mathcal{S}_+^m$. Suppose that we again use the completely disconnected graph $F_0 = (V, \emptyset)$ as the tractable class. Any Gaussian distribution that is Markov with respect to F_0 must have a diagonal covariance matrix, meaning that the set of tractable mean parameters takes the form:

$$\mathcal{M}_{F_0}(G) = \left\{ (\mu, \Sigma) \in \mathbb{R}^m \times \mathcal{S}_+^m \mid \Sigma - \mu\mu^T = \mathrm{diag}(\Sigma - \mu\mu^T) \succeq 0 \right\}. \tag{5.18}$$

---

[1] Strictly speaking, only elements $[\Sigma]_{st}$ with $(s,t) \in E$ are included in the mean parameterization, since the associated canonical parameter is zero for any pair $(u,v) \notin E$.

For any such product distribution, the entropy (negative dual function) has the form:

$$-A^*_{F_0}(\mu, \Sigma) = \frac{m}{2}\log 2\pi e + \frac{1}{2}\log\det\left[\Sigma - \mu\mu^T\right]$$

$$= \frac{m}{2}\log 2\pi e + \frac{1}{2}\sum_{s=1}^{m}\log(\Sigma_{ss} - \mu_s^2).$$

Combining this form of the dual function with the constraints (5.18) yields that for a multivariate Gaussian, the value $A(\theta)$ of the cumulant function is lower bounded by

$$\max_{(\mu,\Sigma)} \left\{\langle\theta, \mu\rangle + \frac{1}{2}\langle\!\langle\Theta, \Sigma\rangle\!\rangle + \frac{1}{2}\sum_{s=1}^{m}(\Sigma_{ss} - \mu_s^2) + \frac{m}{2}\log 2\pi e\right\}, \quad (5.19)$$

such that $\Sigma_{ss} - \mu_s^2 > 0$, $\Sigma_{st} = \mu_s\mu_t$,

where $(\theta, \Theta) \in \Omega$ are the canonical parameters associated with the multivariate Gaussian (see Example 3.3). The optimization problem (5.19) yields the naive mean field lower bound for the multivariate Gaussian.

This optimization problem can be further simplified by substituting the constraints $\Sigma_{st} = \mu_s\mu_t$ directly into the term $\langle\!\langle\Theta, \Sigma\rangle\!\rangle = \sum_{s,t}\Theta_{st}\Sigma_{st}$ that appears in the objective function. Doing so and then taking derivatives with respect to the remaining optimization variables (namely, μ_s and μ_{ss}) yields the stationary condition

$$\frac{1}{2(\mu_{ss} - \mu_s^2)} = -\Theta_{ss}, \quad \text{and} \quad \frac{\mu_s}{2(\mu_{ss} - \mu_s^2)} = \theta_s + \sum_{t \in N(s)}\Theta_{st}\mu_t, \quad (5.20)$$

which must be satisfied for every vertex $s \in V$. One particular set of updates for solving these fixed point equations is the iteration

$$\mu_s \leftarrow -\frac{1}{\Theta_{ss}}\left\{\theta_s + \sum_{t \in N(s)}\Theta_{st}\mu_t\right\}. \quad (5.21)$$

Are these updates convergent, and what is the nature of the fixed points? Interestingly, the updates (5.21) are equivalent, depending on the particular ordering used, to either the *Gauss–Jacobi* or the *Gauss–Seidel* methods [67] for solving the normal equations $\mu = -\Theta^{-1}\theta$.

Therefore, if the Gaussian mean field updates (5.21) converge, they compute the correct mean vector. Moreover, the convergence behavior of such updates is well understood: for instance, the updates (5.21) are guaranteed to converge whenever $-\Theta$ is strictly diagonally dominant; see Demmel [67] for further details on such Gauss–Jacobi and Gauss–Seidel iterations for solving matrix-vector equations.

5.4   Nonconvexity of Mean Field

An important fact about the mean field approach is that the variational problem (5.25) may be nonconvex, so that there may be local minima, and the mean field updates can have multiple solutions. The source of this nonconvexity can be understood in different ways, depending on the formulation of the problem. As an illustration, let us return again to naive mean field for the Ising model.

Example 5.4. (Nonconvexity for Naive Mean Field) We now consider an example, drawn from Jaakkola [120], that illustrates the nonconvexity of naive mean field for a simple model. Consider a pair (X_1, X_2) of binary variates, taking values in the space[2] $\{-1,+1\}^2$, thereby defining a 3D exponential family of the form $p_\theta(x) \propto \exp(\theta_1 x_1 + \theta_2 x_2 + \theta_{12} x_1 x_2)$, with associated mean parameters $\mu_i = \mathbb{E}[X_i]$ and $\mu_{12} = \mathbb{E}[X_1 X_2]$. If the constraint $\mu_{12} = \mu_1 \mu_2$ is imposed directly, as in the formulation (5.16), then the naive mean field objective function for this very special model takes the form:

$$f(\mu_1, \mu_2; \theta) = \theta_{12}\mu_1\mu_2 + \theta_1\mu_1 + \theta_2\mu_2 + H(\mu_1) + H(\mu_2), \qquad (5.22)$$

where $H(\mu_i) = -\frac{1}{2}(1 + \mu_i)\log\frac{1}{2}(1 + \mu_i) - \frac{1}{2}(1 - \mu_i)\log\frac{1}{2}(1 - \mu_i)$ are the singleton entropies for the $\{-1,+1\}$-spin representation.

Now, let us consider a subfamily of such models, given by canonical parameters of the form:

$$(\theta_1, \theta_2, \theta_{12}) = \left(0,\, 0,\, \frac{1}{4}\log\frac{q}{1-q}\right) =: \theta(q),$$

---

[2] This model, known as a "spin" representation, is a simple transformation of the $\{0,1\}^2$ state space considered earlier.
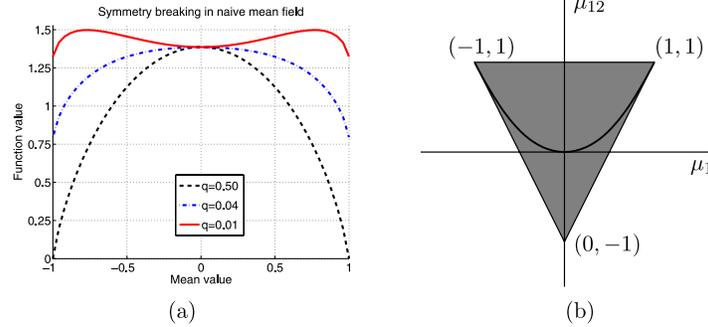
Fig. 5.2 Two different perspectives on the nonconvexity of naive mean field for the Ising model. (a) Illustration of the naive mean field objective function (5.22) for three different parameter values: $q \in \{0.50, 0.04, 0.01\}$. For $q = 0.50$ and $q = 0.04$, the global maximum is achieved at $(\mu_1, \mu_2) = (0, 0)$, whereas for $q = 0.01$, the point $(0, 0)$ is no longer a global maximum. Instead the global maximum is achieved at two nonsymmetric points, $(+\mu, -\mu)$ and $(-\mu, +\mu)$. (b) Nonconvexity can also be seen by examining the shape of the set of fully factorized marginals for a pair of binary variables. The gray area shows the polytope defined by the inequality (5.23), corresponding to the intersection of $\mathcal{M}(G)$ with the hyperplane $\mu_1 = \mu_2$. The nonconvex quadratic set $\mu_{12} = \mu_1^2$ corresponds to the intersection of this projected polytope with the set $\mathcal{M}_{F_0}(G)$ of fully factorized marginals.

where $q \in (0, 1)$ is a parameter. By construction, this model is symmetric in X_1 and X_2, so that for any value of $q \in (0, 1)$, we have $\mathbb{E}[X_1] = \mathbb{E}[X_2] = 0$. Moreover, some calculation shows that $q = \mathbb{P}[X_1 = X_2]$.

For $q = 0.50$, the objective function $f(\mu_1, \mu_2; \theta(0.50))$ achieves its global maximum at $(\mu_1, \mu_2) = (0, 0)$, so that the mean field approximation is exact. (This exactness is to be expected since $\theta(0.50) = (0, 0, 0)$, corresponding to a completely decoupled model.) As q decreases away from 0.50, the objective function f starts to change, until for suitably small q, the point $(\mu_1, \mu_2) = (0, 0)$ is no longer the global maximum — in fact, it is not even a local maximum.

To illustrate this behavior explicitly, we consider the cross-section of f obtained by setting $\mu_1 = \tau$ and $\mu_2 = -\tau$, and then plot the 1D function $f(\tau, -\tau; \theta(q))$ for different values of q. As shown in Figure 5.2(a), for $q = 0.50$, this 1D objective function has a unique global maximum at $\tau = 0$. As q decreases away from 0.50, the objective function gradually flattens out, as shown in the change between $q = 0.50$ and $q = 0.04$. For q sufficiently close to zero, the point $\tau = 0$ is no longer a global

maximum; instead, as shown in the curve for $q = 0.01$, the global maximum is achieved at the two points $\pm\tau^*$ on either side of $\tau = 0$. Thus, for sufficiently small q, the maximum of the objective function (5.22) occurs at a pair $\mu_1^* \neq \mu_2^*$, even though the original model is always symmetric. This phenomenon, known in the physics literature as "spontaneous symmetry-breaking", is a manifestation of nonconvexity, since the optimum of any convex function will always respect symmetries in the underlying problem. Symmetry-breaking is not limited to this toy example, but also occurs with mean field methods applied to larger and more realistic graphical models, for which there may be a large number of competing modes in the objective function.

Alternatively, nonconvexity in naive mean field can be understood in terms of the shape of the constraint set as an inner approximation to $\mathcal{M}$. For a pair of binary variates $\{X_1, X_2\} \in \{-1, 1\}^2$, the set $\mathcal{M}$ is easily characterized: the mean parameters $\mu_i = \mathbb{E}[X_i]$ and $\mu_{12} = \mathbb{E}[X_1 X_2]$ are completely characterized by the four inequalities $1 + ab\mu_{12} + a\mu_1 + b\mu_2 \geq 0$, where $\{a, b\} \in \{-1, 1\}^2$. So as to facilitate visualization, consider a particular projection of this polytope — namely, that corresponding to intersection with the hyperplane $\mu_1 = \mu_2$. In this case, the four inequalities reduce to three simpler ones — namely:

$$\mu_{12} \leq 1, \quad \mu_{12} \geq 2\mu_1 - 1, \quad \mu_{12} \geq -2\mu_1 - 1. \tag{5.23}$$

Figure 5.2(b) shows the resulting 2D polytope, shaded in gray. Now consider the intersection between this projected polytope and the set of factorized marginals $\mathcal{M}_{F_0}(G)$. The factorization condition imposes an additional constraint $\mu_{12} = \mu_1^2$, yielding a quadratic curve lying within the 2D polytope described by the Equations (5.23), as illustrated in Figure 5.2(b). Since this quadratic set is not convex, this establishes that $\mathcal{M}_{F_0}(G)$ is not convex either. Indeed, if it were convex, then its intersection with any hyperplane would also be convex.

The geometric perspective on the set $\mathcal{M}(G)$ and its inner approximation $\mathcal{M}_F(G)$ reveals that more generally, mean field optimization is always nonconvex for any exponential family in which the state space $\mathcal{X}^m$ is finite. Indeed, for any such exponential family, the set $\mathcal{M}(G)$ is
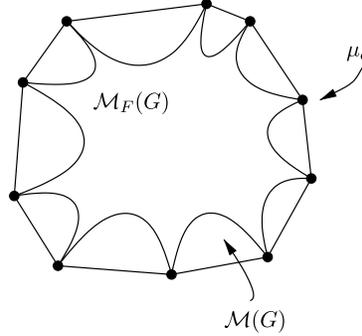
Fig. 5.3 Cartoon illustration of the set $\mathcal{M}_F(G)$ of mean parameters that arise from tractable distributions is a nonconvex inner bound on $\mathcal{M}(G)$. Illustrated here is the case of discrete random variables where $\mathcal{M}(G)$ is a polytope. The circles correspond to mean parameters that arise from delta distributions, and belong to both $\mathcal{M}(G)$ and $\mathcal{M}_F(G)$.

a *finite convex hull*[3]

$$\mathcal{M}(G) = \text{conv}\,\{\phi(e),\, e \in \mathcal{X}^m\} \tag{5.24}$$

in d-dimensional space, with extreme points of the form $\mu_e := \phi(e)$ for some $e \in \mathcal{X}^m$. Figure 5.3 provides a highly idealized illustration of this polytope, and its relation to the mean field inner bound $\mathcal{M}_F(G)$.

We now claim that $\mathcal{M}_F(G)$ — assuming that it is a strict subset of $\mathcal{M}(G)$ — must be a nonconvex set. To establish this claim, we first observe that $\mathcal{M}_F(G)$ contains all of the extreme points $\mu_x = \phi(x)$ of the polytope $\mathcal{M}(G)$. Indeed, the extreme point μ_x is realized by the distribution that places all its mass on x, and such a distribution is Markov with respect to any graph. Therefore, if $\mathcal{M}_F(G)$ were a convex set, then it would have to contain any convex combination of such extreme points. But from the representation (5.24), taking convex combinations of all such extreme points generates the full polytope $\mathcal{M}(G)$. Therefore, whenever $\mathcal{M}_F(G)$ is a proper subset of $\mathcal{M}(G)$, it cannot be a convex set.

Consequently, nonconvexity is an intrinsic property of mean field approximations. As suggested by Example 5.4, this nonconvexity

---

[3] For instance, in the discrete case when the sufficient statistics ϕ are defined by indicator functions in the standard overcomplete basis (3.34), we referred to $\mathcal{M}(G)$ as a marginal polytope.

can have significant operational consequences, including symmetry-breaking, multiple local optima, and sensitivity to initialization. Nonetheless, mean-field methods have been used successfully in a variety of applications, and the lower bounding property of mean field methods is attractive, for instance in the context of parameter estimation, as we discuss at more length in Section 6.

5.5   Structured Mean Field

Of course, the essential principles underlying the mean field approach are not limited to fully factorized distributions. More generally, we can consider classes of tractable distributions that incorporate additional structure. This *structured mean field approach* was first proposed by Saul and Jordan [209], and further developed by various researchers [10, 259, 120].

Here, we capture the structured mean field idea by discussing a general form of the updates for an approximation based on an arbitrary subgraph F of the original graph G. We do not claim that these specific updates are the best for practical purposes; the main goal here is the conceptual one of understanding the structure of the solution. Depending on the particular context, various techniques from nonlinear programming might be suitable for solving the mean field problem (5.8).

Let $\mathcal{I}(F)$ be the subset of indices corresponding to sufficient statistics associated with F, and let $\mu(F) := (\mu_\alpha, \, \alpha \in \mathcal{I}(F))$ be the associated subvector of mean parameters. We use $\mathcal{M}(F)$ to denote the set of realizable mean parameters defined by the subgraph F and by the subset of sufficient statistics picked out by $\mathcal{I}(F)$; it is a subset of $\mathbb{R}^{|\mathcal{I}(F)|}$. It is important to note that $\mathcal{M}(F)$ differs from the previously defined (5.5) set $\mathcal{M}_F(G)$, which is based on the entire set of sufficient statistics ϕ, and so is a subset of $\mathbb{R}^{|\mathcal{I}|}$.

We then observe that the mean field problem (5.8) has the following key properties:

(a) The subvector $\mu(F)$ can be an arbitrary member of $\mathcal{M}(F)$.
(b) The dual function A_F^* actually depends only on $\mu(F)$, and *not* on mean parameters μ_β for indices β in the complement $\mathcal{I}^c(F) := \mathcal{I}(G)\backslash\mathcal{I}(F)$.

Of course, the mean parameters μ_β for indices $\beta \in \mathcal{I}^c(F)$ do play a role in the problem; in particular, they arise within the linear term $\langle \mu, \theta \rangle$. Moreover, each mean parameter μ_β is constrained in a non-linear way by the choice of $\mu(F)$. Accordingly, for each $\beta \in \mathcal{I}^c(F)$, we write $\mu_\beta = g_\beta(\mu(F))$ for some nonlinear function g_β, of which particular examples are given below. Based on these observations, the optimization problem (5.8) can be rewritten in the form:

$$
\max_{\mu \in \mathcal{M}_F(G)} \left\{ \langle \theta, \mu \rangle - A_F^*(\mu) \right\}
$$
$$
= \max_{\mu(F) \in \mathcal{M}(F)} \Bigg\{ \underbrace{\sum_{\alpha \in \mathcal{I}(F)} \theta_\alpha \mu_\alpha + \sum_{\alpha \in \mathcal{I}^c(F)} \theta_\alpha g_\alpha(\mu(F)) - A_F^*(\mu(F))}_{f(\mu(F))} \Bigg\}. \tag{5.25}
$$

On the left-hand side, the optimization takes place over the vector $\mu \in \mathcal{M}_F(G)$, which is of the same dimension as $\theta \in \Omega \subseteq \mathbb{R}^{|\mathcal{I}|}$. The objective function f for the optimization on the right-hand side, in contrast, is a function only of the lower-dimensional vector $\mu(F) \in \mathcal{M}(F) \subseteq \mathbb{R}^{|\mathcal{I}(F)|}$.

To illustrate this transformation, consider the case of naive mean field for the Ising model, where $F \equiv F_0$ is the completely disconnected graph. In this case, each edge $(s,t) \in E$ corresponds to an index in the set $\mathcal{I}^c(F_0)$; moreover, for any such edge, we have $g_{st}(\mu(F_0)) = \mu_s \mu_t$. Since F_0 is the completely disconnected graph, $\mathcal{M}(F_0)$ is simply the hypercube $[0,1]^m$. Therefore, for this particular example, the right-hand side of Equation (5.25) is equivalent to Equation (5.16).

Taking the partial derivatives of the objective function f with respect to some μ_β with $\beta \in \mathcal{I}(F)$ yields:

$$
\frac{\partial f}{\partial \mu_\beta}(\mu(F)) = \theta_\beta + \sum_{\alpha \in \mathcal{I}(G) \setminus \mathcal{I}(F)} \theta_\alpha \frac{\partial g_\alpha}{\partial \mu_\beta}(\mu(F)) - \frac{\partial A_F^*}{\partial \mu_\beta}(\mu(F)). \tag{5.26}
$$

Setting these partial derivatives to zero and rearranging yields the fixed point condition

$$
\nabla A_F^*(\mu(F)) = \theta + \sum_{\alpha \in \mathcal{I}(G) \setminus \mathcal{I}(F)} \theta_\alpha \nabla g_\alpha(\mu(F)).
$$

To obtain a more intuitive form of this fixed point equation, recall from Proposition 3.1 that the gradient ∇A defines the forward mapping, from canonical parameters θ to mean parameters μ. Similarly, as we have shown in Section 3.6, the gradient ∇A^* defines the backward mapping from mean parameters to canonical parameters. Let $\gamma(F)$ denote the canonical parameters that are dually coupled with $\mu(F)$, we can rewrite the fixed point update for component β as

$$\gamma_\beta(F) \leftarrow \theta_\beta + \sum_{\alpha \in \mathcal{I}(G) \backslash \mathcal{I}(F)} \theta_\alpha \frac{\partial g_\alpha}{\partial \mu_\beta}(\mu(F)). \qquad (5.27)$$

After any such update, it is then necessary to adjust all mean parameters $\mu_\delta(F)$ that depend on $\gamma_\beta(F)$ — for instance, via junction tree updates — so that global consistency is maintained for the tractable approximation.

Alternatively, letting A_F be the conjugate dual of the function A_F^* previously defined, we have $\nabla A_F(\gamma(F)) = \mu(F)$. This fact follows from the Legendre duality between (A_F, A_F^*) (see Appendix B.3 for background), and the usual cumulant properties summarized in Proposition 3.1. By exploiting this relation, we obtain an alternative form of the update (5.27), one which involves only the mean parameters $\mu(F)$:

$$\mu_\beta(F) \leftarrow \frac{\partial A_F}{\partial \gamma_\beta}\left(\theta + \sum_{\alpha \in \mathcal{I}(G) \backslash \mathcal{I}(F)} \theta_\alpha \nabla g_\alpha(\mu(F))\right). \qquad (5.28)$$

We illustrate the updates (5.27) and (5.28) with some examples.

Example 5.5 (Re-derivation of Naive MF Updates). We first check that Equation (5.28) reduces to the naive mean field updates (5.17), when $F = F_0$ is the completely disconnected graph. For any product distribution on the Ising model, the subset of relevant mean parameters is given by $\mu(F_0) = (\mu_1, \ldots, \mu_m)$. By the properties of the product distribution, we have $g_{st}(\mu(F_0)) = \mu_s \mu_t$ for all edges (s, t), so that

$$\frac{\partial g_{st}}{\partial \mu_\alpha} = \begin{cases} \mu_t & \text{if } \alpha = s \\ \mu_s & \text{if } \alpha = t \\ 0 & \text{otherwise.} \end{cases}$$

Thus, for a given node index $s \in V$, the right-hand side of (5.27) is given by the sum

$$\theta_s + \sum_{t \in N(s)} \theta_{st} \mu_t. \tag{5.29}$$

Next, we observe that for a fully disconnected graph, the function $A^*_{F_0}$ corresponds to a sum of negative entropies — that is, $A^*_{F_0}(\mu(F_0)) = -\sum_{s \in V} H(\mu_s)$ — and so is a separable function. Consequently, the cumulant function A_{F_0} is also separable, and of the form $A_{F_0}(\gamma(F_0)) = \sum_{s \in V} \log(1 + \exp(\gamma_s))$. The partial derivatives are given by

$$\frac{\partial A_{F_0}}{\partial \gamma_\beta}(\gamma(F_0)) = \frac{\exp(\gamma_\beta)}{1 + \exp(\gamma_\beta)}, \tag{5.30}$$

which is the logistic function. Combining the expression (5.29) with the logistic function form (5.30) of the partial derivative shows that Equation (5.28), when specialized to product distributions and the Ising model, reduces to the naive mean field update (5.17).

Example 5.6 (Structured MF for Factorial HMMs). To provide a more interesting example of the updates (5.27), consider a factorial hidden Markov model, as described in Ghahramani and Jordan [92]. Figure 5.4(a) shows the original model, which consists of a set of M Markov chains ($M = 3$ in this diagram), which share a common observation at each time step (shaded nodes). Although the separate chains are independent *a priori*, the common observation induces an effective coupling among the observations. (Note that the observation nodes are linked by the moralization process that converts the directed graph into an undirected representation.) Thus, an equivalent model is shown in panel (b), where the dotted ellipses represent the induced coupling of each observation. A natural choice of approximating distribution in this case is based on the subgraph F consisting of the decoupled set of M chains, as illustrated in panel (c).

Now consider the nature of the quantities $g_\beta(\mu(F))$, which arise in the cost function (5.25). In this case, any function g_β will be defined
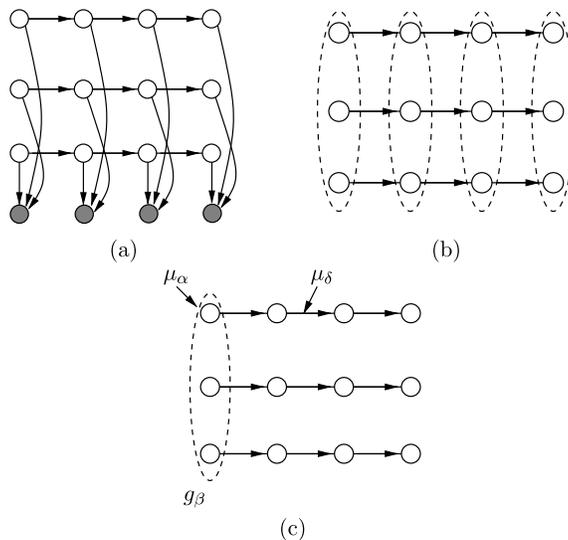
Fig. 5.4 Structured mean field approximation for a factorial HMM. (a) Original model consists of a set of hidden Markov models (defined on chains), coupled at each time by a common observation. (b) An equivalent model, where the ellipses represent interactions among all nodes at a fixed time, induced by the common observation. (c) Approximating distribution formed by a product of chain-structured models. Here μ_α and μ_δ are the sets of mean parameters associated with the indicated vertex and edge, respectively.

on some subset of M nodes that are coupled at a given time slice (e.g., see ellipse in panel (c)). Note that this subset of nodes is independent with respect to the approximating distribution. Therefore, the function $g_\beta(\mu(F))$ will decouple into a product of terms of the form $f_i(\{\mu_i(F)\})$, where each f_i is some function of the mean parameters $\{\mu_i\} \equiv \{\mu_i(F)\}$ associated with node $i = 1, \ldots, M$ in the relevant cluster. For instance, if the factorial HMM involved binary variables and $M = 3$ and $\beta = (stu)$, then $g_{stu}(\mu) = \mu_s \mu_t \mu_u$.

The decoupled nature of the approximation yields valuable savings on the computational side. In particular, the junction tree updates necessary to maintain consistency of the approximation can be performed by applying the forward–backward algorithm (i.e., the sum-product updates as an exact method) to each chain separately. This decoupling also has important consequences for the structure of any mean field fixed point. In particular, it can be seen that no term $g_\beta(\mu(F))$ will

ever depend on mean parameters associated with edges within any of the chains (e.g., μ_δ in panel (c)). Otherwise stated, the partial derivative $\frac{\partial g_\beta}{\partial \mu_\delta}$ is equal to 0 for all $\beta \in \mathcal{I}(G) \backslash \mathcal{I}(F)$. As an immediate consequence of these derivatives vanishing, the mean field canonical parameter $\gamma_\delta(F)$ remains equal to θ_δ for all iterations of the updates (5.27). Any intermediate junction tree steps to maintain consistency will not affect $\gamma_\delta(F)$ either. We conclude that it is, in fact, optimal to simply copy the edge potentials θ_δ from the original distribution onto each of the edges in the structured mean field approximation. In this particular form of structured mean field, only the single node potentials will be altered from their original setting. This conclusion is sensible, since the structured approximation (c) is a factorized approximation on a set of M chains, the internal structure of which is fully preserved in the approximation.

In addition to structured mean field, there are various other extensions to naive mean field, which we mention only in passing here. Jaakkola and Jordan [121] explored the use of mixture distributions in improving the mean field approximation. A large class of techniques, including linear response theory and the TAP method [e.g., 127, 155, 182, 195, 255, 267], seek to improve the mean field approximation by introducing higher-order correction terms. Typically, the lower bound on the log partition function is not preserved by these higher-order methods. Leisinck and Kappen [156] proposed a class of higher-order expansions that generate tighter lower bounds.

6

Variational Methods in Parameter Estimation

Our focus in the previous sections has been on the problems of computing or approximating the cumulant function value $A(\theta)$ and the mean parameters $\mu = \mathbb{E}_\theta[\phi(X)]$, assuming that the canonical parameter θ specifying the density p_θ was known. In this section, we turn to the inverse problem of estimating θ on the basis of observed data. We consider both the frequentist setting, in which the parameter θ is assumed to be fixed but unknown (Sections 6.1 and 6.2), and the Bayesian setting, in which the parameter θ is viewed as a random variable (Section 6.3).

6.1   Estimation in Fully Observed Models

The simplest case of parameter estimation corresponds to the case of *fully observed data*: a collection $X_1^n := \{X^1, \ldots, X^n\}$ of n independent and identically distributed (i.i.d.) m-vectors, each sampled according to p_θ. Suppose that our goal to estimate the unknown parameter θ, which we view as a deterministic but nonrandom quantity for the moment. A classical approach to this estimation problem, dating back to Fisher, is via the principle of *maximum likelihood*: estimate θ by maximizing

the log likelihood of the data, given by $\ell(\theta; X_1^n) := \frac{1}{n} \sum_{i=1}^n \log p_\theta(X^i)$. It is convenient and has no effect on the maximization to rescale the log likelihood by $1/n$, as we have done here.

For exponential families, the rescaled log likelihood takes a particularly simple and intuitive form. In particular, we define *empirical mean parameters* $\widehat{\mu} := \widehat{\mathbb{E}}[\phi(X)] = \frac{1}{n} \sum_{i=1}^n \phi(X^i)$ associated with the sample X_1^n. In terms of these quantities, the log likelihood for an exponential family can be written as

$$\ell(\theta; X_1^n) = \langle \theta, \widehat{\mu} \rangle - A(\theta). \tag{6.1}$$

This expression highlights the close connection to maximum entropy and conjugate duality, as discussed in Section 3.6. The maximum likelihood estimate is the vector $\widehat{\theta} \in \mathbb{R}^d$ maximizing this (random) objective function (6.1). As a corollary of Theorem 3.4, whenever $\widehat{\mu} \in \mathcal{M}^\circ$, there exists a unique maximum likelihood solution. Indeed, by taking derivatives with respect to θ of the objective function (6.1), the maximum likelihood estimate (MLE) is specified by the *moment matching conditions* $\mathbb{E}_{\widehat{\theta}}[\phi(X)] = \widehat{\mu}$. Moreover, by standard results on asymptotics of M-estimators [30, 233], the MLE $\widehat{\theta}$ in regular exponential families is consistent, in that it converges in probability to θ as the sample size n tends to infinity.

6.1.1 Maximum Likelihood for Triangulated Graphs

Our discussion up to this point has ignored the computational issue of solving the moment matching equation so as to obtain the MLE $\widehat{\theta}$. As with other statistical computations in exponential families, the difficulty of solving this problem turns out to depend strongly on the graph structure. For triangulated[1] graphs, the MLE can be written as a closed-form function of the empirical marginals $\widehat{\mu}$, which we describe here.

To illustrate the basic idea with a minimum of notational overhead, let us consider the simplest instance of a triangulated graph: namely, a tree $T = (V, E)$, with discrete random variables X_s taking values in

---

[1] See Section 2.5.2 for further discussion of triangulated graphs.

$\mathcal{X}_s := \{0, 1, \ldots, r_s - 1\}$. As previously discussed in Section 4, this collection of distributions can be represented as an exponential family (4.1), using indicator functions $\mathbb{I}_{s;j}[x_s]$ for the event $\{X_s = j\}$ and $\mathbb{I}_{st;jk}[x_s, x_t]$ for the event $\{X_s = j, X_t = k\}$. Moreover, the mean parameters in this exponential family are marginal probabilities — explicitly,

$$\mu_{s;j} = \mathbb{P}_\theta[X_s = j] \quad \text{for all } s \in V,\ j \in \mathcal{X}_s, \text{ and}$$

$$\mu_{st;jk} = \mathbb{P}_\theta[X_s = j, X_t = k] \quad \text{for all } (s,t) \in E \text{ and } (j,k) \in \mathcal{X}_s \times \mathcal{X}_t.$$

Given an i.i.d. sample $X_1^n := \{X^1, \ldots, X^n\}$, the empirical mean parameters correspond to the singleton and pairwise marginal probabilities

$$\widehat{\mu}_{s;j} = \frac{1}{n} \sum_{i=1}^n \mathbb{I}_{s;j}[X_s^i] \quad \text{and} \quad \widehat{\mu}_{st;jk} = \frac{1}{n} \sum_{i=1}^n \mathbb{I}_{st;jk}[X_s^i, X_t^i], \qquad (6.2)$$

induced by the data.

With this set-up, we can exhibit the closed-form expression for the MLE in terms of the empirical marginals. For this particular exponential family, our assumption that $\widehat{\mu} \in \mathcal{M}^\circ$ means that the empirical marginals are all *strictly* positive. Consequently, the vector $\widehat{\theta}$ with elements

$$\widehat{\theta}_{s;j} := \log \widehat{\mu}_{s;j} \quad \forall\, s \in V, \quad j \in \mathcal{X}_s \quad \text{and} \qquad (6.3a)$$

$$\widehat{\theta}_{st;jk} := \log \frac{\widehat{\mu}_{st;jk}}{\widehat{\mu}_{s;j} \widehat{\mu}_{t;k}} \quad \forall\, (s,t) \in E,\ (j,k) \in \mathcal{X}_s \times \mathcal{X}_t \qquad (6.3b)$$

is well defined.

We claim that the vector $\widehat{\theta}$ is the maximum likelihood estimate for the problem. In order to establish this claim, it suffices to show that the moment matching conditions $\mathbb{E}_{\widehat{\theta}}[\phi(X)] = \widehat{\mu}$ hold for the distribution $p_{\widehat{\theta}}$. From the definition (6.3), this exponential family member has the form

$$p_{\widehat{\theta}}(x) = \exp \left\{ \sum_{s \in V} \widehat{\theta}_s(x_s) + \sum_{(s,t) \in E} \widehat{\theta}_{st}(x_s, x_t) \right\}$$

$$= \prod_{s \in V} \widehat{\mu}(x_s) \prod_{(s,t) \in E} \frac{\widehat{\mu}_{st}(x_s, x_t)}{\widehat{\mu}_s(x_s) \widehat{\mu}_t(x_t)}, \qquad (6.4)$$

where we have used the fact (verifiable by some calculation) that $A(\widehat{\theta}) = 0$ in this parameterization. Moreover, the distribution $p_{\widehat{\theta}}$ has

as its marginal distributions the empirical quantities $\widehat{\mu}_s$ and $\widehat{\mu}_{st}$. This claim follows as a consequence of the junction tree framework (see Proposition 2.1); alternatively, they can be proved directly by an inductive "leaf-stripping" argument, based on a directed version of the factorization (6.4). Consequently, the MLE has an explicit closed-form solution in terms of the empirical marginals for a tree. The same closed-form property holds more generally for any triangulated graph.

6.1.2   Iterative Methods for Computing MLEs

For nontriangulated graphical models, there is no longer a closed-form solution to the MLE problem, and iterative algorithms are needed. As an illustration, here we describe the iterative proportional fitting (IPF) algorithm [61, 59], a type of coordinate ascent method with particularly intuitive updates. To describe it, let us consider a generalization of the sufficient statistics $\{\mathbb{I}_{s;j}(x_s)\}$ and $\{\mathbb{I}_{st;jk}(x_s, x_t)\}$ used for trees, in which we define similar indicator functions over cliques of higher order. More specifically, for any clique C of size k in a given graph G, let us define a set of clique indicator functions

$$\mathbb{I}_J(x_C) = \prod_{s=1}^{k} \mathbb{I}_{s;j_s}[x_s],$$

one for each configuration $J = (j_1, \ldots, j_k)$ over x_C. Each such sufficient statistic is associated with a canonical parameter $\theta_{C;J}$. As before, the associated mean parameters $\mu_{C;J} = \mathbb{E}[\mathbb{I}_{C;J}(X_C)] = \mathbb{P}_\theta[X_C = J]$ are simply marginal probabilities, but now defined over cliques. Similarly, the data defines a set $\widehat{\mu}_{C;J} = \widehat{\mathbb{P}}[X_C = J]$ of empirical marginal probabilities.

Since the MLE problem is differentiable and jointly concave in the vector θ, coordinate ascent algorithms are guaranteed to converge to the global optimum. Using the cumulant generating properties of A (see Proposition 3.1), we can take derivatives with respect to some coordinate $\theta_{C;J}$, thereby obtaining

$$\frac{\partial \ell}{\partial \theta_{C;J}} = \widehat{\mu}_{C;J} - \frac{\partial A}{\partial \theta_{C;J}}(\theta) = \widehat{\mu}_{C;J} - \mu_{C;J}, \qquad (6.5)$$

Iterative Proportional Fitting: At iterations $t = 0, 1, 2, \ldots$

    (i) Choose a clique $C = C(t)$ and compute the local marginal distribution $\mu_{C;J}^{(t)} := \mathbb{P}_{\theta^{(t)}}[X_C = J] \quad \forall \, J \in \mathcal{X}^{|C|}$.

    (ii) Update the canonical parameter vector

$$
\theta_\alpha^{(t+1)} = \begin{cases} \theta_{C;J}^{(t)} + \log \dfrac{\widehat{\mu}_{C;J}}{\mu_{C;J}^{(t)}} & \text{if } \alpha = (C;J) \text{ for some } J \in \mathcal{X}^{|C|}. \\ \theta_\alpha^{(t)} & \text{otherwise.} \end{cases}
$$

Fig. 6.1 Steps in the iterative proportional fitting (IPF) procedure.

where $\mu_{C;J} = \mathbb{E}_\theta[\mathbb{I}_{C;J}(X_C)]$ are the mean parameters defined by the current model. As a consequence, zero gradient points are specified by matching the model marginals $\mathbb{P}_\theta[X_C = J]$ to the empirical marginals $\widehat{\mathbb{P}}[X_C = J]$. With this set-up, we may define a sequence of parameter estimates $\{\theta^{(t)}\}$ via the recursion in Figure 6.1.

These updates have two key properties: first, the log partition function is never changed by the updates. Indeed, we have

$$
\begin{aligned}
A(\theta^{(t+1)}) &= \log \sum_x \exp\left\{ \langle \theta^{(t)}, \phi(x) \rangle \right\} \exp\left\{ \sum_{J \in \mathcal{X}^{|C|}} \log \frac{\widehat{\mu}_{C;J}}{\mu_{C;J}^{(t)}} \mathbb{I}_{C;J}[x_C] \right\} \\
&= \log \mathbb{E}_{\theta^{(t)}} \left[\prod_J \left(\frac{\widehat{\mu}_{C;J}}{\mu_{C;J}^{(t)}} \right)^{\mathbb{I}_{C;J}[x_C]} \right] + A(\theta^{(t)}) \\
&= \log \sum_{J \in \mathcal{X}^{|C|}} \widehat{\mu}_{C;J} + A(\theta^{(t)}) \\
&= A(\theta^{(t)}).
\end{aligned}
$$

Note that this invariance in A arises due to the overcompleteness of this particular exponential family — in particular, since $\sum_J \mathbb{I}_{C;J}(x_C) = 1$ for all cliques C. Second, the update in step (ii) enforces exactly the zero-gradient condition (6.5). Indeed, after the parameter update $\theta^{(t)} \to \theta^{(t+1)}$, we have

$$
\mathbb{E}_{\theta^{(t+1)}}[\mathbb{I}_{C;J}(X_C)] = \frac{\widehat{\mu}_{C;J}}{\mu_{C;J}^{(t)}} \, \mathbb{E}_{\theta^{(t)}}[\mathbb{I}_{C;J}(X_C)] = \widehat{\mu}_{C;J}.
$$

Consequently, this IPF algorithm corresponds to as a *blockwise coordinate ascent* method for maximizing the objective function (6.1). At each iteration, the algorithm picks a particular block of coordinates — that is, $\theta_{C;J}$, for all configurations $J \in \mathcal{X}^{|C|}$ defined on the block — and maximizes the objective function over this subset of coordinates. Consequently, standard theory on blockwise coordinate ascent methods [21] can be used to establish convergence of the IPF procedure. More generally, this iterative proportional fitting procedure is a special case of a broader class of iterative scaling, or successive projection algorithms; see papers [59, 60, 61] or the book [45] for further details on such algorithms and their properties.

From a computational perspective, however, there are still some open issues associated with the IPF algorithm in application to graphical models. Note that step (i) in the updates assume a "black box" routine that, given the current canonical parameter vector $\theta^{(t)}$, returns the associated vector of mean parameters $\mu^{(t)}$. For graphs of low treewidth, this computation can be carried out with the junction tree algorithm. For more general graphs not amenable to the junction tree method, one could imagine using approximate sampling methods or variational methods. The use of such approximate methods and their impact on parameter estimation is still an active area of research [225, 231, 241, 243, 253].

6.2 Partially Observed Models and Expectation–Maximization

A more challenging version of parameter estimation arises in the *partially observed* setting, in which the random vector $X \sim p_\theta$ is not observed directly, but indirectly via a "noisy" version Y of X. This formulation includes the special case in which some subset where the remaining variates are unobserved — that is, "latent" or "hidden". The expectation–maximization (EM) algorithm of Dempster et al. [68] provides a general approach to computing MLEs in this partially observed setting. Although the EM algorithm is often presented as an alternation between an expectation step (E step) and a maximization step (M step), it is also possible to take a variational perspective on EM,

and view both steps as maximization steps [60, 179]. Such a perspective illustrates how variational inference algorithms can be used in place of exact inference algorithms in the E step within the EM framework, and in particular, how the mean field approach is especially appropriate for this task.

A brief outline of our presentation in this section is as follows. We begin by deriving the EM algorithm in the exponential family setting, showing how the E step reduces to the computation of expected sufficient statistics — i.e., mean parameters. As we have seen, the variational framework provides a general class of methods for computing approximations of mean parameters. This observation suggests a general class of *variational EM algorithms*, in which the approximation provided by a variational inference algorithm is substituted for the mean parameters in the E step. In general, as a consequence of making such a substitution, one loses the guarantees of exactness that are associated with the EM algorithm. In the specific case of mean field algorithms, however, one is still guaranteed that the method performs coordinate ascent on a *lower bound* of the likelihood function.

6.2.1   Exact EM Algorithm in Exponential Families

We begin by deriving the exact EM algorithm for exponential families. Suppose that the set of random variables is partitioned into a vector Y of *observed* variables, and a vector X of *unobserved* variables, and the probability model is a joint exponential family distribution for (X, Y):

$$p_\theta(x, y) = \exp\big\{ \langle \theta, \phi(x, y) \rangle - A(\theta) \big\}. \qquad (6.6)$$

Given an observation $Y = y$, we can also form the conditional distribution

$$p_\theta(x \mid y) = \frac{\exp\{\langle \theta, \phi(x, y) \rangle\}}{\int_{\mathcal{X}^m} \exp\{\langle \theta, \phi(x, y) \rangle\} \nu(dx)}$$

$$:= \exp\{\langle \theta, \phi(x, y) \rangle - A_y(\theta)\}, \qquad (6.7)$$

where for each *fixed* y, the log partition function A_y associated with this conditional distribution is given by the integral

$$A_y(\theta) := \log \int_{\mathcal{X}^m} \exp\{\langle \theta, \phi(x, y) \rangle\} \nu(dx). \qquad (6.8)$$

Thus, we see that the conditional distribution $p_\theta(x \mid y)$ belongs to an d-dimensional exponential family, with vector $\phi(\cdot, y)$ of sufficient statistics.

The maximum likelihood estimate $\widehat{\theta}$ is obtained by maximizing log probability of the observed data y, which is referred to as the *incomplete log likelihood* in the setting of EM. This incomplete log likelihood is given by the integral

$$\ell(\theta; y) := \log \int_{\mathcal{X}} \exp\{\langle \theta, \phi(x, y) \rangle \,-\, A(\theta)\} \nu(dx) \,=\, A_y(\theta) - A(\theta) \quad (6.9)$$

where the second equality follows by definition (6.8) of A_y.

We now describe how to obtain a variational lower bound on the incomplete log likelihood. For each fixed y, the set $\mathcal{M}_y$ of valid mean parameters in this exponential family takes the form

$$\mathcal{M}_y = \left\{ \mu \in \mathbb{R}^d \mid \mu = \mathbb{E}_p[\phi(X, y)] \quad \text{for some } p \right\}, \quad (6.10)$$

where p is any density function over X, taken with respect to underlying base measure ν. Consequently, applying Theorem 3.4 to this exponential family provides the variational representation

$$A_y(\theta) = \sup_{\mu_y \in \mathcal{M}_y} \left\{ \langle \theta, \mu_y \rangle - A_y^*(\mu_y) \right\}, \quad (6.11)$$

where the conjugate dual is also defined variationally as

$$A_y^*(\mu_y) := \sup_{\theta \in \mathrm{dom}(A_y)} \left\{ \langle \mu_y, \theta \rangle - A_y(\theta) \right\}. \quad (6.12)$$

From the variational representation (6.11), we obtain the lower bound $A_y(\theta) \geq \langle \mu_y, \theta \rangle - A_y^*(\mu_y)$, valid for any $\mu_y \in \mathcal{M}_y$. Using the representation (6.9), we obtain the lower bound for the incomplete log likelihood:

$$\ell(\theta; y) \geq \langle \mu_y, \theta \rangle - A_y^*(\mu_y) - A(\theta) =: \mathcal{L}(\mu_y, \theta). \quad (6.13)$$

With this set-up, the EM algorithm is coordinate ascent on this function $\mathcal{L}$ defining the lower bound (6.13):

$$(\textbf{E step}) \qquad \mu_y^{(t+1)} = \arg \max_{\mu_y \in \mathcal{M}_y} \mathcal{L}(\mu_y, \theta^{(t)}) \qquad (6.14\text{a})$$

$$(\textbf{M step}) \qquad \theta^{(t+1)} = \arg \max_{\theta \in \Omega} \mathcal{L}(\mu_y^{(t+1)}, \theta). \qquad (6.14\text{b})$$

To see the correspondence with the traditional presentation of the EM algorithm, note first that the maximization underlying the E step reduces to

$$\max_{\mu_y \in \mathcal{M}_y} \{\langle \mu_y, \theta^{(t)} \rangle - A_y^*(\mu_y)\}, \tag{6.15}$$

which by the variational representation (6.11) is equal to $A_y(\theta^{(t)})$, with the maximizing argument equal to the mean parameter that is dually coupled with $\theta^{(t)}$. Otherwise stated, the vector $\mu_y^{(t+1)}$ that is computed by maximization in the first argument of $\mathcal{L}(\mu_y, \theta)$ is exactly the expectation $\mu_y{}^{(t+1)} = \mathbb{E}_{\theta^{(t)}}[\phi(X, y)]$, a computation that is traditionally referred to as the E step, for obvious reasons. Moreover, the maximization underlying the M step reduces to

$$\max_{\theta \in \Omega} \{\langle \mu_y{}^{(t+1)}, \theta \rangle - A(\theta)\}, \tag{6.16}$$

which is simply a maximum likelihood problem based on the expected sufficient statistics $\mu_y{}^{(t+1)}$ — traditionally referred to as the M step.

Moreover, given that the value achieved by the E step on the right-hand side of (6.15) is equal to $A_y(\theta^{(t)})$, the inequality in (6.13) becomes an equality by (6.9). Thus, after the E step, the lower bound $\mathcal{L}(\mu_y, \theta^{(t)})$ is actually equal to the incomplete log likelihood $\ell(\theta^{(t)}; y)$, and the subsequent maximization of $\mathcal{L}$ with respect to θ in the M step is guaranteed to increase the log likelihood as well.

Example 6.1 (EM for Gaussian Mixtures). We illustrate the EM algorithm with the classical example of the Gaussian mixture model, discussed earlier in Example 3.4. Let us focus on a scalar mixture model for simplicity, for which the observed vector Y is Gaussian mixture vector with r components, where the unobserved vector $X \in \{0, 1, \ldots, r-1\}$ indexes the components. The complete likelihood of this mixture model can be written as

$$p_\theta(x, y) = \exp \left\{ \sum_{j=0}^{r-1} \mathbb{I}_j(x) \big[\alpha_j + \gamma_j y + \widetilde{\gamma}_j y^2 - A_j(\gamma_j, \widetilde{\gamma}_j) \big] - A(\alpha) \right\},$$
$$\tag{6.17}$$

where $\theta = (\alpha, \gamma, \widetilde{\gamma})$, with the vector $\alpha \in \mathbb{R}^r$ parameterizing the multinomial distribution over the hidden vector X, and the pair $(\gamma_j, \widetilde{\gamma}_j) \in \mathbb{R}^2$ parameterizing the Gaussian distribution of the jth mixture component. The log normalization function $A_j(\gamma_j, \widetilde{\gamma}_j)$ is for the (conditionally) Gaussian distribution of Y given $X = j$, whereas the function $A(\alpha) = \log[\sum_{j=0}^{r-1} \exp(\alpha_j)]$ normalizes the multinomial distribution. When the complete likelihood is viewed as an exponential family, the sufficient statistics are the collection of triplets

$$\Psi_j(x, y) := \{\mathbb{I}_j(x), \ \mathbb{I}_j(x)y, \ \mathbb{I}_j(x)y^2\} \quad \text{for } j = 0, \ldots, r - 1.$$

Consider a collection of i.i.d. observations $y^1, \ldots, y^n$. To each observation y^i, we associate a triplet $(\mu^i, \eta^i, \widetilde{\eta}^i) \in \mathbb{R}^r \times \mathbb{R}^r \times \mathbb{R}^r$, corresponding to expectations of the triplet of sufficient statistics $\Psi_j(X, y^i)$, $j = 0, \ldots, r - 1$. Since the conditional distribution has the form

$$p(x \mid y^i, \theta) \propto \exp\left\{ \sum_{j=0}^{r-1} \mathbb{I}_j(x) \left[\alpha_j + \gamma_j y^i + \widetilde{\gamma}_j (y^i)^2 - A_j(\gamma_j, \widetilde{\gamma}_j) \right] \right\},$$

some straightforward computation yields that the probability of the jth mixture component — or equivalently, the mean parameter $\mu_j^i = \mathbb{E}[\mathbb{I}_j(X)]$ — is given by

$$
\begin{aligned}
\mu_j^i &= \mathbb{E}[\mathbb{I}_j(X)] \\
&= \frac{\exp\left\{ \alpha_j + \gamma_j y^i + \widetilde{\gamma}_j (y^i)^2 - A_j(\gamma_j, \widetilde{\gamma}_j) \right\}}{\sum_{k=0}^{r-1} \exp\left\{ \alpha_k + \gamma_k y^i + \widetilde{\gamma}_k (y^i)^2 - A_k(\gamma_k, \widetilde{\gamma}_k) \right\}}.
\end{aligned}
\tag{6.18}
$$

Similarly, the remaining mean parameters are

$$\eta_j^i = \mathbb{E}[\mathbb{I}_j(X)y^i] = \mu_j^i y^i, \quad \text{and} \tag{6.19a}$$

$$\widetilde{\eta}_j^i = \mathbb{E}[\mathbb{I}_j(X)y^i] = \mu_j^i (y^i)^2. \tag{6.19b}$$

The computation of these expectations (6.18) and (6.19) correspond to the E step for this particular model.

The M step requires solving the MLE optimization problem (6.16) over the triplet $\theta = (\alpha, \gamma, \widetilde{\gamma})$, using the expected sufficient statistics computed in the E step. Some computation shows that this maximization

takes the form:

$$\arg\max_{(\alpha,\gamma,\widetilde{\gamma})\in\Omega}\left\{\left\langle\alpha,\sum_{i=1}^{n}\mu^i\right\rangle+\left\langle\gamma,\sum_{i=1}^{n}\mu^i y^i\right\rangle\right.$$

$$\left.+\left\langle\widetilde{\gamma},\sum_{i=1}^{n}\mu^i(y^i)^2\right\rangle-\sum_{j=0}^{r-1}\sum_{i=1}^{n}\mu_j^i A_j(\gamma_j,\widetilde{\gamma}_j)-nA(\alpha)\right\}.$$

Consequently, the optimization decouples into separate maximization problems: one for the vector α parameterizing the mixture component distribution, and one for each of the pairs $(\gamma_j,\widetilde{\gamma}_j)$ specifying the Gaussian mixture components. By the moment-matching property of maximum likelihood estimates, the optimum solution is to update α such that

$$\mathbb{E}_\alpha[\mathbb{I}_j(X)]=\frac{1}{n}\sum_{i=1}^{n}\mu_j^i\quad\text{for each }j=0,\dots,r-1,$$

and to update the canonical parameters $(\gamma_j,\widetilde{\gamma}_j)$ specifying the jth Gaussian mixture component $(Y\mid X=j)$ such that the associated mean parameters — corresponding to the Gaussian mean and second moment — are matched to the data as follows:

$$\mathbb{E}_{\gamma_j,\widetilde{\gamma}_j}[Y\mid X=j]=\frac{\sum_{i=1}^{n}\mu_j^i y_j^i}{\sum_{i=1}^{n}\mu_j^i},\quad\text{and}\tag{6.20}$$

$$\mathbb{E}_{\gamma_j,\widetilde{\gamma}_j}[Y^2\mid X=j]=\frac{\sum_{i=1}^{n}\mu_j^i(y_j^i)^2}{\sum_{i=1}^{n}\mu_j^i}.\tag{6.21}$$

In these equations, the expectations are taken under the exponential family distribution specified by $(\gamma_j,\widetilde{\gamma}_j)$.

___

6.2.2   Variational EM

What if it is infeasible to compute the expected sufficient statistics? One possible response to this problem is to make use of a mean field approximation for the E step. In particular, given some class of tractable subgraphs F, recall the set $\mathcal{M}_F(G)$, as defined in Equation (5.5), corresponding to those mean parameters that can be obtained by

distributions that factor according to F. By using $\mathcal{M}_F(G)$ as an inner approximation to the set $\mathcal{M}_y$, we can compute an approximate version of the E step (6.15), of the form

$$(\textbf{Mean field E step}) \qquad \max_{\mu \in \mathcal{M}_F(G)} \left\{ \langle \mu, \theta^{(t)} \rangle - A_y^*(\mu) \right\}. \qquad (6.22)$$

This variational E step thus involves replacing the exact mean parameter $\mathbb{E}_{\theta^{(t)}}[\phi(X, y)]$, under the current model $\theta^{(t)}$, with the approximate set of mean parameters computed by a mean field algorithm. Despite this (possibly crude) approximation, the resulting variational EM algorithm is a still coordinate ascent algorithm for $\mathcal{L}$. However, given that the E step no longer closes the gap between the auxiliary function $\mathcal{L}$ and the incomplete log likelihood, it is no longer the case that the algorithm necessarily goes uphill in the latter quantity. Nonetheless, the variational mean field EM algorithm still has the attractive interpretation of maximizing a *lower bound* on the incomplete log likelihood.

In Section 4, we described a broad class of variational methods, including belief propagation and expectation-propagation, that also generate approximations to mean parameters. It is tempting, and common in practice, to substitute the approximate mean parameters obtained from these relaxations in the place of the expected sufficient statistics in defining a "variational EM algorithm." Such a substitution is particularly tempting given that these methods can yield better approximations to mean parameters than the mean field approach. Care must be taken in working with these algorithms, however, because the underlying relaxations do not (in general) guarantee lower bounds on the log partition function.[2] In the absence of guaranteed lower bounds, the connection to EM is thus less clear than in the mean field case; in particular, the algorithm is not guaranteed to maximize a lower bound on the incomplete log likelihood.

6.3 Variational Bayes

All of the variational approximations that we have discussed thus far in the survey, as well as those to be discussed in later sections,

---

[2] However, see Sudderth et al. [224] for results on certain classes of attractive graphical models for which the Bethe approximation does provide such a lower bound.

are applicable in principle to Bayesian inference problems, where the parameters are endowed with probability distributions and viewed as random variables. In the literature on the topic, however, the term "variational Bayes" has been reserved thus far for the application of the mean-field variational method to Bayesian inference [16, 91]. In this section, we review this application and use the terminology of variational Bayes to refer to the application, but it should be borne in mind that this is but one potential application of variational methodology to Bayesian inference.

We consider a general set-up akin to that used in the preceding development of the EM algorithm. In particular, let the data be partitioned into an observed component Y and an unobserved component X, and assume that the complete data likelihood lies in some exponential family

$$p(x, y \mid \theta) = \exp\left\{\langle \eta(\theta), \phi(x, y)\rangle - A(\eta(\theta))\right\}. \qquad (6.23)$$

The function $\eta : \mathbb{R}^d \to \mathbb{R}^d$ provides some additional flexibility in the parameterization of the exponential family; this is convenient in the Bayesian setting.[3] We assume moreover that the prior distribution over Θ also lies in some exponential family, of the *conjugate prior form*:

$$p_{\xi, \lambda}(\theta) = \exp\left\{\langle \xi, \eta(\theta)\rangle - \lambda A(\eta(\theta)) - B(\xi, \lambda)\right\}. \qquad (6.24)$$

Note that this exponential family is specified by the sufficient statistics $\{\eta(\theta), -A(\eta(\theta))\} \in \mathbb{R}^d \times \mathbb{R}$, with associated canonical parameters $(\xi, \lambda) \in \mathbb{R}^d \times \mathbb{R}$. Its cumulant function B is defined in the usual way

$$B(\xi, \lambda) := \log \int \exp\left\{\langle \xi, \eta(\theta)\rangle - \lambda A(\eta(\theta))\right\} d\theta,$$

and expectations of the sufficient statistics $\eta(\theta)$ and $-A(\eta(\theta))$ can be obtained in the usual way by taking derivatives of $B(\xi, \lambda)$ with respect to ξ and λ.

---

[3] Up to this point, we have considered only exponential families in canonical form, for which $[\eta(\theta)]_i = \theta_i$ for all $i = 1, \ldots, d$. This is without loss of generality, because any regular, minimal exponential family can be reparameterized in such a canonical form [43]. However, it can be convenient to express exponential families with nonidentity η.

The class of models specified by the pair of Equations (6.23) and (6.24) is broad. It includes as special cases Gaussian mixture models with Dirichlet priors, and the latent Dirichlet allocation model from Example 3.5.

We now consider a central problem in Bayesian inference, that of computing the *marginal likelihood* $p_{\xi^*,\lambda^*}(y)$, where y is an observed data point and where (ξ^*,λ^*) are fixed values of the hyperparameters. This computation entails averaging over *both* the unobserved variates X and the random parameters Θ. Working in the log domain, we have.

$$\log p_{\xi^*,\lambda^*}(y) = \log \int \left[\int p(x, y \mid \theta)\, dx \right] p_{\xi^*,\lambda^*}(\theta)\, d\theta$$

$$= \log \int p_{\xi^*,\lambda^*}(\theta)\, p(y \mid \theta)\, d\theta. \qquad (6.25)$$

We now multiply and divide by $p_{\xi,\lambda}(\theta)$ and use Jensen's inequality (see Proposition 5.1), to obtain the following lower bound, valid for any choice of (ξ,λ) in the domain of B:

$$\log p_{\xi^*,\lambda^*}(y) \geq \mathbb{E}_{\xi,\lambda}[\log p(y \mid \Theta)] + \mathbb{E}_{\xi,\lambda}\left[\log \frac{p_{\xi^*,\lambda^*}(\Theta)}{p_{\xi,\lambda}(\Theta)}\right], \qquad (6.26)$$

with equality for $(\xi,\lambda) = (\xi^*,\lambda^*)$. Here $\mathbb{E}_{\xi,\lambda}$ denotes averaging over Θ with respect to the distribution $p_{\xi,\lambda}(\theta)$.

From Equation (6.9), we have $\log p(y \mid \Theta) = A_y(\eta(\Theta)) - A(\eta(\Theta))$, which when substituted into Equation (6.26) yields

$$\log p_{\xi^*,\lambda^*}(y) \geq \mathbb{E}_{\xi,\lambda}\left[A_y(\eta(\Theta)) - A(\eta(\Theta))\right] + \mathbb{E}_{\xi,\lambda}\left[\log \frac{p_{\xi^*,\lambda^*}(\Theta)}{p_{\xi,\lambda}(\Theta)}\right],$$

where A_y is the cumulant function of the conditional density $p(x \mid y, \theta)$.

For each fixed y, recall the set $\mathcal{M}_y$ of mean parameters of the form $\mu = \mathbb{E}[\phi(X, y)]$. For any realization of Θ, we could in principle apply the mean field lower bound (6.13) on the log likelihood $A_y(\Theta)$, using a value $\mu(\Theta) \in \mathcal{M}_y$ that varies with Θ. We would thereby obtain that the marginal log likelihood $\log p_{\xi^*,\lambda^*}(y)$ is lower bounded by

$$\mathbb{E}_{\xi,\lambda}\left[\langle \mu(\Theta), \eta(\Theta)\rangle - A_y^*(\mu(\Theta)) - A(\eta(\Theta))\right] + \mathbb{E}_{\xi,\lambda}\left[\log \frac{p_{\xi^*,\lambda^*}(\Theta)}{p_{\xi,\lambda}(\Theta)}\right].$$

$$(6.27)$$

At this stage, even after this sequence of lower bounds, if we were to optimize over the set of joint distributions on (X, Θ) — or equivalently, over the choice $\mu(\Theta)$ and the hyperparameters (ξ, λ) specifying the distribution of Θ — the optimized lower bound (6.27) *would be equal* to the original marginal log likelihood $\log p_{\xi^*, \lambda^*}(y)$.

The variational Bayes algorithm is based on optimizing this lower bound using only product distributions over the pair (X, Θ). Such optimization is often described as "free-form", in that beyond the assumption of a product distribution, the factors composing this product distribution are allowed to be arbitrary. But as we have seen in Section 3.1, the maximum entropy principle underlying the variational framework yields solutions that necessarily have exponential form. Therefore, we can exploit conjugate duality to obtain compact forms for these solutions, working with dually coupled sets of parameters for the distributions over both X and Θ.

We now derive the variational Bayes algorithm as a coordinate-ascent method for solving the mean field variational problem over product distributions. We begin by reformulating the objective function into a form that allows us to exploit conjugate duality. Since μ is independent of Θ, the optimization problem (6.27) can be simplified to

$$\left[\langle \mu, \bar{\eta} \rangle - A_y^*(\mu) - \overline{A} \right] + \mathbb{E}_{\xi, \lambda} \left[\log \frac{p_{\xi^*, \lambda^*}(\Theta)}{p_{\xi, \lambda}(\Theta)} \right], \qquad (6.28)$$

where $\bar{\eta} := \mathbb{E}_{\xi, \lambda}[\eta(\Theta)]$ and $\overline{A} := \mathbb{E}_{\xi, \lambda}[A(\Theta)]$. Using the exponential form (6.24) of $p_{\xi, \lambda}$, we have

$$\mathbb{E}_{\xi, \lambda} \left[\log \frac{p_{\xi^*, \lambda^*}(\Theta)}{p_{\xi, \lambda}(\Theta)} \right]$$
$$= \langle \bar{\eta}, \xi^* - \xi \rangle + \langle -\overline{A}, \lambda^* - \lambda \rangle - B(\xi^*, \lambda^*) + B(\xi, \lambda). \quad (6.29)$$

By the conjugate duality between (B, B^*), and since $(\bar{\eta}, \overline{A})$ are dually coupled with (ξ, λ) by construction, we have

$$B^*(\bar{\eta}, \overline{A}) = \langle \bar{\eta}, \xi \rangle + \langle -\overline{A}, \lambda \rangle - B(\xi, \lambda). \qquad (6.30)$$

By combining Equations (6.28) through (6.30) and performing some algebra, we find that the decoupled optimization problem is equivalent

to maximizing the function

$$\langle \mu + \xi^*, \overline{\eta} \rangle - A_y^*(\mu) + \langle \lambda^* + 1, -\overline{A} \rangle - B^*(\overline{\eta}, \overline{A}) \tag{6.31}$$

over $\mu \in \mathcal{M}_y$ and $(\overline{\eta}, \overline{A}) \in \mathrm{dom}(B)$. Performing coordinate ascent on this function amounts to first maximizing first over μ, and then maximizing over the mean parameters[4] $(\overline{\eta}, \overline{A})$. Doing so generates a sequence of iterates $(\mu^{(t)}, \overline{\eta}^{(t)}, \overline{A}^{(t)})$, which are updated as follows:

$$\mu^{(t+1)} = \arg \max_{\mu \in \mathcal{M}_y} \left\{ \langle \mu, \overline{\eta}^{(t)} \rangle - A_y^*(\mu) \right\}, \quad \text{and} \tag{6.32}$$

$$(\overline{\eta}^{(t+1)}, \overline{A}^{(t+1)})$$
$$= \arg \max_{(\overline{\eta}, \overline{A})} \left\{ \langle \mu^{(t+1)} + \xi^*, \overline{\eta} \rangle - (1 + \lambda^*) \overline{A} - B^*(\overline{\eta}, \overline{A}) \right\}. \tag{6.33}$$

In fact, both of these coordinate-wise optimizations have explicit solutions. The explicit form of Equation (6.32) is given by

$$\mu^{(t+1)} = \mathbb{E}_{\overline{\eta}^{(t)}}[\phi(X, y)], \tag{6.34}$$

as can be verified by taking derivatives in Equation (6.32), and using the cumulant properties of A_y^* (see Proposition 3.1). By analogy to the EM algorithm, this step is referred to as the "VB-E step." Similarly, in the "VB-M step" we first update the hyperparameters (ξ, λ) as

$$(\xi^{(t+1)}, \lambda^{(t+1)}) = (\xi^* + \mu^{(t+1)}, \lambda^* + 1),$$

and then use these updated hyperparameters to compute the new mean parameters $\overline{\eta}$:

$$\overline{\eta}^{(t+1)} = \mathbb{E}_{(\xi^{(t+1)}, \lambda^{(t+1)})}[\eta(\Theta)]. \tag{6.35}$$

In summary, the variational Bayes algorithm is an iterative algorithm that alternates between two steps. The VB-E step (6.34) entails computing the conditional expectation of the sufficient statistics given the data

$$(\textbf{VB-E step}) \qquad \mu^{(t+1)} = \int \phi(x, y) p(x \mid y, \overline{\eta}^{(t)}) dx,$$

---

[4] Equivalently, by the one-to-one correspondence between exponential and mean parameters (Theorem 3.3), this step corresponds to maximization over the associated canonical parameters (ξ, λ).

with the conditional distribution specified by the current parameter estimate $\bar{\eta}^{(t)}$. The VB-M step (6.35) entails updating the hyperparameters to incorporate the data, and then recomputing the averaged parameter

$$(\textbf{VB-M step}) \qquad \bar{\eta}^{(t+1)} = \int \eta(\theta) \, p_{\xi^{(t+1)},\lambda^{(t+1)}}(\theta) \, d\theta.$$

The ordinary variational EM algorithm can be viewed as a "degenerate" form of these updates, in which the prior distribution over Θ is always a delta function at a single point estimate of θ.

7

Convex Relaxations and Upper Bounds

Up to this point, we have considered two broad classes of approximate variational methods: Bethe approximations and their extensions (Section 4), and mean field methods (Section 5). Mean field methods provide not only approximate mean parameters but also lower bounds on the log partition function. In contrast, the Bethe method and its extensions lead to neither upper or lower bounds on the log partition function. We have seen that both classes of methods are, at least in general, based on *nonconvex* variational problems. For mean field methods, the nonconvexity stems from the nature of the inner approximation to the set of mean parameters, as illustrated in Figure 5.3. For the Bethe-type approaches, the lack of convexity in the objective function — in particular, the entropy approximation — is the source. Regardless of the underlying cause, such nonconvexity can have undesirable consequences, including multiple optima and sensitivity to the problem parameters (for the variational problem itself), and convergence issues and dependence on initialization (for iterative algorithms used to solve the variational problem).

Given that the underlying exact variational principle (3.45) is certainly convex, it is natural to consider variational approximations that

retain this convexity. In this section, we describe a broad class of such convex variational approximations, based on approximating the set $\mathcal{M}$ with a *convex set*, and replacing the dual function A^* with a *convex function.* These two steps lead to a new variational principle, which represents a computationally tractable alternative to the original problem. Since this approximate variational principle is based on maximizing a concave and continuous function over a convex and compact set, the global optimum is unique, and is achieved. If in addition, the negative dual function $-A^*$, corresponding to the entropy function, is replaced by an upper bound, and the set $\mathcal{M}$ is approximated by a convex outer bound, then the global optimum of the approximate variational principle also provides an upper bound on the log partition function. These upper bounds are complementary to the lower bounds provided by mean field procedures (see Section 5).

We begin by describing a general class of methods based on approximating the entropy by a convex combination of easily computable entropies, thereby obtaining a tractable relaxation that is guaranteed to be convex. As our first example illustrates, following this procedure using tree-structured distributions as the tractable class leads to the family of "convexified" Bethe variational problems, and associated reweighted sum-product algorithms [243, 246]. However, this basic procedure for "convexification" is quite broadly applicable; as we describe, it yields convex analogs of other known variational methods, including Kikuchi and region-graph methods [246, 251, 260], as well as expectation-propagation approximation. It has also suggested novel variational methods, also based on the notion of convex combinations, including those based on planar graph decomposition [94]. Finally, there are other convex variational relaxations that are not directly based on convex combinations of tractable distributions, including the method of conditional entropy decompositions [95], and methods based on semidefinite constraints and log-determinant programming [248]. Apart from these known algorithms, there are a large number of novel methods based on convex variational relaxations that await discovery.

We conclude the section by discussing some benefits of convexity in variational methods. In addition to the obvious one of unique optima,

possible benefits include guarantees of algorithmic stability and useful-
ness as surrogate likelihoods in parameter estimation.

7.1 Generic Convex Combinations and Convex Surrogates

We begin with a generic description of approximations based
on convex combinations of tractable distributions. Consider a d-
dimensional exponential family, defined by a vector of sufficient statis-
tics $\phi = (\phi_\alpha, \ \alpha \in \mathcal{I})$, and an associated vector of canonical parameters
$\theta = (\theta_\alpha, \ \alpha \in \mathcal{I})$. Although computing mean parameters for an arbitrary
$\theta \in \Omega$ might be intractable, it is frequently the case that such compu-
tations are tractable for certain special choices of the canonical param-
eters. If the exponential family is viewed as a Markov random field
defined by some underlying graph G, many such special choices can
be indexed by particular subgraphs F, say belonging to some class $\mathfrak{D}$.
Examples that we consider below include $\mathfrak{D}$ corresponding to the set of
all spanning trees of G, or planar subgraphs of G. In terms of the expo-
nential family, we can think of the subgraph F as extracting a subset
of indices $\mathcal{I}(F)$ from the full index set $\mathcal{I}$ of potential functions, thereby
defining a lower-dimensional exponential family based on $d(F) < d$ suf-
ficient statistics. We let $\mathcal{M}(F)$ denote the lower-dimensional set of mean
parameters associated with this exponential family; concretely, this set
has the form:

$$\mathcal{M}(F) := \left\{ \mu \in \mathbb{R}^{|\mathcal{I}(F)|} \mid \exists p \ \text{ s.t. } \ \mu_\alpha = \mathbb{E}_p[\phi(X)] \ \forall \alpha \in \mathcal{I}(F) \right\}. \quad (7.1)$$

For any mean parameter $\mu \in \mathcal{M}$, let $\mu \mapsto \mu(F)$ represent the coor-
dinate projection mapping from the full space $\mathcal{I}$ to the subset $\mathcal{I}(F)$
of indices associated with F. By definition of the sets $\mathcal{M} \subseteq \mathbb{R}^d$ and
$\mathcal{M}(F) \subseteq \mathbb{R}^{d(F)}$, we are guaranteed that the projected vector $\mu(F)$ is an
element of $\mathcal{M}(F)$. Let $-A^*(\mu(F))$ be the negative dual function value
(or entropy) associated with the projected mean parameter $\mu(F)$. A
simple consequence of Theorem 3.4 is that this entropy is always an
upper bound on the negative dual function $A^*(\mu)$ defined in the origi-
nal d-dimensional exponential family. With a slight abuse of notation,
we use $H(\mu)$ to denote the negative dual function or entropy $-A^*(\mu)$,
and similarly $H(\mu(F))$ to denote the entropy $-A^*(\mu(F))$.

Proposition 7.1 (Maximum Entropy Bounds). Given any mean parameter $\mu \in \mathcal{M}$ and its projection $\mu(F)$ onto any subgraph F, we have the bound

$$A^*(\mu(F)) \leq A^*(\mu), \qquad (7.2)$$

or alternatively stated, $H(\mu(F)) \geq H(\mu)$.

From Theorem 3.4, the negative dual value $-A^*(\mu)$ corresponds to the entropy of the exponential family member with mean parameters μ, with a similar interpretation for $A^*(\mu(F))$. Alternatively phrased, Proposition 7.1 states the entropy of the exponential family member $p_{\mu(F)}$ is always larger than the entropy of the distribution p_μ. Intuitively, the distribution p_μ is obtained from a maximum entropy problem with more constraints — corresponding to the indices $\alpha \in \mathcal{I} \backslash \mathcal{I}(F)$ — and so has lower entropy. Our proof makes this intuition precise.

Proof. From Theorem 3.4, the dual function is realized as the solution of the optimization problem

$$A^*(\mu) = \sup_{\theta \in \mathbb{R}^d} \left\{ \langle \mu, \theta \rangle - A(\theta) \right\}. \qquad (7.3)$$

Since the exponential family defined by F involves only a subset $\mathcal{I}(F)$ of the potential functions, the associated dual function $A^*(\mu(F))$ is realized by the lower-dimensional optimization problem

$$A^*(\mu(F)) = \sup_{\theta(F) \in \mathbb{R}^{d(F)}} \left\{ \langle \mu(F), \theta(F) \rangle - A(\theta(F)) \right\}.$$

But this optimization problem can be recast as a restricted version of the original problem (7.3):

$$A^*(\mu(F)) = \sup_{\substack{\theta \in \mathbb{R}^d, \\ \theta_\alpha = 0 \ \forall \alpha \notin \mathcal{I}(F)}} \left\{ \langle \mu, \theta \rangle - A(\theta) \right\},$$

from which the claim follows. $\square$

Given the bound (7.2) for each subgraph F, any convex combination of subgraph-structured entropies is also an upper bound on the original entropy. In particular, if we consider a probability distribution over the set of tractable graphs, meaning a probability vector $\rho \in \mathbb{R}^{|\mathfrak{D}|}$ such that

$$\rho(F) \geq 0 \text{ for all tractable } F \in \mathfrak{D}, \text{ and } \quad \sum_F \rho(F) = 1. \qquad (7.4)$$

Any such distribution generates the upper bound

$$H(\mu) \leq \mathbb{E}_\rho[H(\mu(F))] \; := \; \sum_{F \in \mathfrak{D}} \rho(F) H(\mu(F)). \qquad (7.5)$$

Having derived an upper bound on the entropy, it remains to obtain a convex outer bound on the set of mean parameters $\mathcal{M}$. An important restriction is that each of the subgraph-structured entropies $H(\mu(F)) = -A^*(\mu(F))$ must be well-defined on this outer bounding set. The simplest outer approximation with this property is defined by requiring that each projected mean parameter $\mu(F)$ belong to the projected set $\mathcal{M}(F)$. In particular, let us define the constraint set

$$\mathcal{L}(G; \mathfrak{D}) := \big\{ \tau \in \mathbb{R}^d \mid \tau(F) \in \mathcal{M}(F) \quad \forall F \in \mathfrak{D} \big\}. \qquad (7.6)$$

Our shift in notation, from μ to τ, is deliberate, since $\mathcal{L}(G; \mathfrak{D})$ is an outer bound on $\mathcal{M}(G)$. (Indeed, by definition of the sets $\mathcal{M}(F)$, any mean parameter $\mu \in \mathcal{M}(G)$, when projected down to $\mu(F)$, is also globally realizable, so that $\mu \in \mathcal{L}(G; \mathfrak{D})$.) Moreover, $\mathcal{L}(G; \mathfrak{D})$ is a convex set, since each of the subsets $\mathcal{M}(F)$ in its definition are convex.

Overall, these two ingredients — the convex upper bound (7.5) and the convex outer bound (7.6) — yield the following approximate variational principle

$$B_{\mathfrak{D}}(\theta; \rho) := \sup_{\tau \in \mathcal{L}(G; \mathfrak{D})} \left\{ \langle \tau, \theta \rangle + \sum_{F \in \mathfrak{D}} \rho(F) H(\tau(F)) \right\}. \qquad (7.7)$$

Note that the objective function defining $B_{\mathfrak{D}}$ is concave; moreover, the constraint set $\cap_F \mathcal{M}_F$ is convex, since it is the intersection of the sets $\mathcal{M}_F$, each of which are individually convex sets. Overall, then, the function $B_{\mathfrak{D}}$ is a new convex function that approximates the original cumulant function A. We refer to such a function as a *convex surrogate* to A.

Of course, in order for variational principles of the form (7.7) to be useful, it must be possible to evaluate and optimize the objective function. Accordingly, in the following sections, we provide some specific examples of the tractable class $\mathfrak{D}$ that lead to interesting and practical algorithms.

7.2   Variational Methods from Convex Relaxations

In this section, we describe a number of versions of the convex surrogate (7.7), including convex versions of the Bethe/Kikuchi problems, as well as convex versions of expectation-propagation and other moment-matching methods.

7.2.1   Tree-reweighted Sum-Product and Bethe

We begin by deriving the tree-reweighted sum-product algorithm and the tree-reweighted Bethe variational principle [243, 246], corresponding to a "convexified" analog of the ordinary Bethe variational principle described in Section 4. Given an undirected graph $G = (V, E)$, consider a pairwise Markov random field

$$p_\theta(x) \propto \exp\left\{ \sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t) \right\}, \qquad (7.8)$$

where we are using the standard overcomplete representation based on indicator functions at single nodes and edges (see Equation (3.34)). Let the tractable class $\mathfrak{D}$ be the set $\mathfrak{T}$ of all spanning trees $T = (V, E(T))$ of the graph G. (A spanning tree of a graph is a tree-structured subgraph whose vertex set covers the original graph.) Letting ρ denote a probability distribution over the set of spanning trees, Equation (7.5) yields the upper bound $H(\mu) \leq \sum_T \rho(T) H(\mu(T))$, based on a convex combination of tree-structured entropies.

An attractive property of tree-structured entropies is that they decompose additively in terms of entropies associated with the vertices and edges of the tree. More specifically, these entropies are defined by the mean parameters, which (in the canonical overcomplete representation (3.34)) correspond to singleton marginal distributions $\mu_s(\cdot)$

defined at each vertex $s \in V$, and a joint pairwise marginal distribution $\mu_{st}(\cdot, \cdot)$ defined for each edge $(s,t) \in E(T)$. As discussed earlier in Section 4, the factorization (4.8) of any tree-structured probability distribution yields the entropy decomposition

$$H(\mu(T)) = \sum_{s \in V} H_s(\mu_s) - \sum_{(s,t) \in E(T)} I_{st}(\mu_{st}). \qquad (7.9)$$

Now consider the averaged form of the bound (7.5). Since the trees are all spanning, the entropy term H_s for node $s \in V$ receives a weight of one in this average. On the other hand, the mutual information term I_{st} for edge (s,t) receives the weight $\rho_{st} = \mathbb{E}_\rho\big[\mathbb{I}[(s,t) \in E(T)]\big]$, where $\mathbb{I}[(s,t) \in E(T)]$ is an indicator function for the event that edge (s,t) is included in the edge set $E(T)$ of a given tree T. Overall, we obtain the following upper bound on the exact entropy:

$$H(\mu) \leq \sum_{s \in V} H_s(\mu_s) - \sum_{(s,t) \in E} \rho_{st} I_{st}(\mu_{st}). \qquad (7.10)$$

We refer to the edge weight ρ_{st} as the *edge appearance probability*, since it reflects the probability mass associated with edge (s,t). The vector $\rho = (\rho_{st}, (s,t) \in E)$ of edge appearance probabilities belong to a set called the *spanning tree polytope*, as discussed at more length in Theorem 7.2 to follow.

Let us now consider the form of the outer bound $\mathcal{L}(G; \mathfrak{T})$ on the set $\mathcal{M}$. For the pairwise MRF with the overcomplete parameterization under consideration, the set $\mathcal{M}$ is simply the marginal polytope $\mathbb{M}(G)$. On the other hand, the set $\mathbb{M}(T)$ is simply the marginal polytope for the tree T, which from our earlier development (see Proposition 4.1) is equivalent to $\mathbb{L}(T)$. Consequently, the constraint $\mu(T) \in \mathbb{M}(T)$ is equivalent to enforcing nonnegativity constraints, normalization (at each vertex) and marginalization (across each edge) of the tree. Enforcing the inclusion $\mu(T) \in \mathbb{M}(T)$ for *all* trees $T \in \mathfrak{T}$ is equivalent to enforcing the marginalization on *every* edge of the full graph G. We conclude that in this particular case, the set $\mathcal{L}(G; \mathfrak{T})$ is equivalent to the set $\mathbb{L}(G)$ of locally consistent pseudomarginals, as defined earlier (4.7).

Overall, then, we obtain a variational problem that can be viewed as a "convexified" form of the Bethe variational problem. We summarize our findings in the following result [243, 246]:

Theorem 7.2 (Tree-Reweighted Bethe and Sum-Product).

(a) For any choice of edge appearance vector $(\rho_{st}, (s,t) \in E)$ in the spanning tree polytope, the cumulant function $A(\theta)$ evaluated at θ is upper bounded by the solution of the tree-reweighted Bethe variational problem (BVP):

$$B_{\mathfrak{T}}(\theta; \rho_e) := \max_{\tau \in \mathbb{L}(G)} \left\{ \langle \tau, \theta \rangle + \sum_{s \in V} H_s(\tau_s) - \sum_{(s,t) \in E} \rho_{st} I_{st}(\tau_{st}) \right\}.$$
$$(7.11)$$

For any edge appearance vector such that $\rho_{st} > 0$ for all edges (s,t), this problem is strictly convex with a unique optimum.

(b) The tree-reweighted BVP can be solved using the tree-reweighted sum-product updates

$$M_{ts}(x_s) \leftarrow \kappa \sum_{x_t' \in \mathcal{X}_t} \varphi_{st}(x_s, x_t') \frac{\prod_{v \in N(t) \setminus s} \left[M_{vt}(x_t') \right]^{\rho_{vt}}}{\left[M_{st}(x_t') \right]^{(1 - \rho_{ts})}}, \quad (7.12)$$

where $\varphi_{st}(x_s, x_t') := \exp\left(\frac{1}{\rho_{st}} \theta_{st}(x_s, x_t') + \theta_t(x_t') \right)$. The updates (7.12) have a unique fixed point under the assumptions of part (a).

We make a few comments on Theorem 7.2, before providing the proof.

Valid edge weights: Observe that the tree-reweighted Bethe variational problem (7.11) is closely related to the ordinary Bethe problem (4.16). In particular, if we set $\rho_{st} = 1$ for all edges $(s,t) \in E$, then the two formulations are equivalent. However, the condition $\rho_{st} = 1$ implies that every edge appears in every spanning tree of the graph with probability one, which can happen if and only if the graph is actually tree-structured. More generally, the set of valid edge appearance vectors ρ_e
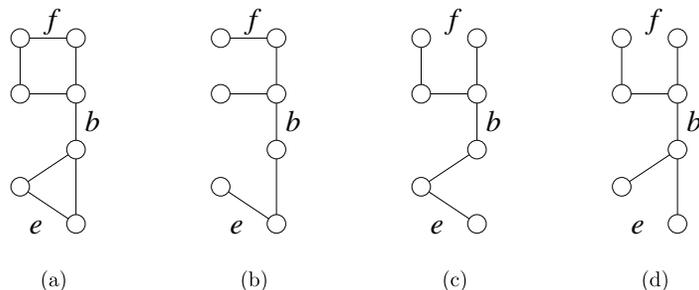
Fig. 7.1 Illustration of valid edge appearance probabilities. Original graph is shown in panel (a). Probability $1/3$ is assigned to each of the three spanning trees $\{T_i \mid i = 1, 2, 3\}$ shown in panels (b)–(d). Edge b appears in all three trees so that $\rho_b = 1$. Edges e and f appear in two and one of the spanning trees, respectively, which gives rise to edge appearance probabilities $\rho_e = 2/3$ and $\rho_f = 1/3$.

must belong to the so-called *spanning tree polytope* [54, 73] associated with G. Note that these edge appearance probabilities must satisfy various constraints, depending on the structure of the graph. A simple example should help to provide intuition.

Example 7.1 (Edge Appearance Probabilities). Figure 7.1(a) shows a graph, and panels (b) through (d) show three of its spanning trees $\{T^1, T^2, T^3\}$. Suppose that we form a uniform distribution ρ over these trees by assigning probability $\rho(T^i) = 1/3$ to each T^i, $i = 1, 2, 3$. Consider the edge with label f; notice that it appears in T^1, but in neither of T^2 and T^3. Therefore, under the uniform distribution ρ, the associated edge appearance probability is $\rho_f = 1/3$. Since edge e appears in two of the three spanning trees, similar reasoning establishes that $\rho_e = 2/3$. Finally, observe that edge b appears in any spanning tree (i.e., it is a bridge), so that it must have edge appearance probability $\rho_b = 1$.

In their work on fractional belief propagation, Wiegerinck and Heskes [261] examined the class of reweighted Bethe problems of the form (7.11), but without the requirement that the weights ρ_{st} belong to the spanning tree polytope. Although loosening this requirement does yield a richer family of variational problems, in general one loses

the guarantee of convexity, and (hence) that of a unique global optimum. On the other hand, Weiss et al. [251] have pointed out that other choices of weights ρ_{st}, not necessarily in the spanning tree polytope, can also lead to convex variational problems. In general, convexity and the upper bounding property are not equivalent. For instance, for any single cycle graph, setting $\rho_{st} = 1$ for all edges (i.e., the ordinary BVP choice) yields a convex variational problem [251], but the value of the Bethe variational problem does not upper bound the cumulant function value. Various other researchers [110, 167, 188, 189] also discuss the choice of edge/clique weights in Bethe/Kikuchi approximations, and its consequences for convexity.

Properties of tree-reweighted sum-product: In analogy to the ordinary Bethe problem and sum-product algorithm, the fixed point of tree-reweighted sum-product (TRW) message-passing (7.12) specifies the optimal solution of the variational problem (7.11) as follows:

$$\tau_s^*(x_s) = \kappa \, \exp\left\{\theta_s(x_s)\right\} \prod_{v \in N(s)} \left[M_{vs}^*(x_s)\right]^{\rho_{vs}} \qquad (7.13a)$$

$$\tau_{st}^*(x_s, x_t) = \kappa \, \varphi_{st}(x_s, x_t) \, \frac{\displaystyle\prod_{v \in N(s)\backslash t} \left[M_{vs}^*(x_s)\right]^{\rho_{vs}}}{\left[M_{ts}^*(x_s)\right]^{(1-\rho_{st})}} \, \frac{\displaystyle\prod_{v \in N(t)\backslash s} \left[M_{vt}^*(x_t)\right]^{\rho_{vt}}}{\left[M_{st}^*(x_t)\right]^{(1-\rho_{ts})}},$$
$$(7.13b)$$

where $\varphi_{st}(x_s, x_t) := \exp\{\frac{1}{\rho_{st}}\theta_{st}(x_s, x_t) + \theta_s(x_s) + \theta_t(x_t)\}$. In contrast to the ordinary sum-product algorithm, the fixed point (and associated optimum (τ_s^*, τ_{st}^*)) is unique for any valid vector of edge appearances. Roosta et al. [204] provide sufficient conditions for convergence, based on contraction arguments such as those used for ordinary sum-product [90, 118, 178, 230]. In practical terms, the updates (7.12) appear to always converge if damped forms of the updates are used (i.e., setting $\log M^{\text{new}} = (1 - \lambda) \log M^{\text{old}} + \lambda \log M$, where M^{old} is the previous vector of messages, and $\lambda \in (0, 1]$ is the damping parameter). As an alternative, Globerson and Jaakkola [96] proposed a related message-passing algorithm based on oriented trees that is guaranteed to converge, but appears to do so more slowly than damped TRW-message passing. Another possibility would be to adapt other double-loop algorithms [110, 111, 254, 270], originally developed for

the ordinary Bethe/Kikuchi problems, to solve these convex minimization problems; see Hazan and Shashua [108] for some recent work along these lines.

Optimal choice of edge appearance probabilities: Note that Equation (7.11) actually describes a family of variational problems, one for each choice of probability distribution over the set of spanning trees $\mathfrak{T}$. A natural question thus arises: what is the "optimal" choice of this probability distribution, or equivalently the choice of the edge appearance vector $(\rho_{st},\ (s,t) \in E)$? One notion of optimality is to choose the edge appearance probabilities that leads to the tightest upper bound on the cumulant function (i.e., for which the gap $B_{\mathfrak{T}}(\theta; \rho) - A(\theta)$ is smallest). In principle, it appears as if the computation of this optimum might be difficult, since it seems to entail searching over all probability distributions over the set $\mathfrak{T}$ of all spanning trees of the graph. The number $|\mathfrak{T}|$ of spanning trees can be computed via the matrix-tree theorem [222], and for sufficiently dense graphs, it grows rapidly enough to preclude direct optimization over $\mathfrak{T}$. However, Wainwright et al. [246] show that it is possible to directly optimize the vector $(\rho_{st},\ (s,t) \in E)$ of edge appearance probabilities, subject to the constraint of membership in the spanning tree polytope. Although this polytope has a prohibitively large number of inequalities, it is possible to maximize a linear function over it — equivalently, to solve a maximum weight spanning tree problem — by a greedy algorithm [212]. Using this fact, it is possible to develop a version of the conditional gradient algorithm [21] for efficiently computing the optimal choice of edge appearance probabilities (see the paper [246] for details).

Proof of Theorem 7.2:

We begin with the proof of part (a). Our discussion preceding the statement of the result establishes that $B_{\mathfrak{T}}$ is an upper bound on A. It remains to establish the strict convexity, and resulting uniqueness of the global optimum. Note that the cost function defining the function $B_{\mathfrak{T}}$ consists of a linear term $\langle \theta, \tau \rangle$ and a convex combination $\mathbb{E}_\rho[-A^*(\tau(T))]$ of tree entropies, and hence is concave. Moreover, the constraint set $\mathbb{L}(G)$ is a polytope, and hence convex. Therefore, the variational problem (7.11) is always convex. We now establish unique-

ness of the optimum when $\rho_e > 0$. To simplify details of the proof, we assume without loss of generality that we are working in a minimal representation. (If the variational problem is formulated in an over-complete representation, it can be reduced to an equivalent minimal formulation.) To establish uniqueness, it suffices to establish that the function $\mathbb{E}_\rho[A^*(\tau(T))]$ is strictly convex when $\rho_e > 0$. This function is a convex combination of functions of the form $A^*(\tau(T))$, each of which is strictly convex in the (nonzero) components of $\tau(T)$, but independent of the other components in the full vector τ. For any vector $\lambda \in \mathbb{R}^d$, define $\Pi^T(\lambda)_\alpha = \lambda_\alpha$ if $\alpha \in \mathcal{I}(T)$, and $\Pi^T(\lambda)_\alpha = 0$ otherwise. We then have

$$\langle \lambda, \nabla^2 A^*(\tau(T))\lambda \rangle = \langle \Pi^T(\lambda), \nabla^2 A^*(\tau(T))\Pi^T(\lambda) \rangle \geq 0,$$

with strict inequality unless $\Pi^T(\lambda) = 0$. Now the condition $\rho_e > 0$ for all $e \in E$ ensures that $\lambda \neq 0$ implies that $\Pi^T(\lambda)$ must be different from zero for at least one tree T'. Therefore, for any $\lambda \neq 0$, we have

$$\langle \lambda, \mathbb{E}_\rho[\nabla^2 A^*(\tau(T))] \, \lambda \rangle \geq \langle \Pi^{T'}(\lambda), \nabla^2 A^*(\tau(T'))\Pi^{T'}(\lambda) \rangle \, > \, 0,$$

which establishes the assertion of strict convexity.

We now turn to proof of part (b). Derivation of the TRW-S updates (7.12) is entirely analogous to the proof of Theorem 4.2: setting up the Lagrangian, taking derivatives, and performing some algebra yields the result (see the paper [246] for full details). The uniqueness follows from the strict convexity established in part (a). $\square$

7.2.2   Reweighted Kikuchi Approximations

A natural extension of convex combinations of trees, entirely analogous to the transition from the Bethe to Kikuchi variational problems, is to take convex combinations of hypertrees. This extension was sketched out in Wainwright et al. [246], and has been further studied by other researchers [251, 260]. Here we describe the basic idea with one illustrative example. For a given treewidth t, consider the set of all hypertrees of width $\mathfrak{T}(t)$ of width less than or equal to t. Of course, the underlying assumption is that t is sufficiently small that performing exact computations on hypertrees of this width is feasible. Applying Proposition 7.1 to a convex combination based on a probability distribution

$\rho = (\rho(T), \, T \in \mathfrak{T}(t))$ over the set of all hypertrees of width at most t, we obtain the following upper bound on the entropy:

$$H(\mu) \leq \mathbb{E}_\rho[H(\mu(T))] = -\sum_T \rho(T)H(\mu(T)). \qquad (7.14)$$

For a fixed distribution ρ, our strategy is to optimize the right-hand side of this upper bound over all pseudomarginals that are consistent on each hypertree. The resulting constraint set is precisely the polytope $\mathbb{L}_t(G)$ defined in Equation (4.51). Overall, the hypertree analog of Theorem 7.2 asserts that the log partition function A is upper bounded by the hypertree-based convex surrogate:

$$A(\theta) \leq B_{\mathfrak{T}(t)}(\theta; \rho) := \max_{\tau \in \mathbb{L}(G)} \left\{ \langle \tau, \theta \rangle + \mathbb{E}_\rho[H(\tau(T))] \right\}. \qquad (7.15)$$

(Again, we switch from μ to τ to reflect the fact that elements of the constraint set $\mathbb{L}(G)$ need not be globally realizable.) Note that the cost function in this optimization problem is concave for all choices of distributions ρ over the hypertrees. The variational problem (7.15) is the hypertree analog of Equation (7.11); indeed, it reduces to the latter equation in the special case $t = 1$.

Example 7.2 (Convex Combinations of Hypertrees). Let us derive an explicit form of Equation (7.15) for a particular hypergraph and choice of hypertrees. The original graph is the 3×3 grid, as illustrated in the earlier Figure D.1(a). Based on this grid, we construct the augmented hypergraph shown in Figure 7.2(a), which has the hyperedge set E given by

$$\{(1245), (2356), (4578), (5689), (25), (45), (56), (58), (5), (1), (3), (7), (9)\}.$$

It is straightforward to verify that it satisfies the single counting criterion (see Appendix D for background).

Now consider a convex combination of four hypertrees, each obtained by removing one of the 4-hyperedges from the edge set. For instance, shown in Figure 7.2(b) is one particular acyclic substructure T^1 with hyperedge set $E(T^1)$

$$\{(1245), (2356), (4578), (25), (45), (56), (58), (5), (1), (3), (7), (9)\},$$
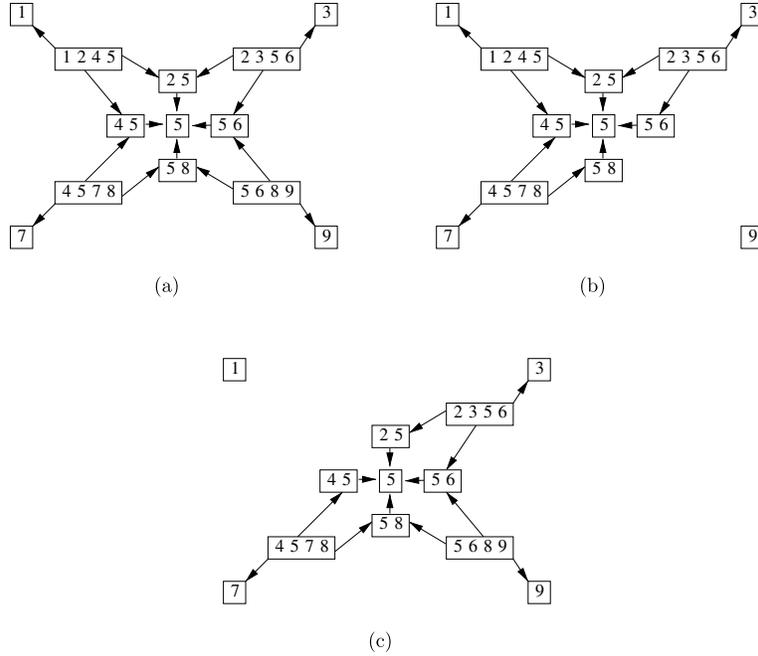
Fig. 7.2 Hyperforests embedded within augmented hypergraphs. (a) An augmented hypergraph for the 3×3 grid with maximal hyperedges of size four that satisfies the single counting criterion. (b) One hyperforest of width three embedded within (a). (c) A second hyperforest of width three.

obtained by removing (5689) from the full hyperedge set E. To be precise, the structure T^1 so defined is a spanning hyperforest, since it consists of two connected components (namely, the isolated hyperedge (9) along with the larger hypertree). This choice, as opposed to a spanning hypertree, turns out to simplify the development to follow. Figure 7.2(c) shows the analogous spanning hyperforest T^2 obtained by removing hyperedge (1245); the final two hyperforests T^3 and T^4 are defined analogously.

To specify the associated hypertree factorization, we first compute the form of φ_h for the maximal hyperedges (i.e., of size four). For instance, looking at the hyperedge $h = (1245)$, we see that hyperedges (25), (45), (5), and (1) are contained within it. Thus, using the definition in Equation (4.40), we write (suppressing the functional

dependence on x):

$$\varphi_{1245} = \frac{\tau_{1245}}{\varphi_{25}\,\varphi_{45}\,\varphi_5\varphi_1} = \frac{\tau_{1245}}{\frac{\tau_{25}}{\tau_5}\frac{\tau_{45}}{\tau_5}\tau_5\tau_1} = \frac{\tau_{1245}\,\tau_5}{\tau_{25}\tau_{45}\tau_1}.$$

Having calculated all the functions φ_h, we can combine them, using the hypertree factorization (4.42), in order to obtain the following factorization for a distribution on T^1:

$$p_{\tau(T^1)}(x) = \left[\frac{\tau_{1245}\,\tau_5}{\tau_{25}\tau_{45}\tau_1}\right]\left[\frac{\tau_{2356}\,\tau_5}{\tau_{25}\tau_{56}\tau_3}\right]\left[\frac{\tau_{4578}\,\tau_5}{\tau_{45}\tau_{58}\tau_7}\right]$$

$$\times \left[\frac{\tau_{25}}{\tau_5}\right]\left[\frac{\tau_{45}}{\tau_5}\right]\left[\frac{\tau_{56}}{\tau_5}\right]\left[\frac{\tau_{58}}{\tau_5}\right][\tau_1][\tau_3][\tau_5][\tau_7][\tau_9]. \qquad (7.16)$$

Here each term within square brackets corresponds to φ_h for some hyperedge $h \in E(T^1)$; for instance, the first three terms correspond to the three maximal 4-hyperedges in T^1. Although this factorization could be simplified, leaving it in its current form makes the connection to Kikuchi approximations more explicit; in particular, the factorization (7.16) leads immediately to a decomposition of the entropy. In an analogous manner, it is straightforward to derive factorizations and entropy decompositions for the remaining three hyperforests $\{T^i, i = 2, 3, 4\}$.

Now let $E_4 = \{(1245), (2356), (5689), (4578)\}$ denote the set of all 4-hyperedges. We then form the convex combination of the four (negative) entropies with uniform weight $1/4$ on each T^i; this weighted sum $\sum_{i=1}^4 \frac{1}{4}A^*(\tau(T^i))$ takes the form:

$$\frac{3}{4}\sum_{h \in E_4}\sum_{x_h}\tau_h(x_h)\log\varphi_h(x_h) + \sum_{s \in \{2,4,6,8\}}\sum_{x_{s5}}\tau_{s5}(x_{s5})\log\frac{\tau_{s5}(x_{s5})}{\tau_5(x_5)}$$

$$+ \sum_{s \in \{1,3,5,7,9\}}\sum_{x_s}\tau_s(x_s)\log\tau_s(x_s). \qquad (7.17)$$

The weight $3/4$ arises because each of the maximal hyperedges $h \in E_4$ appears in three of the four hypertrees. All of the (nonmaximal) hyperedge terms receive a weight of one, because they appear in all four hypertrees. Overall, then, these weights represent hyperedge appearance probabilities for this particular example, in analogy to ordinary edge appearance probabilities in the tree case. We now simplify the

expression in Equation (7.17) by expanding and collecting terms; doing so yields that the sum $-\sum_{i=1}^{4}\frac{1}{4}A^{*}(\tau(T^{i}))$ is equal to the following weighted combination of entropies:

$$\frac{3}{4}[H_{1245} + H_{2356} + H_{5689} + H_{4578}] - \frac{1}{2}[H_{25} + H_{45} + H_{56} + H_{58}]$$
$$+\frac{1}{4}[H_1 + H_3 + H_7 + H_9]. \tag{7.18}$$

If, on the other hand, starting from Equation (7.17) again, suppose that we included each maximal hyperedge with a weight of 1, instead of 3/4. Then, after some simplification, we would find that (the negative of) Equation (7.17) is equal to the following combination of local entropies

$$[H_{1245} + H_{2356} + H_{5689} + H_{4578}] - [H_{25} + H_{45} + H_{56} + H_{58}] + H_5,$$

which is equivalent to the Kikuchi approximation derived in Example 4.6. However, the choice of all ones for the hyperedge appearance probabilities is invalid — that is, it could never arise from taking a convex combination of hypertree entropies.

More generally, any entropy approximation formed by taking such convex combinations of hypertree entropies will necessarily be convex. In contrast, with the exception of certain special cases [110, 167, 188, 189], Kikuchi and other hypergraph-based entropy approximations are typically not convex. In analogy to the tree-reweighted sum-product algorithm, it is possible to develop hypertree-reweighted forms of generalized sum-product updates [251, 260]. With a suitable choice of convex combination, the underlying variational problem will be strictly convex, and so the associated hypertree-reweighted sum-product algorithms will have a unique fixed point. An important difference from the case of ordinary trees, however, is that optimization of the hyperedge weights ρ_e cannot be performed exactly using a greedy algorithm. Indeed, the hypertree analog of the maximum weight spanning tree problem is known to be NP-hard [129].

7.2.3 Convex Forms of Expectation-Propagation

As discussed in Section 4.3, the family of expectation-propagation algorithms [172, 175], as well as related moment-matching algorithms [64, 115, 185], can be understood in terms of term-by-term entropy approximations. From this perspective, it is clear how to derive convex analogs of the variational problem (4.69) that underlies expectation-propagation.

In particular, recall from Equation (4.75) the form of the term-by-term entropy approximation

$$H_{\mathrm{ep}}(\tau,\widetilde{\tau}) := H(\tau) + \sum_{\ell=1}^{d_I} \big[H(\tau,\widetilde{\tau}^\ell) - H(\tau) \big],$$

where $H(\tau)$ represents the entropy of the base distribution, and $H(\tau,\widetilde{\tau}^\ell)$ represents the entropy of the base plus ℓth term. Implicitly, this entropy approximation is weighting each term $i = 1,\ldots,d_I$ with weight one. Suppose instead that we consider a family of nonnegative weights $\{\rho(1),\ldots,\rho(d_I)\}$ over the different terms, and use the reweighted term-by-term entropy approximation

$$H_{\mathrm{ep}}(\tau,\widetilde{\tau};\rho) := H(\tau) + \sum_{\ell=1}^{d_I} \rho(\ell)\big[H(\tau,\widetilde{\tau}^\ell) - H(\tau) \big]. \qquad (7.19)$$

For suitable choice of weights — for instance, if we enforce the constraint $\sum_{\ell=1}^{d_I} \rho(\ell) = 1$ — this reweighted entropy approximation is concave. Other choices of nonnegative weights could also produce concave entropy approximations. The EP outer bound $\mathcal{L}(\phi;\Phi)$ on the set $\mathcal{M}(\phi)$, as defined in Equation (4.67), is a convex set by definition. Consequently, if we combine a convex entropy approximation (7.19) with this outer bound, we obtain "convexified" forms of the EP variational problem. Following the line of development in Section 4.3, it is also possible to derive Lagrangian algorithms for solving these optimization problems. Due to underlying structure, the updates can again be cast in terms of moment-matching steps. In contrast to standard forms of EP, a benefit of these algorithms would be the existence of unique fixed points, assured by the convexity of the underlying variational principle. With different motivations, not directly related to convexity and

uniqueness, Minka [173] has discussed reweighted forms of EP, referred to as "power EP." Seeger et al. [214] reported that reweighted forms of EP appear to have better empirical convergence properties than standard EP. It would be interesting to study connections between the convexity of the EP entropy approximation, and the convergence and stability properties of the EP updates.

7.2.4   Other entropy approximations

Trees (and the associated hierarchy of hypertrees) represent the best known classes of tractable graphical models. However, exact computation is also tractable for certain other classes of graphical models. Unlike hypertrees, certain models may be intractable in general, but tractable for specific settings of the parameters. For instance, consider the subclass of planar graphs (which can be drawn in the 2D plane without any crossing of the edges, for instance 2D grid models). Although exact computation for an arbitrary Markov random field on a planar graph is known to be intractable, if one considers only binary random variables without any observation potentials (i.e., Ising models of the form $p_\theta(x) \propto \exp\{\sum_{(s,t)\in E} \theta_{st} x_s x_t\}$), then it is possible to compute the cumulant function $A(\theta)$ exactly, using clever combinatorial reductions due independently to Fisher [81] and Kastelyn [132]. Globerson and Jaakkola [94] exploit these facts in order to derive a variational relaxation based on convex combination of planar subgraphs, as well as iterative algorithms for computing the optimal bound. They provide experimental results for various types of graphs (both grids and fully connected graphs), with results superior to the TRW relaxation (7.11) for most coupling strengths, albeit with higher computational cost. As they point out, it should also be possible to combine hypertrees with planar graphs to form composite approximations.

7.3   Other Convex Variational Methods

The previous section focused exclusively on variational methods based on the idea of convex combinations, as stated in Equation (7.7). However, not all convex variational methods need arise in this way; more generally, a convex variational method requires only (a) some type of

convex approximation to the dual function A^* (negative entropy), and (b) convex outer bound on the set $\mathcal{M}$. This basic two-step procedure can be used to derive many convex relaxations of the marginalization problem.

7.3.1   Semidefinite Constraints and Log-determinant Bound

We illustrate this two-step procedure by describing a log-determinant relaxation for discrete Markov random fields [248]. This method differs qualitatively from the previously described Bethe/Kikuchi methods, in that it uses a semidefinite outer bound on the marginal polytope $\mathbb{M}(G)$. Although the basic ideas are more generally applicable, let us focus for concreteness on the case of a binary random vector $X \in \{0,1\}^m$, with a distribution in the Ising form:

$$p_\theta(x) = \exp\left\{ \sum_{s \in V} \theta_s x_s + \sum_{(s,t)} \theta_{st} x_s x_t - A(\theta) \right\}. \qquad (7.20)$$

Without loss of generality,[1] we assume that the underlying graph is the complete graph K_m, so that the marginal polytope of interest is $\mathbb{M}(K_m)$. With this set-up, the mean parameterization consists of the vector $\mu \in \mathbb{M}(K_m) \subset \mathbb{R}^{m + \binom{m}{2}}$, with elements $\mu_s = \mathbb{E}[X_s]$ for each node, and $\mu_{st} = \mathbb{E}[X_s X_t]$ for each edge.

First, we describe how to approximate the negative dual function $-A^*$, or entropy function. An important characterization of the multivariate Gaussian is as the maximum entropy distribution subject to covariance constraints [57]. In particular, the differential entropy $h(\widetilde{X})$ of any continuous random vector $\widetilde{X}$ is upper bounded by the entropy of a Gaussian with matched covariance. In analytical terms, we have

$$h(\widetilde{X}) \leq \frac{1}{2} \log \det \mathrm{cov}(\widetilde{X}) + \frac{m}{2} \log(2\pi e), \qquad (7.21)$$

where $\mathrm{cov}(\widetilde{X})$ is the covariance matrix of $\widetilde{X}$. The upper bound (7.21) is not directly applicable to a random vector taking values in a discrete space (since differential entropy in this case diverges to minus

---

[1] A problem defined on an arbitrary $G = (V, E)$ can be embedded into the complete graph by setting $\theta_{st} = 0$ for all $(s,t) \notin E$.

infinity). Therefore, in order to exploit this bound for the discrete random vector $X \in \{0,1\}^m$ of interest, it is necessary to construct a suitably matched continuous version of X. One method to do so is by the addition of an independent random vector U, such that the atoms in the distribution of X are smoothed out. In particular, let us define the continuous random vector $\widetilde{X} := X + U$, where U is independent of X, with independent components distributed uniformly as $U_s \sim \mathcal{U}[-\frac{1}{2}, \frac{1}{2}]$. A key property of this construction is that it matches the discrete entropy of X with the differential entropy of $\widetilde{X}$ — that is, $h(\widetilde{X}) = H(X)$; see Wainwright and Jordan [248] for the proof. Since X and U are continuous by construction, a straightforward computation yields $\operatorname{cov}(\widetilde{X}) = \operatorname{cov}(X) + \frac{1}{12}I_m$, so Equation (7.21) yields the upper bound

$$H(X) \leq \frac{1}{2}\log\det\left[\operatorname{cov}(X) + \frac{1}{12}I_m\right] + \frac{m}{2}\log(2\pi e).$$

Second, we need to provide an outer bound on the marginal polytope for which the entropy approximation above is well defined. The simplest such outer bound is obtained by observing that the covariance matrix $\operatorname{cov}(X)$ must always be positive semidefinite. (Indeed, for any vector $a \in \mathbb{R}^m$, we have $\langle a, \operatorname{cov}(X)a \rangle = \operatorname{var}(\langle a, X \rangle) \geq 0$.) Note that elements of the matrix $\operatorname{cov}(X)$ can be expressed in terms of the mean parameters μ_s and μ_{st}. In particular, consider the $(m+1) \times (m+1)$ matrix

$$N_1[\mu] = \mathbb{E}\left[\begin{bmatrix} 1 \\ X \end{bmatrix} \begin{bmatrix} 1 & X \end{bmatrix}\right]$$

$$= \begin{bmatrix} 1 & \mu_1 & \mu_2 & \cdots & \cdots & \mu_{m-1} & \mu_m \\ \mu_1 & \mu_1 & \mu_{12} & \cdots & \cdots & \mu_{1(m-1)} & \mu_{1m} \\ \mu_2 & \mu_{21} & \mu_2 & \mu_{23} & \cdots & \cdots & \mu_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mu_{m-1} & \mu_{(m-1)1} & \cdots & \cdots & \cdots & \mu_{(m-1)} & \mu_{(m-1)m} \\ \mu_m & \mu_{m1} & \cdots & \cdots & \cdots & \mu_{m(m-1)} & \mu_m \end{bmatrix}.$$

By the Schur complement formula [114], the constraint $N_1[\mu] \succeq 0$ is equivalent to the constraint $\operatorname{cov}(X) \succeq 0$. Consequently, the set

$$\mathbb{S}_1(K_m) := \{\tau \in \mathbb{R}^{m+\binom{m}{2}} \mid N_1[\tau] \succeq 0\}$$

is an outer bound on the marginal polytope $\mathbb{M}(K_m)$. It is not difficult to see that the set $\mathbb{S}_1(K_m)$ is closed, convex, and also bounded, since it is contained within the hypercube $[0,1]^{m+\binom{m}{2}}$. Again, our switch in notation from μ to τ is deliberate, since vectors τ belonging to the semidefinite constraint set $\mathbb{S}_1(K_m)$ need not be globally consistent (see Section 9 for more details).

We now have the two necessary ingredients to define a convex variational principle based on a log-determinant relaxation [248]. In addition to the constraint $\tau \in \mathbb{S}_1(K_m)$, one might imagine enforcing other constraints (e.g., the linear constraints defining the set $\mathbb{L}(G)$). Accordingly, let $\mathbb{B}(K_m)$ denote any convex and compact outer bound on $\mathbb{M}(K_m)$ that is contained within $\mathbb{S}_1(K_m)$. With this notation, the cumulant function $A(\theta)$ is upper bounded by the log-determinant convex surrogate $B_{\mathrm{LD}}(\theta)$, defined by the variational problem

$$\max_{\tau \in \mathbb{B}(K_m)} \left\{ \langle \theta, \tau \rangle + \frac{1}{2} \log \det \left[N_1[\tau] + \frac{1}{12} \mathrm{blkdiag}[0, I_m] \right] \right\} + \frac{m}{2} \log(2\pi e),$$
(7.22)

where $\mathrm{blkdiag}[0, I_m]$ is an $(m+1) \times (m+1)$ block-diagonal matrix. By enforcing the inclusion $\mathbb{B}(K_m) \subseteq \mathbb{S}_1(K_m)$ we guarantee that the matrix $N_1[\tau]$, and hence the matrix sum

$$N_1[\tau] + \frac{1}{12} \mathrm{blkdiag}[0, I_m]$$

will always be positive semidefinite. Importantly, the optimization problem (7.22) is a standard determinant maximization problem, for which efficient interior point methods have been developed [e.g., 235]. Wainwright and Jordan [248] derive an efficient algorithm for solving a slightly weakened form of the log-determinant problem (7.22), and provide empirical results for various types of graphs. Banerjee et al. [8] exploit the log-determinant relaxation (7.22) for a sparse model selection procedure, and develop coordinate-wise algorithms for solving a broader class of convex programs.

7.3.2 Other Entropy Approximations and Polytope Bounds

There are many other approaches, using different approximations to the entropy and outer bounds on the marginal polytope, that lead

to interesting convex variational principles. For instance, Globerson and Jaakkola [95] exploit the notion of conditional entropy decomposition in order to construct whole families of upper bounds on the entropy, including as a special case the hypertree-based entropy bounds that underlie the reweighted Bethe/Kikuchi approaches. Sontag and Jaakkola [219] examine the effect of tightened outer bounds on the marginal polytope, including the so-called cycle inequalities [69, 187]. Among other issues, they examine the differing effects of changing the entropy approximation versus refining the outer bound on the marginal polytope. These questions remain an active and ongoing area of research.

7.4 Algorithmic Stability

As discussed earlier, an obvious benefit of approximate marginalization methods based on convex variational principles is the resulting uniqueness of the minima, and hence lack of dependence on algorithmic details, or initial conditions. In this section, we turn to a less obvious but equally important benefit of convexity, of particular relevance in a statistical setting. More specifically, in typical applications of graphical models, the model parameters or graph topologies themselves are uncertain. (For instance, they may have been estimated from noisy or incomplete data.) In such a setting, then, it is natural to ask whether the output of a given variational method remains stable under small perturbations to the model parameters. Not all variational algorithms are stable in this sense; for instance, both mean field methods and the ordinary sum-product algorithm can be highly unstable, with small perturbations in the model parameters leading to very large changes in the fixed point(s) and outputs of the algorithm.

In this section, we show that any variational method based on a suitable convex optimization problem — in contrast to mean field and ordinary sum-product — has an associated guarantee of Lipschitz stability. In particular, let $\tau(\theta)$ denote the output when some variational method is applied to the model p_θ. Given two norms $\|\cdot\|_a$ and $\|\cdot\|_b$, we say that the method is *globally Lipschitz stable* if there exists some

finite constant L such that

$$\|\tau(\theta) - \tau(\theta')\|_a \leq L\|\theta - \theta'\|_b, \tag{7.23}$$

for any $\theta, \theta' \in \Omega$. This definition is one way of formalizing the notion of *algorithmic stability*: namely, that if the difference $\|\theta - \theta'\|_b$ between models indexed by θ and θ' is small, then the difference $\|\tau(\theta) - \tau(\theta')\|_a$ between the algorithm's outputs is correspondingly small. As an illustration, Figure 7.3 compares the behavior of the ordinary sum-product algorithm — known to be instable in certain regimes — to the tree-reweighted sum-product algorithm. Note how the sum-product algorithm degrades rapidly beyond a certain critical coupling strength, whereas the performance of the tree-reweighted algorithm varies smoothly as a function of the coupling strength.

Let us now try to gain some theoretical insight into which variational methods satisfy this Lipschitz property. Consider a generic variational
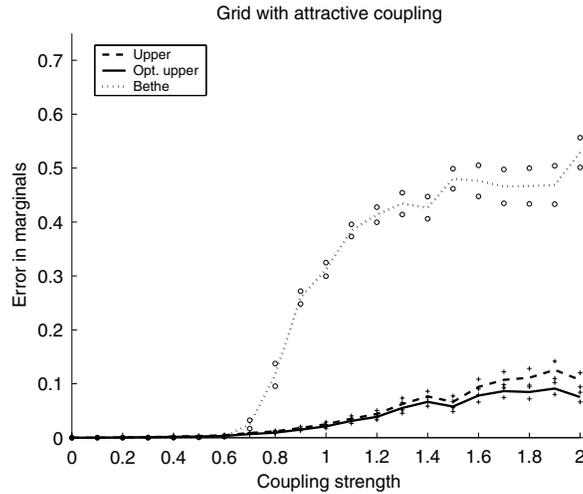

Fig. 7.3 Contrast between the instability of ordinary sum-product and stability of tree-reweighted sum-product [246]. Plots show the error between the true marginals and correct marginals versus the coupling strength in a binary pairwise Markov random field. Note that the ordinary sum-product algorithm is very accurate up to a critical coupling (≈ 0.70), after which it degrades rapidly. On the other hand, the performance of TRW message-passing varies smoothly as a function of the coupling strength. The plot shows two versions of the TRW sum-product algorithm, either based on uniform edge weights $\rho_{st} = 1/2$, or edge weights optimized to minimize the upper bound.

method of the form:

$$B(\theta) = \sup_{\tau \in \mathcal{L}} \{\langle \theta, \tau \rangle - B^*(\tau)\}, \tag{7.24}$$

where $\mathcal{L}$ is a convex outer bound on the set $\mathcal{M}$, and B^* is a convex approximation to the dual function A^*. Let us consider the properties of the surrogate function B that we have defined. As mentioned previously, it is always a convex function of θ. In terms of properties beyond convexity, they are determined by the nature of the function B^*. For instance, suppose that B^* is strictly convex; then the optimum (7.24) is uniquely attained, so that Danskin's theorem [21] ensures that B is differentiable.

In terms of Lipschitz stability, it turns out that a somewhat stronger notion of convexity is required; in particular, a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is *strongly convex* with parameter c if for all $y, z \in \mathbb{R}^d$,

$$f(z) \geq f(y) + \langle \nabla f(y), z - y \rangle + \frac{c}{2} \|z - y\|^2. \tag{7.25}$$

Note that any differentiable convex function satisfies this inequality (7.25) with $c = 0$; the essence of strong convexity is that the same inequality is satisfied with slack $\frac{c}{2} \|z - y\|^2$. (In fact, an equivalent characterization of strong convexity is to require that the function $f(z) - \frac{c}{2} \|z\|^2$ be convex [112].) In the setting of the convex surrogate (7.24), suppose that the function B is strictly convex, and the (negative) entropy approximation B^* is strongly convex, say with parameter $1/L > 0$. Under this assumption, it can be shown [241] that the output $\tau(\theta) = \nabla B(\theta)$ of the variational method is Lipschitz stable, in the sense of definition (7.23), with parameter L and norms $\|\cdot\|_a = \|\cdot\|_b = \|\cdot\|_2$.

Thus, when using variational methods of the form (7.24), the issue of algorithmic stability reduces to strong convexity of the negative entropy approximation B^*. What existing variational methods are based on strongly convex entropy approximations? For instance, the tree-reweighted Bethe entropy (7.10) is strongly convex [241] for any pairwise Markov random field, so that as a corollary, the tree-reweighted sum-product algorithm is guaranteed to be globally Lipschitz stable. This Lipschitz stability provides a theoretical explanation of the behavior illustrated in Figure 7.3. Similarly, empirical

results due to Wiegerinck [260] show that suitably reweighted forms of generalized belief propagation (GBP) also behave in a stable manner (unlike standard GBP), and this stability can be confirmed theoretically via strong convexity. It can also be shown that the log-determinant relaxation (7.22) is Lipschitz stable.

7.5 Convex Surrogates in Parameter Estimation

Until now, we have discussed convex variational methods in the context of computing approximations to the cumulant function value $A(\theta)$, as well as approximate mean parameters μ. Here, we discuss an alternative use of convex surrogates of the form (7.24), namely for approximate parameter estimation. Recall from Section 6 our discussion of maximum likelihood (ML) estimation in exponential families: it entails maximizing the rescaled log likelihood, given by

$$\ell(\theta; X_1^n) = \ell(\theta) = \langle \theta, \widehat{\mu} \rangle - A(\theta), \qquad (7.26)$$

where $\widehat{\mu} := \frac{1}{n} \sum_{i=1}^n \phi(X^i)$ is the empirical expectation of the sufficient statistics given the data $X_1^n = (X^1, \ldots, X^n)$. Recall also that the derivative of the log likelihood takes the form $\nabla \ell(\theta) = \widehat{\mu} - \mu(\theta)$, where $\mu(\theta) = \mathbb{E}_\theta[\phi(X)]$ are the mean parameters under the current model parameter. As discussed in Section 6, this fact motivates a natural approach to approximate parameter estimation, in which the true model parameters $\mu(\theta)$ are approximated by the output $\tau(\theta)$ of some variational method. For instance, this approach has frequently been suggested, using sum-product to compute the approximate mean parameters $\tau(\theta)$. Such heuristics may yield good results in some settings; however, if the underlying variational method is nonconvex, such a heuristic cannot be interpreted as minimizing an approximation to the likelihood. Indeed, given multiple fixed points, its behavior can be unstable and erratic; the papers [241, 243] provide some cautionary instances of poor behavior with ordinary sum-product.

7.5.1 Surrogate Likelihoods

On the other hand, the perspective of convex surrogates to A, of the form (7.24), suggests a related but more principled method for

approximate ML estimation, based on maximizing a concave approximation to the likelihood. Let us assume that the outer bound $\mathcal{L}$ is compact and the negative entropy approximation B^* is strictly convex, so that the maximum (7.24) is uniquely attained, say at some $\tau(\theta) \in \mathcal{L}$. Recall that Danskin's theorem [21] shows that the convex surrogate B is differentiable, with $\nabla B(\theta) = \tau(\theta)$. (Note the parallel with the gradient ∇A, and its link to the mean parameterization, as summarized in Proposition 3.1). Given a convex surrogate B, let us define the associated *surrogate likelihood* as

$$\ell_B(\theta; X_1^n) = \ell_B(\theta) := \langle \theta, \widehat{\mu} \rangle - B(\theta). \qquad (7.27)$$

Assuming that B is strictly convex, then this surrogate likelihood is strictly concave, and so (at least for compact parameter spaces Ω), we have the well-defined *surrogate ML estimate*

$$\widetilde{\theta}_B := \arg\max_{\theta \in \Omega} \ell_B(\theta; X_1^n).$$

Note that when B upper bounds the true cumulant function (as in the convex variational methods described in Section 7.1), then the surrogate likelihood ℓ_B lower bounds the true likelihood, and $\widetilde{\theta}_B$ is obtained from maximizing this lower bound.

7.5.2   Optimizing Surrogate Likelihoods

In general, it is relatively straightforward to compute the surrogate estimate $\widetilde{\theta} = \widetilde{\theta}_B$. Indeed, from our discussion above, the derivative of the surrogate likelihood is $\nabla \ell_B(\theta) = \widehat{\mu} - \tau(\theta)$, where $\tau(\theta) = \nabla B(\theta)$ are the approximate mean parameters computed by the variational method associated with B. Since the optimization problem is concave, a standard gradient ascent algorithm (among other possibilities) could be used to compute $\widetilde{\theta}_B$.

Interestingly, in certain cases, the unregularized surrogate MLE can be specified in closed form, without the need for any optimization. As an example, consider the reweighted Bethe surrogate (7.11), and the associated surrogate likelihood. The surrogate MLE is defined by the fixed point relation $\nabla \ell_B(\widetilde{\theta}) = 0$, which has a very intuitive interpretation. Namely, the model parameters $\widetilde{\theta}$ are chosen such that, when

the TRW sum-product updates (7.12) are applied to the model $p_{\widetilde{\theta}}$, the resulting pseudo mean parameters $\tau(\widetilde{\theta})$ computed by the algorithm are equal to the empirical mean parameters $\widehat{\mu}$. Assuming that the empirical mean vector has strictly positive elements, the unique vector $\widetilde{\theta}$ with this property has the closed-form expression:

$$\forall\, s \in V,\, j \in \mathcal{X}_s, \quad \widetilde{\theta}_{s;j} \;=\; \log \widehat{\mu}_{s;j}, \quad \text{and} \qquad (7.28\text{a})$$

$$\forall\, (s,t) \in E,\, (j,k) \in \mathcal{X}_s \times \mathcal{X}_t, \quad \widetilde{\theta}_{st;jk} \;=\; \rho_{st}\, \log \frac{\widehat{\mu}_{st;jk}}{\widehat{\mu}_{s;j}\widehat{\mu}_{t;k}}. \qquad (7.28\text{b})$$

Here $\widehat{\mu}_{s;j} = \widehat{\mathbb{E}}[\mathbb{I}_j(X_s)]$ is the empirical probability of the event $\{X_s = j\}$, and $\widehat{\mu}_{st;jk}$ is the empirical probability of the event $\{X_s = j, X_t = k\}$. By construction, the vector $\widetilde{\theta}$ defined by Equation (7.28) has the property that $\widehat{\mu}$ are the pseudomarginals associated with mode $p_{\widetilde{\theta}}$. The proof of this fact relies on the tree-based reparameterization interpretation of the ordinary sum-product algorithm, as well as its reweighted extensions [243, 246]; see Section 4.1.4 for details on reparameterization. Similarly, closed-form solutions can be obtained for reweighted forms of Kikuchi and region graph approximations (Section 7.2.2), and convexified forms of expectation-propagation and other moment-matching algorithms (Section 7.2.3). Such a closed-form solution eliminates the need for any iterative algorithms in the complete data setting (as described here), and can substantially reduce computational overhead if a variational method is used within an inner loop in the incomplete data setting.

7.5.3 Penalized Surrogate Likelihoods

Rather than maximizing the log likelihood (7.26) itself, it is often better to maximize a penalized likelihood function $\widetilde{\ell}(\theta; \lambda) := \ell(\theta) - \lambda R(\theta)$, where $\lambda > 0$ is a regularization constant and R is a penalty function, assumed to be convex but not necessarily differentiable (e.g., ℓ_1 penalization). Such procedures can be motivated either from a frequentist perspective (and justified in terms of improved loss and risk), or from a Bayesian perspective, where the quantity $\lambda R(\theta)$ might arise from a prior over the parameter space. If maximization of the likelihood is intractable then regularized likelihood is still intractable, so let

us consider approximate procedures based on maximizing a regularized surrogate likelihood (RSL) $\widetilde{\ell}_B(\theta; \lambda) := \ell_B(\theta) - \lambda R(\theta)$. Again, given that ℓ_B is convex and differentiable, this optimization problem could be solved by a variety of standard methods.

Here, we describe an alternative formulation of optimizing the penalized surrogate likelihood, which exploits the variational manner in which the surrogate B was defined (7.24). If we reformulate the problem as one of minimizing the negative RSL, substituting the definition (7.24) yields the equivalent saddlepoint problem:

$$
\inf_{\theta \in \Omega} \left\{ -\langle \theta, \widehat{\mu} \rangle + B(\theta) + \lambda R(\theta) \right\}
$$

$$
= \inf_{\theta \in \Omega} \left\{ -\langle \theta, \widehat{\mu} \rangle + \sup_{\tau \in \mathcal{L}} \{ \langle \theta, \tau \rangle - B^*(\tau) \} + \lambda R(\theta) \right\}
$$

$$
= \inf_{\theta \in \Omega} \sup_{\tau \in \mathcal{L}} \{ \langle \theta, \tau - \widehat{\mu} \rangle - B^*(\tau) + \lambda R(\theta) \}.
$$

Note that this saddlepoint problem involves convex minimization and concave maximization; therefore, under standard regularity assumptions (e.g., either compactness of the constraint sets, or coercivity of the function; see Section VII of Hiriart-Urruty and Lemaréchal [112]), we can exchange the sup and inf to obtain

$$
\inf_{\theta \in \Omega} \{ -\ell_B(\theta) + \lambda R(\theta) \} = \sup_{\tau \in \mathcal{L}} \inf_{\theta \in \Omega} \{ \langle \theta, \tau - \widehat{\mu} \rangle - B^*(\tau) + \lambda R(\theta) \}
$$

$$
= \sup_{\tau \in \mathcal{L}} \left\{ -B^*(\tau) - \lambda \sup_{\theta \in \Omega} \left\{ \langle \theta, \frac{\tau - \widehat{\mu}}{\lambda} \rangle - R(\theta) \right\} \right\}
$$

$$
= \sup_{\tau \in \mathcal{L}} \left\{ -B^*(\tau) - \lambda \, R_\Omega^* \left(\frac{\tau - \widehat{\mu}}{\lambda} \right) \right\}, \qquad (7.29)
$$

where R_Ω^* is the conjugate dual function of $R(\theta) + \mathbb{I}_\Omega(\theta)$. Consequently, assuming that it is feasible to compute the dual function R_Ω^*, maximizing the RSL reduces to an unconstrained optimization problem involving the function B^* used to define the surrogate function B. We illustrate with a particular family of examples:

Example 7.3 (Regularized Surrogate Bethe Likelihood). Suppose that we use the surrogate function $B_{\mathfrak{T}}(\theta; \rho_e)$ defined in

Theorem 7.2 to form a surrogate likelihood ℓ_B, and that for our regularizing function, we use an ℓ_q norm $\|\theta\|_q$ with $q \geq 1$. For the discrete Markov random fields to which the reweighted Bethe surrogate $B_{\mathfrak{T}}(\theta; \rho_e)$ applies, the parameter space is simply $\Omega = \mathbb{R}^d$, so that we can calculate

$$
R^*(\tau) = \sup_{\theta \in \mathbb{R}^d} \{\langle \theta, \tau \rangle - \|\theta\|_q\}
$$

$$
= \begin{cases} 0 & \text{if } \|\tau\|_{q'} \leq 1, \quad \text{where } 1/q' = 1 - 1/q, \\ +\infty & \text{otherwise,} \end{cases}
$$

using the fact that $\|\theta\|_q = \sup_{\|\tau\|_{q'} \leq 1} \langle \tau, \theta \rangle$. Thus, the dual RSL problem (7.29) takes the form:

$$
\sup_{\|\tau - \widehat{\mu}\|_{q'} \leq \lambda} \left[-B^*(\tau) \right] = \sup_{\|\tau - \widehat{\mu}\|_{q'} \leq \lambda} \left\{ \sum_{s \in V} H_s(\tau_s) - \sum_{(s,t) \in E} \rho_{st} I_{st}(\tau_{st}) \right\}.
$$

Note that this problem has a very intuitive interpretation: it involves maximizing the Bethe entropy, subject to an $\ell_{q'}$-norm constraint on the distance to the empirical mean parameters $\widehat{\mu}$. As pointed out by several authors [63, 71], a similar dual interpretation also exists for regularized exact maximum likelihood. The choice of regularization translates into a certain uncertainty or robustness constraint in the dual domain. For instance, given ℓ_2 regularization ($q = 2$) on θ, we have $q' = 2$ so that the dual regularization is also ℓ_2-based. On the other hand, for ℓ_1-regularization, ($q = 1$) the corresponding dual norm is ℓ_∞ ($q' = +\infty$), so that we have box constraints in the dual domain.

There remain a large number of open research questions associated with the use of convex surrogates for approximate estimation. An immediate statistical issue is to understand the consistency, asymptotic normality and other issues associated with an estimate $\widetilde{\theta}_B$ obtained from a convex surrogate. These properties are straightforward to analyze, since the estimate $\widetilde{\theta}_B$ is a particular case of an M-estimator [233]. Consequently, under mild conditions on the surrogate (so as to ensure uniform convergence of the empirical version to the population version), we are guaranteed that $\widetilde{\theta}_B$ converges in probability to the minimizer of

the population surrogate likelihood — namely

$$\theta_B^* = \arg\min_{\theta \in \Omega} \big\{ \langle \theta, \mu \rangle - B(\theta) \big\}, \qquad (7.30)$$

where $\mu = \mathbb{E}_{\theta^*}[\phi(X)]$ are the mean parameters under the true distribution p_{θ^*}. When the convex surrogate is strictly convex, the unique minimizer of the population version is the parameter vector θ_B^* such that $\nabla B(\theta_B^*) = \mu$. Consequently, unless the mapping ∇B coincides with the mapping ∇A, the estimators based on convex surrogates of the type described here can be inconsistent, meaning that the random variable $\widetilde{\theta}_B$ may not converge to the true model parameter θ^*.

Of course, such inconsistency is undesirable if the true parameter θ^* is of primary interest. However, in other settings, it may actually be desirable to use an inconsistent estimator. For instance, suppose that we are interested in constructing a model with good predictive power, and moreover that in constructing new predictions, we are constrained to use only "tractable" algorithms. For instance, we might be prevented from computing the exact likelihoods and marginal distributions of a graphical model, but instead required to use only approximate algorithms, such as sum-product or expectation-propagation. In such settings, one can imagine that it might be beneficial to use an "incorrect" model, since we are applying an approximate algorithm to generate predictions. Based on this intuition, Wainwright [241] proves that when performing prediction using the reweighted sum-product algorithm, the parameter estimates from the tree-reweighted Bethe surrogate, even though they are different from the true model, yield lower prediction error than an oracle given access to the true model, but also required to use the reweighted sum-product algorithm. It is likely that such phenomena also arise in other settings where computation is limited.

8

Integer Programming, Max-product, and Linear Programming Relaxations

Thus far, the bulk of our discussion has focused on variational methods that, when applied to a given model p_θ, yield approximations to the mean parameters $\mu = \mathbb{E}_\theta[\phi(X)]$, as well as to the log partition or cumulant function $A(\theta)$. In this section, we turn our attention to a related but distinct problem — namely, that of computing *a mode or most probable configuration* associated with the distribution p_θ. It turns out that the mode problem has a variational formulation in which the set $\mathcal{M}$ of mean parameters once again plays a central role. More specifically, since mode computation is of significant interest for discrete random vectors, the marginal polytopes $\mathbb{M}(G)$ play a central role through much of this section.

8.1 Variational Formulation of Computing Modes

Given a member p_θ of some exponential family, the mode computation problem refers to computing a configuration $x^* \in \mathcal{X}^m$ that belongs to the set

$$\arg \max_{x \in \mathcal{X}^m} p_\theta(x) := \{y \in \mathcal{X}^m \mid p_\theta(y) \geq p_\theta(x) \quad \forall\, x \in \mathcal{X}^m\}. \quad (8.1)$$

We assume that at least one mode exists, so that this $\arg\max$ set is nonempty. Moreover, it is convenient for development in the sequel to note the equivalence

$$\arg \max_{x \in \mathcal{X}^m} p_\theta(x) = \arg \max_{x \in \mathcal{X}^m} \langle \theta, \phi(x) \rangle, \qquad (8.2)$$

which follows from the exponential form of the density p_θ, and the fact that the cumulant function $A(\theta)$ does not depend on x.

It is not immediately obvious how the MAP problem (8.2) is related to the variational principle from Theorem 3.4. As it turns out, the link lies in the notion of the "zero-temperature limit". We begin by providing intuition for the more formal result to follow. From its definition and Theorem 3.4, for any parameter vector $\theta \in \Omega$, the cumulant function has the following two representations:

$$
\begin{aligned}
A(\theta) &:= \log \int \exp\left\{\langle \theta, \phi(x) \rangle\right\} \nu(dx) \\
&= \sup_{\mu \in \mathcal{M}} \left\{\langle \theta, \mu \rangle - A^*(\mu)\right\}. \qquad (8.3)
\end{aligned}
$$

Now suppose that we rescale the canonical parameter θ by some scalar $\beta > 0$. For the sake of this argument, let us assume that $\beta\theta \in \Omega$ for all $\beta > 0$. Such a rescaling will put more weight, in a relative sense, on regions of the sample space $\mathcal{X}^m$ for which $\langle \theta, \phi(x) \rangle$ is large. Ultimately, as $\beta \to +\infty$, probability mass should remain only on configurations x^* in the set $\arg\max_{x \in \mathcal{X}^m} \langle \theta, \phi(x) \rangle$.

This intuition suggests that the behavior of the function $A(\beta\theta)$ should have a close connection to the problem of computing $\max_{x \in \mathcal{X}^m} \langle \theta, \phi(x) \rangle$. Since $A(\beta\theta)$ may diverge as $\beta \to +\infty$, it is most natural to consider the limiting behavior of the scaled quantity $A(\beta\theta)/\beta$. More formally, we state the following:

Theorem 8.1   For all $\theta \in \Omega$, the problem of mode computation has the following alternative representations:

$$\max_{x \in \mathcal{X}^m} \langle \theta, \phi(x) \rangle = \max_{\mu \in \bar{\mathcal{M}}} \langle \theta, \mu \rangle, \quad \text{and} \qquad (8.4a)$$

$$\max_{x \in \mathcal{X}^m} \langle \theta, \phi(x) \rangle = \lim_{\beta \to +\infty} \frac{A(\beta\theta)}{\beta}. \qquad (8.4b)$$

We provide the proof of this result in Appendix B.5; here we elaborate on its connections to the general variational principle (3.45), stated in Theorem 3.4. As a particular consequence of the general variational principle, we have

$$\lim_{\beta \to +\infty} \frac{A(\beta\theta)}{\beta} = \lim_{\beta \to +\infty} \frac{1}{\beta} \sup_{\mu \in \mathcal{M}} \left\{ \langle \beta\theta, \mu \rangle - A^*(\mu) \right\}$$

$$= \lim_{\beta \to +\infty} \sup_{\mu \in \mathcal{M}} \left\{ \langle \theta, \mu \rangle - \frac{1}{\beta} A^*(\mu) \right\}.$$

In this context, one implication of Theorem 8.1 is that the order of the limit over β and the supremum over μ can be exchanged. It is important to note that such an exchange is *not* correct in general; in this case, justification is provided by the convexity of A^* (see Appendix B.5 for details).

As with the variational principle of Theorem 3.4, only certain special cases of the variational problem $\max_{\mu \in \bar{\mathcal{M}}} \langle \theta, \mu \rangle$ can be solved exactly in a computationally efficient manner. The case of tree-structured Markov random fields on discrete random variables, discussed in Section 8.2, is one such case; another important exactly solvable case is the Gauss Markov random field, discussed in Section 8.3 and Appendix C.3. With reference to other (typically intractable) models, since the objective function itself is linear, the sole source of difficulty is the set $\mathcal{M}$, and in particular, the complexity of characterizing it with a small number of constraints. In the particular case of a discrete Markov random field associated with a graph G, the set $\mathcal{M}$ corresponds to the marginal polytope $\mathbb{M}(G)$, as discussed previously in Sections 3 and 4. For a discrete random vector, the mode-finding problem $\max_{x \in \mathcal{X}^m} \langle \theta, \phi(x) \rangle$ is an integer program (IP), since it involves searching over a finite discrete space. An implication of Theorem 8.1, when specialized to this setting, is that this IP is equivalent to a linear program over the marginal polytope. Since integer programming problems are NP-hard in general, this equivalence underscores the inherent complexity of marginal polytopes.

This type of transformation — namely, from an integer program to a linear program over the convex hull of its solutions — is a standard technique in integer programming and combinatorial optimization [e.g., 24, 103]. The field of polyhedral combinatorics

[69, 103, 180, 212] is devoted to understanding the structure of polytopes arising from various classes of discrete problems. As we describe in this section, the perspective of graphical models provides additional insight into the structure of these marginal polytopes.

8.2   Max-product and Linear Programming on Trees

As discussed in Section 4, an important class of exactly solvable models are Markov random fields defined on tree-structured graphs, say $T = (V, E)$. Our earlier discussion focused on the computation of mean parameters and the cumulant function, showing that the sum-product algorithm can be understood as an iterative algorithm for exactly solving the Bethe variational problem on trees [268, 269]. In this section, we discuss the parallel link between the ordinary max-product algorithm and linear programming [245].

For a discrete MRF on the tree, the set $\mathcal{M}$ is given by the marginal polytope $\mathbb{M}(T)$, whose elements consist of a marginal probability vector $\mu_s(\cdot)$ for each node, and joint probability matrix $\mu_{st}(\cdot, \cdot)$ for each edge $(s, t) \in E$. Recall from Proposition 4.1 that for a tree, the polytope $\mathbb{M}(T)$ is equivalent to the constraint set $\mathbb{L}(T)$, and so has a compact description in terms of nonnegativity, local normalization, and edge marginalization conditions. Using this fact, we now show how the mode-finding problem for a tree-structured problem can be reformulated as a simple linear program (LP) over the set $\mathbb{L}(G)$.

Using $\mathbb{E}_{\mu_s}[\theta_s(x_s)] := \sum_{x_s} \mu_s(x_s)\theta_s(x_s)$ to expectation under the marginal distribution μ_s and with a similar definition for $\mathbb{E}_{\mu_{st}}$, let us define the cost function

$$\langle \mu, \theta \rangle := \sum_{s \in V} \mathbb{E}_{\mu_s}[\theta_s(x_s)] + \sum_{(s,t) \in E} \mathbb{E}_{\mu_{st}}[\theta_{st}(x_s, x_t)].$$

With this notation, Theorem 8.1 implies that the MAP problem for a tree is equivalent to

$$\max_{x \in \mathcal{X}^m} \left[\sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t) \right] = \max_{\mu \in \mathbb{L}(T)} \langle \mu, \theta \rangle. \qquad (8.5)$$

The left-hand side is the standard representation of the MAP problem as an integer program, whereas the right-hand side — an optimization

problem over the polytope $\mathbb{L}(T)$ with a linear objective function — is a linear program.

We now have two links: first, the equivalence between the original mode-finding integer program and the LP (8.5), and second, via Theorem 8.1, a connection between this linear program and the zero-temperature limit of the Bethe variational principle. These links raise the intriguing possibility: *is there a precise connection between the max-product algorithm and the linear program* (8.5)*?* The max-product algorithm — like its relative the sum-product algorithm — is a distributed graph-based algorithm, which operates by passing a message M_{ts} along each direction $t \to s$ of each edge of the tree. For discrete random variables $X_s \in \{0, 1, \ldots, r-1\}$, each message is a r-vector, and the messages are updated according to recursion

$$M_{ts}(x_s) \; \leftarrow \; \kappa \max_{x_t \in \mathcal{X}_t} \left[\exp\left\{ \theta_{st}(x_s, x_t) + \theta_t(x_t) \right\} \prod_{u \in N(t) \backslash s} M_{ut}(x_t) \right]. \tag{8.6}$$

The following result [245] gives an affirmative answer to the question above: for tree-structured graphs, the max-product updates (8.6) are a Lagrangian method for solving the dual of the linear program (8.5). This result can be viewed as the max-product analog to the connection between sum-product and the Bethe variational problem on trees, and its exactness for computing marginals, as summarized in Theorem 4.2.

To set up the Lagrangian, for each $x_s \in \mathcal{X}_s$, let $\lambda_{st}(x_s)$ be a Lagrange multiplier associated with the marginalization constraint $C_{ts}(x_s) = 0$, where

$$C_{ts}(x_s) := \mu_s(x_s) - \sum_{x_t} \mu_{st}(x_s, x_t). \tag{8.7}$$

Let $\mathbb{N} \subset \mathbb{R}^d$ be the set of μ that are nonnegative and appropriately normalized:

$$\mathbb{N} := \left\{ \mu \in \mathbb{R}^d \mid \mu \geq 0, \ \sum_{x_s} \mu_s(x_s) = 1, \ \sum_{x_s, x_t} \mu_{st}(x_s, x_t) = 1 \right\}. \tag{8.8}$$

With this notation, we have the following result:

Proposition 8.2 (Max-product and LP Duality). Consider the dual function $\mathcal{Q}$ defined by the following partial Lagrangian formulation of the tree-structured LP (8.5):

$$\mathcal{Q}(\lambda) := \max_{\mu \in \mathbb{N}} \mathcal{L}(\mu; \lambda), \quad \text{where}$$

$$\mathcal{L}(\mu; \lambda) := \langle \theta, \mu \rangle + \sum_{(s,t) \in E(T)} \left[\sum_{x_s} \lambda_{ts}(x_s) C_{ts}(x_s) + \sum_{x_t} \lambda_{st}(x_t) C_{st}(x_t) \right].$$

For any fixed point M^* of the max-product updates (8.6), the vector $\lambda^* := \log M^*$, where the logarithm is taken elementwise, is an optimal solution of the dual problem $\min_\lambda \mathcal{Q}(\lambda)$.

Proof. We begin by deriving a convenient representation of the dual function $\mathcal{Q}$. First, let us convert the tree to a directed version by first designating some node, say node 1 without loss of generality, as the root, and then directing all the edges from parent to child $t \to s$. With regard to this rooted tree, the objective function $\langle \theta, \mu \rangle$ has the alternative decomposition:

$$\sum_{x_1} \mu_1(x_1) \theta_1(x_1) + \sum_{t \to s} \sum_{x_t, x_s} \mu_{st}(x_s, x_t) \big[\theta_{st}(x_s, x_t) + \theta_s(x_s) \big].$$

With this form of the cost function, the dual function can be put into the form

$$\mathcal{Q}(\lambda) := \max_{\mu \in \mathbb{N}} \left\{ \sum_{x_1} \mu_1(x_1) \nu_1(x_1) \right.$$

$$\left. + \sum_{t \to s} \sum_{x_t, x_s} \mu_{st}(x_s, x_t) \big[\nu_{st}(x_s, x_t) - \nu_t(x_t) \big] \right\}, \qquad (8.9)$$

where the quantities ν_s and ν_{st} are defined in terms of λ and θ as:

$$\nu_t(x_t) = \theta_t(x_t) + \sum_{u \in N(t)} \lambda_{ut}(x_t), \quad \text{and} \qquad (8.10a)$$

$$\nu_{st}(x_s, x_t) = \theta_{st}(x_s, x_t) + \theta_s(x_s) + \theta_t(x_t)$$
$$+ \sum_{u \in N(s) \backslash t} \lambda_{us}(x_s) + \sum_{u \in N(t) \backslash s} \lambda_{ut}(x_t). \qquad (8.10b)$$

Taking the maximum over $\mu \in \mathbb{N}$ in Equation (8.9) yields that the dual function has the explicit form:

$$\mathcal{Q}(\lambda) = \max_{x_1} \left[\nu_1(x_1) \right] + \sum_{t \to s} \max_{x_s, x_t} \left[\nu_{st}(x_s, x_t) - \nu_t(x_t) \right]. \qquad (8.11)$$

Using the representation (8.11), we now have the necessary ingredients to make the connection to the max-product algorithm. Given a vector of messages M in the max-product algorithm, we use it to define a vector of Lagrange multipliers via $\lambda = \log M$, where the logarithm is taken elementwise. With a bit of algebra, it can be seen that a message vector M^* is a fixed point of the max-product updates (8.6) if and only if the associated ν_s^* and ν_{st}^*, as defined by $\lambda^* := \log M^*$, satisfy the *edgewise consistency* condition

$$\max_{x_s} \nu_{st}^*(x_s, x_t) = \nu_t^*(x_t) + \kappa_{st} \qquad (8.12)$$

for all $x_t \in \mathcal{X}_t$, where κ_{st} is a constant independent of x. We now show that any such λ^* is a dual optimal solution.

We first claim that under the edgewise consistency condition on a tree-structured graph, we can always find at least one configuration $x^* = (x_1^*, x_2^*, \ldots, x_m^*)$ that satisfies the conditions:

$$x_s^* \in \arg\max_{x_s} \nu_s^*(x_s) \qquad \forall\, s \in V, \quad \text{and} \qquad (8.13\text{a})$$

$$(x_s^*, x_t^*) \in \arg\max_{x_s, x_t} \nu_{st}^*(x_s, x_t) \qquad \forall\, (s,t) \in E. \qquad (8.13\text{b})$$

Indeed, such a configuration can be constructed recursively as follows. First, for the root node 1, choose x_1^* to achieve the maximum $\max_{x_1} \nu_1^*(x_1)$. Second, for each child $t \in N(1)$, choose x_t to maximize $\nu_{t,1}(x_t, x_1^*)$, and then iterate the process. The edgewise consistency (8.12) guarantees that any configuration $(x_1^*, x_2^*, \ldots, x_m^*)$ constructed in this way satisfies the conditions (8.13a) and (8.13b).

The edgewise consistency condition (8.12) also guarantees the following equalities:

$$\max_{x_s, x_t} [\nu_{st}^*(x_s, x_t) - \nu_t^*(x_t)] = \max_{x_t} [\nu_t^*(x_t) + \kappa_{st} - \nu_t^*(x_t)]$$
$$= \kappa_{st}$$
$$= \nu_{st}^*(x_s^*, x_t^*) - \nu_t^*(x_t^*).$$

Combining these relations yields the following expression for the dual value at λ^*:

$$\mathcal{Q}(\lambda^*) = \nu_1^*(x_1^*) + \sum_{t \to s} \big[\nu_{st}^*(x_s^*, x_t^*) - \nu_t^*(x_t^*) \big].$$

Next, applying the definition (8.10) of ν^* and simplifying, we find that

$$\mathcal{Q}(\lambda^*) = \theta_1(x_1^*) + \sum_{t \to s} \big[\theta_{st}(x_s^*, x_t^*) + \theta_s(x_s^*) \big].$$

Now consider the primal solution defined by $\mu_s^*(x_s) := \mathbb{I}_{x_s^*}[x_s]$ and $\mu_{st}^*(x_s, x_t) = \mathbb{I}_{x_s^*}[x_s]\,\mathbb{I}_{x_t^*}[x_t]$, where $\mathbb{I}_{x_s^*}[x_s]$ is an indicator function for the event $\{x_s = x_s^*\}$. It is clear that μ^* is primal feasible; moreover, the primal cost is equal to

$$\sum_{x_1} \mu_1^*(x_1)\theta_1(x_1) + \sum_{t \to s} \sum_{x_t, x_s} \mu_{st}^*(x_s, x_t)\big[\theta_{st}(x_s, x_t) + \theta_s(x_s)\big]$$

$$= \theta_r(x_r^*) + \sum_{t \to s} \big[\theta_{st}(x_s^*, x_t^*) + \theta_s(x_s^*)\big],$$

which is precisely equal to $\mathcal{Q}(\lambda^*)$. Therefore, by strong duality for linear programs [24], the pair (μ^*, λ^*) is primal–dual optimal. $\qquad\square$

Remark 8.1 A careful examination of the proof of Proposition 8.2 shows that several steps rely heavily on the fact that the underlying graph is a tree. In fact, the corresponding result for a graph with cycles *fails* to hold, as we discuss at length in Section 8.4.

8.3 Max-product For Gaussians and Other Convex Problems

Multivariate Gaussians are another special class of Markov random fields for which there are various guarantees of correctness associated with the max-product (or min-sum) algorithm. Given an undirected graph $G = (V, E)$, consider a Gaussian MRF in exponential form:

$$p_{(\theta, \Theta)}(x) = \exp\big\{ \langle \theta, \, x \rangle + \langle\!\langle \Theta, \, xx^T \rangle\!\rangle - A(\theta, \Theta) \big\}$$

$$= \exp\Big\{ \sum_{s \in V} \theta_s x_s + \sum_{s,t} \Theta_{st} x_s x_t - A(\theta, \Theta) \Big\}. \quad (8.14)$$

Recall that the graph structure is reflected in the sparsity of the matrix Θ, with $\Theta_{uv} = 0$ whenever $(u,v) \notin E$.

In Appendix C.3, we describe how taking the zero-temperature limit for a multivariate Gaussian, according to Theorem 8.1, leads to either a quadratic program, or a semidefinite program. There are various ways in which these convex programs could be solved; here we discuss known results for the application of the max-product algorithm.

To describe the max-product as applied to a multivariate Gaussian, we first need to convert the model into pairwise form. Let us define potential functions of the form

$$\gamma_s(x_s) = \theta_s x_s + \theta_s x_s^2, \quad \forall\, s \in V, \quad \text{and} \tag{8.15a}$$

$$\gamma_{st}(x_s, x_t) = \gamma_{ss;t}\, x_s^2 + 2\Theta_{st} x_s x_t + \gamma_{tt;s}\, x_t^2 \quad \forall\, (s,t) \in E, \tag{8.15b}$$

where the free parameters $\{\gamma_s, \gamma_{ss;t}\}$ must satisfy the constraint

$$\gamma_s + \sum_{t \in N(s)} \gamma_{ss;t} = \Theta_{ss}.$$

Under this condition, the multivariate Gaussian density (8.14) can equivalently expressed as the pairwise Markov random field

$$p_\gamma(x) \propto \exp\left\{ \sum_{s \in V} \gamma_s(x_s) + \sum_{(s,t) \in E} \gamma_{st}(x_s, x_t) \right\}. \tag{8.16}$$

With this set-up, we can now describe the max-product message-passing updates. For graphical models involving continuous random variables, each message $M_{ts}(x_s)$ is a real-valued function of the indeterminate x_s. These functional messages are updated according to the usual max-product recursion (8.6), with the potentials θ_s and θ_{st} in Equation (8.6) replaced by their γ_s and γ_{st} analogs from the pairwise factorization (8.16). In general continuous models, it is computationally challenging to represent and compute with these functions, as they are infinite-dimensional quantities. An attractive feature of the Gaussian problems these messages can be compactly parameterized; in particular, assuming a scalar Gaussian random variable X_s at each node, any message must be an exponentiated-quadratic function of its argument, of the form $M_{ts}(x_s) \propto \exp(ax_s + bx_s^2)$. Consequently,

max-product message-passing updates (8.6) can be efficiently implemented with one recursion for the mean term (number a), and a second recursion for the variance component (see the papers [250, 262] for further details).

For Gaussian max-product applied to a tree-structured problem, the updates are guaranteed to converge, and compute both the correct means $\mu_s = \mathbb{E}[X_s]$ and variances $\sigma_s^2 = \mathbb{E}[X_s^2] - \mu_s^2$ at each node [262]. For general graphs with cycles, the max-product algorithm is no longer guaranteed to converge. However, Weiss and Freeman [250] and independently Rusmevichientong and van Roy [205] showed that if Gaussian max-product (or equivalently, Gaussian sum-product) converges, then the fixed point specifies the correct Gaussian means μ_s, but the estimates of the node variances σ_s^2 need not be correct. The correctness of Gaussian max-product for mean computation also follows as a consequence of the reparameterization properties of the sum- and max-product algorithms [242, 244]. Weiss and Freeman [250] also showed that Gaussian max-product converges for arbitrary graphs if the precision matrix ($-\Theta$ in our notation) satisfies a certain diagonal dominance condition. This sufficient condition for convergence was substantially tightened in later work by Malioutov et al. [161], using the notions of walk-summability and pairwise normalizability; see also the paper [176] for further refinements. Moallemi and van Roy [177] consider the more general problem of maximizing an arbitrary function of the form (8.16), where the potentials $\{\gamma_s, \gamma_{st}\}$ are required only to define a convex function. The quadratic functions (8.15) for the multivariate Gaussian are a special case of this general set-up. For this family of pairwise separable convex programs, Moallemi and van Roy [177] established convergence of the max-product updates under a certain scaled diagonal dominance condition.

8.4   First-order LP Relaxation and Reweighted Max-product

In this section, we return to Markov random fields with discrete random variables, for which the mode-finding problem is an integer program, as opposed to the quadratic program that arises in the Gaussian case.

For a general graph with cycles, this discrete mode-finding problem is known to be computationally intractable, since it includes as special cases many problems known to be NP-complete, among them MAX-CUT and related satisfiability problems [131]. Given the computational difficulties associated with exact solutions, it is appropriate to consider approximate algorithms.

An important class of approximate methods for solving integer and combinatorial optimization problems is based on *linear programming relaxation*. The basic idea is to approximate the convex hull of the solution set by a set of linear constraints, and solve the resulting linear program. If the obtained solution is integral, then the LP relaxation is tight, whereas in other cases, the obtained solution may be fractional, corresponding to looseness of the relaxation. Frequently, LP relaxations are developed in a manner tailored to specific subclasses of combinatorial problems [180, 236].

In this section, we discuss the linear programming (LP) relaxation that emerges from the Bethe variational principle, or its zero-temperature limit. This LP relaxation, when specialized to particular combinatorial problems — among them node cover, independent set, matching, and satisfiability problems — recovers various classical LP relaxations as special cases. Finally, we discuss connections between this LP relaxation and max-product message-passing. For general MRFs, max-product itself is *not* a method for solving this LP, as we demonstrate with a concrete counterexample [245]. However, it turns out that suitably reweighted versions of max-product are intimately linked to this LP relaxation, which provides an entry point to an ongoing line of research on LP relaxations and message-passing algorithms on graphs.

8.4.1 Basic Properties of First-order LP Relaxation

For a general graph, the set $\mathbb{L}(G)$ no longer provides an exact characterization of the marginal polytope $\mathbb{M}(G)$, but is always guaranteed to outer bound it. Consequently, the following inequality is valid for any graph:

$$\max_{x \in \mathcal{X}^m} \langle \theta, \phi(x) \rangle = \max_{\mu \in \mathbb{M}(G)} \langle \theta, \mu \rangle \leq \max_{\tau \in \mathbb{L}(G)} \langle \theta, \tau \rangle. \qquad (8.17)$$

Since the relaxed constraint set $\mathbb{L}(G)$ is a polytope, the right-hand side of Equation (8.17) is a linear program, which we refer to as the *first-order LP relaxation*.[1] Most importantly, the right-hand side is a linear program, where the number of constraints defining $\mathbb{L}(G)$ grows only linearly in the graph size, so that it can be solved in polynomial time for any graph [211].

Special cases of this LP relaxation (8.17) have studied in past and ongoing work by various authors, including the special cases of $\{0,1\}$-quadratic programs [105], metric labeling with Potts models [135, 48], error-control coding problems [76, 78, 239, 228, 50, 62], independent set problems [180, 207], and various types of matching problems [15, 116, 206]. We begin by discussing some generic properties of the LP relaxation, before discussing some of these particular examples in more detail.

By standard properties of linear programs [211], the optimal value of the relaxed LP must be attained by at least one an extreme point of the polytope $\mathbb{L}(G)$. (A element of a convex set is an extreme point if it cannot be expressed as a convex combination of two distinct elements of the set; see Appendix A.2.4 for more background). We say that an extreme point of $\mathbb{L}(G)$ is *integral* if all of its components are zero or one, and *fractional* otherwise. By its definition, any extreme point $\mathbb{M}(G)$ is of the form $\mu_y := \phi(y)$, where $y \in \mathcal{X}^m$ is a fixed configuration. Recall that in the canonical overcomplete representation (3.34), the sufficient statistic vector ϕ consists of $\{0,1\}$-valued indicator functions, so that $\mu_y = \phi(y)$ is a vector with $\{0,1\}$ components. The following result specifies the relation between extreme points of $\mathbb{M}(G)$ and those of $\mathbb{L}(G)$:

Proposition 8.3 The extreme points of $\mathbb{L}(G)$ and $\mathbb{M}(G)$ are related as follows:

(a) All the extreme points of $\mathbb{M}(G)$ are integral, and each one is also an extreme point of $\mathbb{L}(G)$.

---

[1] The term "first-order" refers to its status as the first in a natural hierarchy of relaxations, based on the treewidth of the underlying graph, as discussed at more length in Section 8.5.

(b) For any graph with cycles, $\mathbb{L}(G)$ also includes additional extreme points with fractional elements that lie strictly outside $\mathbb{M}(G)$.

Proof. (a) Any extreme point of $\mathbb{M}(G)$ is of the form $\phi(y)$, for some configuration $y \in \mathcal{X}^m$. Each of these extreme points has 0–1 components, and so is integral. Since $\mathbb{L}(G)$ is a polytope in

$$d = \sum_{s \in V} r_s + \sum_{(s,t) \in E} r_s r_t$$

dimensions, in order to show that $\phi(y)$ is also an extreme point of $\mathbb{L}(G)$, it is equivalent [24] to show that there are d constraints of $\mathbb{L}(G)$ that are active at $\phi(y)$ and are also linearly independent. For any $y \in \mathcal{X}^m$, we have $\mathbb{I}_k(x_s) = 0$ for all $k \in \mathcal{X}_s \backslash \{y_s\}$, and $\mathbb{I}_k(x_s)\mathbb{I}_l(x_t) = 0$ for all $(k,l) \in (\mathcal{X}_s \times \mathcal{X}_t) \backslash \{y_s, y_t\}$. All of these active inequality constraints are linearly independent, and there are a total of

$$d' = \sum_{s \in V}(r_s - 1) + \sum_{(s,t) \in E}(r_s r_t - 1) = d - m - |E|$$

such constraints. All of the normalization and marginalization constraints are also satisfied by the vector $\mu_y = \phi(y)$, but not all of them are linearly independent (when added to the active inequality constraints). However, we can add a normalization constraint for each vertex $s = 1, \ldots, m$, as well as a normalization constraint for each edge $(s,t) \in E$, while still preserving linear independence. Adding these $m + |E|$ equality constraints to the d' inequality constraints yields a total of d linearly independent constraints of $\mathbb{L}(G)$ that are satisfied by μ_y, which implies that it is an extreme point [24].

(b) Example 8.1 provides a constructive procedure for constructing fractional extreme points of the polytope $\mathbb{L}(G)$ for any graph G with cycles. $\qquad\square$

The distinction between fractional and integral extreme points is crucial, because it determines whether or not the LP relaxation (8.17) specified by $\mathbb{L}(G)$ is tight. In particular, there are only two possible

outcomes to solving the relaxation:

- (a) The optimum is attained at an extreme point of $\mathbb{M}(G)$, in which case the upper bound in Equation (8.17) is tight, and a mode can be obtained.
- (b) The optimum is attained only at one or more fractional extreme points of $\mathbb{L}(G)$, which lie strictly outside $\mathbb{M}(G)$. In this case, the upper bound of Equation (8.17) is loose, and the optimal solution to the LP relaxation does not specify the optimal configuration. In this case, one can imagine various types of rounding procedures for producing near-optimal solutions [236].

When the graph has cycles, it is possible to explicitly construct a fractional extreme point of the relaxed polytope $\mathbb{L}(G)$.

Example 8.1 (Fractional Extreme Points of $\mathbb{L}(G)$). As an illustration, let us explicitly construct a fractional extreme point for the simplest problem on which the first-order relaxation (8.17) is not always exact: a binary random vector $X \in \{0,1\}^3$ following an Ising model on the complete graph K_3. Consider the canonical parameter θ shown in matrix form in Figure 8.1(a). When $\beta_{st} < 0$, then configurations with $x_s \neq x_t$ are favored, so that the interaction is repulsive. In contrast, when $\beta_{st} > 0$, the interaction is attractive, because it favors configurations with $x_s = x_t$. When $\beta_{st} > 0$ for all $(s,t) \in E$, it can be shown [138] that the first-order LP relaxation (8.17) is tight, for any choice of the single node parameters $\{\theta_s, s \in V\}$. In contrast, when $\beta_{st} < 0$ for all edges, then there are choices of $\theta_s, s \in V$ for which the relaxation breaks down.

The following canonical parameter corresponds to a direction for which the relaxation (8.17) is *not* tight, and hence exposes a fractional extreme point. First, choose $\theta_s = (0, 0)$ for $s = 1,2,3$, and then set $\beta_{st} = \beta < 0$ for all edges (s,t), and use these values to define the pairwise potentials θ_{st} via the construction in Figure 8.1(a). Observe that for any configuration $x \in \{0,1\}^3$, we must have $x_s = x_t$ for at least one edge $(s,t) \in E$. Therefore, any $\mu \in \mathbb{M}(G)$ must place nonzero mass

$$\theta_s \;=\; \begin{bmatrix} \theta_{s;0} & \theta_{s;1} \end{bmatrix}$$

$$\theta_{st} \;=\; \begin{bmatrix} \beta_{st} & 0 \\ 0 & \beta_{st} \end{bmatrix}$$

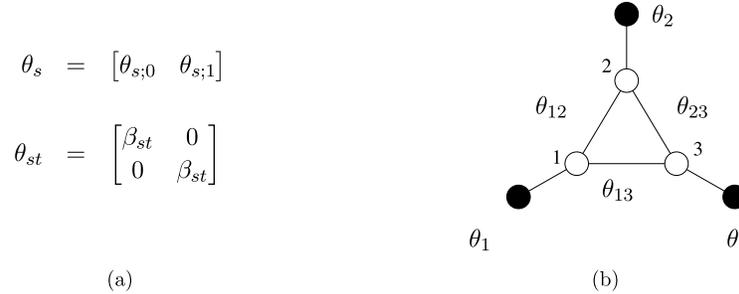(a)   (b)

Fig. 8.1 The smallest graph $G = (V, E)$ on which the relaxation (8.17) can fail to be tight. For $\beta_{st} \geq 0$ for all $(s,t) \in E$, the relaxation is tight for any choice of $\theta_s, s \in V$. On the other hand, if $\beta_{st} < 0$ for all edges (s,t), the relaxation will fail for certain choices of $\theta_s, s \in V$.

on at least one term of θ involving β, whence $\max_{\mu \in \mathbb{M}(G)} \langle \theta, \mu \rangle < 0$. In fact, this optimal value is exactly equal to $\beta < 0$. On the other hand, consider the pseudomarginal $\tau^* \in \mathbb{L}(G)$ formed by the singleton and pairwise pseudomarginals defined as follows:

$$\tau_s^* := \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}^T \quad \text{for } s \in V, \quad \text{and}$$

$$\tau_{st}^* = \begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix} \quad \text{for } (s,t) \in E.$$

Observe that $\langle \theta, \tau^* \rangle = 0$. Since $\theta_\alpha \leq 0$ for all elements α, this value is the optimum of $\langle \theta, \tau \rangle$ over $\mathbb{L}(G)$, thereby showing that the relaxation (8.17) is *not* tight.

Finally, to establish that τ^* is an extreme point of the polytope $\mathbb{L}(G)$, it is sufficient [24] to show that $\langle \theta, \tau \rangle < 0$ for all $\tau \neq \tau^*$. If $\langle \theta, \tau \rangle = 0$, then for all $(s,t) \in E$ the pairwise pseudomarginals must be of the form:

$$\tau_{st} := \begin{bmatrix} 0 & \alpha_{st} \\ 1 - \alpha_{st} & 0 \end{bmatrix}$$

for some $\alpha_{st} \in [0,1]$. Enforcing the marginalization constraints on these pairwise pseudomarginals yields the constraints $\alpha_{12} = \alpha_{13} = \alpha_{23}$ and $1 - \alpha_{12} = \alpha_{23}$, whence $\alpha_{st} = 0.5$ is the only possibility. Therefore, we conclude that the pseudomarginal vector τ^* is a fractional extreme point of the polytope $\mathbb{L}(G)$.

Given the possibility of fractional extreme points in the first-order LP relaxation (8.5), it is natural to ask the question: do fractional solutions yield partial information about the set of optimal solutions to the original integer program? One way in which to formalize this question is through the notion of persistence [105]. In particular, letting $\mathcal{O}^* := \arg\max_{x \in \mathcal{X}^m} \langle \theta, \phi(x) \rangle$ denote the set of optima to an integer program, we have:

Definition 8.1   Given a fractional solution τ to the LP relaxation (8.5), let $I \subset V$ represent the subset of vertices for which τ_s has only integral elements, say fixing $x_s = x_s^*$ for all $s \in I$. The fractional solution is *strongly persistent* if any optimal integral solution $y^* \in \mathcal{O}^*$ satisfies $y_s^* = x_s^*$ for all $s \in I$. The fractional solution is *weakly persistent* if there exists some $y^* \in \mathcal{O}^*$ such that $y_s^* = x_s^*$ for all $s \in I$.

Thus, any persistent fractional solution (whether weak or strong) can be used to fix a subset of elements in the integer program, while still being assured that there exists an optimal integral solution that is consistent. Strong persistency ensures that no candidate solutions are eliminated by the fixing procedure. Hammer et al. [105] studied the roof-dual relaxation for binary quadratic programs, an LP relaxation which is equivalent to specializing the first-order LP to binary variables, and proved the following result:

Proposition 8.4   Suppose that the first-order LP relaxation (8.5) is applied to the binary quadratic program

$$\max_{x \in \{0,1\}^m} \left\{ \sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in E} \theta_{st} x_s x_t \right\}. \tag{8.18}$$

Then any fractional solution is strongly persistent.

As will be discussed at more length in Example 8.4, the class of binary QPs includes as special cases various classical problems from the combinatorics literature, among them MAX-2SAT, independent set, MAX-CUT, and vertex cover. For all of these problems, then,

the first-order LP relaxation is strongly persistent. Unfortunately, this strong persistency fails to extend to the first-order relaxation (8.5) with nonbinary variables; see the discussion following Example 8.3 for a counterexample.

8.4.2 Connection to Max-Product Message-Passing

In analogy to the general connection between the Bethe variational problem and the sum-product algorithm (see Section 4), one might postulate that Proposition 8.2 could be extended to graphs with cycles — specifically, that the max-product algorithm solves the dual of the tree-based relaxation (8.17). In general, this conjecture is false, as shown by the following counterexample [245].

Example 8.2 (Max-product does not Solve the LP). Consider the diamond graph G_{dia} shown in Figure 8.2, and suppose that we wish to maximize a cost function of the form:

$$\alpha(x_1 + x_4) + \beta(x_2 + x_3) + \gamma \sum_{(s,t) \in E} \mathbb{I}[x_s \neq x_t]. \qquad (8.19)$$

Here the maximization is over all binary vectors $x \in \{0,1\}^4$, and α, β, and γ are parameters to be specified. By design, the cost function (8.19) is such that if we make γ sufficiently negative, then any optimal solution will either be $0^4 := (0,0,0,0)$ or $1^4 := (1,1,1,1)$. As an extreme example, if we set $\alpha = 0.31$, $\beta = -0.30$, and $\gamma = -\infty$, we see immediately that the optimal solution is 1^4. (Note that setting $\gamma = -\infty$ is equivalent to imposing the "hard-core" constraint that $x_s = x_t$ for all $(s,t) \in E$.)

A classical way of studying the ordinary sum- and max-product algorithms, dating back to the work of Gallager [87] and Wiberg et al. [258], is via the computation tree associated with the message-passing updates. As illustrated in Figure 8.2(b), the computation tree is rooted at a particular vertex (1 in this case), and it tracks the paths of messages that reach this root node. In general, the $(n+1)$th level of tree includes all vertices t such that a path of length n joins t to the root. For the particular example shown in Figure 8.2(b), in the
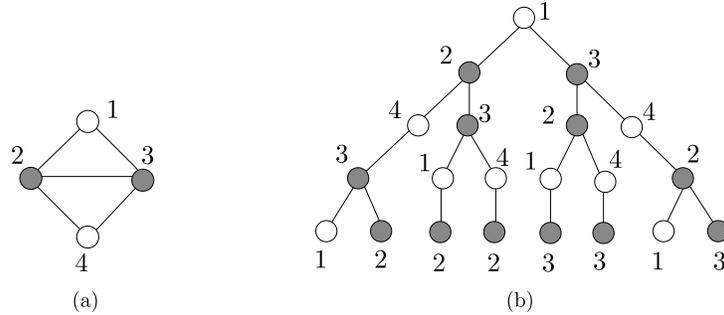
Fig. 8.2 (a) Simple diamond graph G_{dia}. (b) Associated computation tree after four rounds of message-passing. Max-product solves exactly the modified integer program defined by the computation tree.

first iteration represented by the second level of the tree in panel (b), the root node 1 receives messages from nodes 2 and 3, and at the second iteration represented by the third level of the tree, it receives one message from node 3 (relayed by node 2), one message from node 2 (relayed by node 3), and two messages from node 4, one via node 2 and the other via node 3, and so on down the tree.

The significance of this computation tree is based on the following observation: by definition, the decision of the max-product algorithm at the root node 1 after n iterations is optimal with respect to the *modified integer program* defined by the computation tree with $n + 1$ levels. That is, the max-product decision will be $\widehat{x}_1 = 1$ if and only if the optimal configuration in the tree with $x_1 = 1$ has higher probability than the optimal configuration with $x_1 = 0$. For the diamond graph under consideration and given the "hard-core" constraints imposed by setting $\gamma = -\infty$, the only two possible configurations in any computation tree are all-zeros, or all-ones. Thus, the max-product reduces to comparing the total weight on the computation tree associated with all-ones to that associated with all-zeros.

However, the computation tree in Figure 8.2(b) has a curious property: due to the inhomogeneous node degrees — more specifically, with nodes 2 and 3 having three neighbors, and 1 and 4 having only two neighbors — nodes 2 and 3 receive a disproportionate representation in the computation tree. This fact is clear by inspection, and can be

verified rigorously by setting up a simple recursion to compute the appearance fractions of nodes 2 and 3 versus nodes 1 and 4. Doing so shows that nodes 2 and 3 appear roughly $\rho \approx 1.0893$ more frequently than nodes 1 and 4. As a consequence, the ordinary max-product algorithm makes its decision according to the threshold rule

$$\widehat{x}_{\mathrm{MP}} = \begin{cases} 1^4 & \text{if } \alpha + \rho\beta > 0 \\ 0^4 & \text{otherwise,} \end{cases} \tag{8.20}$$

whereas the correct decision rule is based on thresholding $\alpha + \beta$. Consequently, for any $(\alpha, \beta) \in \mathbb{R}^2$ such that $\alpha + \beta > 0$ but $\alpha + \rho\beta < 0$, the max-product algorithm outputs an incorrect configuration. For instance, setting $\alpha = 0.31$ and $\beta = -0.30$ yields one such counterexample, as discussed in Wainwright et al. [245]. Kulesza and Pereira [144] provide a detailed analysis of the max-product message-passing updates for this example, analytically deriving the updates and explicitly demonstrating convergence to incorrect configurations.

Thus far, we have demonstrated a simple problem for which the max-product algorithm fails. What is the connection to the LP relaxation (8.17)? It turns out that the problem instance that we have constructed is an instance of a supermodular maximization problem. As discussed at more length in Example 8.4 to follow. it can be shown [138] that the first-order LP relaxation (8.17) is tight for maximizing any binary quadratic cost function with supermodular interactions. The cost function (8.19) is supermodular for all weights $\gamma \leq 0$, so that in particular, the first-order LP relaxation is tight for the cost function specified by $(\alpha, \beta, \gamma) = (0.31, -0.30, -\infty)$. However, as we have just shown, ordinary max-product fails for this problem, so it is not solving the LP relaxation.

As discussed at more length in Section 8.4.4, for some problems with special combinatorial structure, the max-product algorithm does solve the first-order LP relaxation (8.5). These instances include the problem of bipartite matching and weighted b-matching. However, establishing a general connection between message-passing and the first-order LP relaxation (8.5) requires developing related but different message-passing algorithms, the topic to which we now turn.

8.4.3   Reweighted Max-product and Other Modified Message-passing Schemes

In this section, we begin by presenting the tree-reweighted max-product updates [245], and describing their connection to the first-order LP relaxation (8.5). Recall that in Theorem 7.2, we established a connection between the tree-reweighted Bethe variational problem (7.11), and the tree-reweighted sum-product updates (7.12). Note that the constraint set in the tree-reweighted Bethe variational problem is exactly the polytope $\mathbb{L}(G)$ that defines the first-order LP relaxation (8.5). This fact suggests that there should be a connection between the "zero-temperature limit" of the tree-reweighted Bethe variational problem and the first-order LP relaxation. In particular, recalling the convex surrogate $B(\theta) = B_{\mathfrak{T}}(\theta; \rho_e)$ defined by the tree-reweighted Bethe variational problem from Theorem 7.2, let us consider the limit $B(\beta\theta)/\beta$ as $\beta \to +\infty$. From the variational definition (7.11), we have

$$\lim_{\beta \to +\infty} \frac{B(\beta\theta)}{\beta} = \lim_{\beta \to +\infty} \left[\sup_{\tau \in \mathbb{L}(G)} \left\{ \langle \theta, \tau \rangle - \frac{1}{\beta} B^*(\tau) \right\} \right].$$

As discussed previously, convexity allows us to exchange the order of the limit and supremum, so that we conclude that the zero-temperature limit of the convex surrogate B is simply the first-order LP relaxation (8.5).

Based on this intuition, it is natural to suspect that the *tree-reweighted max-product* algorithm should have a general connection to the first-order LP relaxation. In analogy to the TRW-sum-product updates (7.12), the reweighted max-product updates take the form:

$$M_{ts}(x_s) \leftarrow \kappa \max_{x'_t \in \mathcal{X}_t} \left\{ \exp\left(\frac{1}{\rho_{st}} \theta_{st}(x_s, x'_t) + \theta_t(x'_t) \right) \right.$$
$$\left. \times \frac{\displaystyle\prod_{v \in N(t) \setminus s} \left[M_{vt}(x'_t) \right]^{\rho_{vt}}}{\left[M_{st}(x'_t) \right]^{(1-\rho_{ts})}} \right\}. \qquad (8.21)$$

where $(\rho_{st}, (s,t) \in E)$ is a vector of positive edge weights in the spanning tree polytope.[2]

As with the reweighted sum-product updates, these messages define a collection of *pseudo-max-marginals* of the form

$$\nu_s(x_s) \propto \exp(\theta_s(x_s)) \prod_{t \in N(s)} [M_{ts}(x_s)]^{\rho_{ts}}, \qquad \text{and} \qquad (8.22a)$$

$$\nu_s(x_s, x_t) \propto \exp(\gamma_{st}(x_s, x_t))$$
$$\times \frac{\displaystyle\prod_{u \in N(s)\backslash t} [M_{us}(x_s)]^{\rho_{us}}}{[M_{ts}(x_s)]^{1-\rho_{st}}} \frac{\displaystyle\prod_{u \in N(t)\backslash s} [M_{ut}(x_t)]^{\rho_{ut}}}{[M_{st}(x_t)]^{1-\rho_{st}}}, \qquad (8.22b)$$

where $\gamma_{st}(x_s, x_t) := \theta_s(x_s) + \theta_t(x_t) + \frac{\theta_{st}(x_s, x_t)}{\rho_{st}}$.

Pseudo-max-marginals that satisfy certain conditions can be used to specify an optimal configuration $\widehat{x}_{\mathrm{TRW}}$. As detailed in the paper [245], the most general sufficient condition is that there exists a configuration $\widehat{x} = \widehat{x}_{\mathrm{TRW}}$ that is nodewise and edgewise optimal across the entire graph, meaning that

$$\widehat{x}_s \in \arg\max_{x_s} \nu_s(x_s) \quad \forall\, s \in V, \quad \text{and} \qquad (8.23a)$$

$$(\widehat{x}_s, \widehat{x}_t) \in \arg\max_{x_s, x_t} \nu_{st}(x_s, x_t) \quad \forall\, (s,t) \in E. \qquad (8.23b)$$

In this case, we say that the pseudo-max-marginals ν satisfy the *strong tree agreement condition* (STA).

Under this condition, Wainwright et al. [245] showed the following:

Proposition 8.5 Given a weight vector ρ_e in the spanning tree polytope, any STA fixed point (M^*, ν^*) of TRW-max-product specifies an optimal dual solution for the first-order tree LP relaxation (8.5).

Fixed points ν^* satisfying conditions (8.23) are the most practically relevant, since in this case, the fixed point can be used to determine the configuration $\widehat{x}_{\mathrm{TRW}}$, which is guaranteed to be globally optimal for the original problem — that is, an element of

---

[2] See Theorem 7.2 on the reweighted sum-product algorithm and the accompanying discussion for more details on the choice of these edge weights.

$\arg\max_{x \in \mathcal{X}^m} \langle \theta, \phi(x) \rangle$ — so that the LP relaxation solves the original MAP problem. An interesting theoretical question is whether *any* fixed point (M^*, ν^*), regardless of whether it satisfies the criteria (8.23), specifies an optimal solution to the dual of the first-order LP relaxation (8.5). This question was left open by Wainwright et al. [245], and later resolved by Kolmogorov [137], who provided a counterexample, involving nonbinary variables, for which a TRW max-product fixed point does not correspond to a dual-optimal solution. In subsequent work, Kolmogorov and Wainwright [138] showed that for pairwise MRFs with binary variables, the equivalence between TRW message-passing and the LP relaxation is exact in all cases: any fixed point of TRW max-product specifies a dual-optimal solution to the first-order LP relaxation (8.5). Examples of pairwise MRFs with binary variables for which TRW message-passing always solves the LP relaxation include the Ising ground state problem, as well as various combinatorial problems such as the independent set problem and the vertex cover problem; see Example 8.4 for further discussion of these examples.

Kolmogorov [137] also established convergence guarantees for a certain sequential scheduling of TRW updates (known as TRW-S updates), and showed empirically that TRW-S updates converge faster than standard parallel scheduling of the TRW max-product updates (8.21). Various forms of these reweighted max-product algorithms have been applied in problems such as segmentation and disparity problems in computer vision [137, 139, 168, 227, 251, 265], error-control coding [76], side-chain prediction [251, 266], and sensor fusion [46, 49]. There also turn out to be a number of interesting connections between TRW max-product and a line of research, due to Schlesinger and collaborators, previously published in the Russian literature [140, 210]. The survey [256] provides a detailed overview of this line of work, and some connections to reweighted max-product and LP relaxation.

In addition to the basic TRW algorithm [245] and the TRW-S scheduling studied by Kolmogorov [137], other researchers have proposed distributed algorithms for solving the tree-based LP relaxation (8.5), including subgradient methods [76, 139], dual coordinate ascent methods [97, 239], annealing-type methods [124, 251], proximal optimization schemes [197], and adaptive LP solvers [228].

Weiss et al. [251] discuss connections between the zero-temperature limits of convex free energy problems, including the tree-reweighted Bethe problem from Theorem 7.2 as a special case, and optima of the first-order LP (8.5). Ravikumar et al. [197] discuss various rounding schemes that can be used for finite termination of LP-solving algorithms, with guarantees of correctness for the rounded solutions.

8.4.4 Examples of the First-order LP Relaxation

In this section, we discuss various examples of the first-order LP relaxation (8.5). One line of ongoing work in communication and information theory studies the behavior of the LP relaxation for decoding in error-control coding. When applied to combinatorial problems, the first-order LP relaxation recovers various known methods from the integer programming and approximation literature [180, 236]. In the special case of binary variables, there are a number of links to the literature on pseudo-Boolean optimization [37, 105].

Example 8.3 (LP Relaxations for Error-control Coding). We begin by discussing an instance of the first-order LP relaxation (8.5), introduced by Feldman et al. [78] for decoding low-density parity check (LDPC) codes. Recall from Example 3.6 the notion of an error-correcting code, and its definition as an exponential family. The code $\mathbb{C}$ is a subset of the Boolean hypercube $\{0,1\}^m$, defined by a set of parity checks ψ_a. The base measure ν is the counting measure restricted to the set $\mathbb{C}$ of all valid codewords, and the m-dimensional vector of sufficient statistics is given by $\phi(x) = (x_1, \ldots, x_m)$. From Example 3.9, the set $\mathcal{M}$ is given by the convex hull of all possible codewords, which is known as the *codeword polytope*. Figure 8.3 provides a toy example of an LDPC code over bits $(x_1, x_2, x_3, x_4) \in \{0,1\}^4$, and with two parity checks ψ_a and ψ_b, corresponding to the constraints $x_1 \oplus x_2 \oplus x_3 = 0$ and $x_2 \oplus x_3 \oplus x_4 = 0$, respectively. A family of codes is said to be *low-density* if the degrees of the parity checks and variable nodes in this graphical representation remain bounded as the blocklength m is increased.
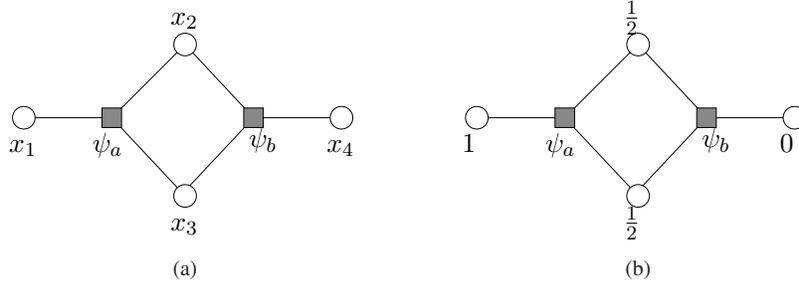
Fig. 8.3 (a) The factor graph representation of a toy binary linear code on four bits $(x_1, x_2, x_3, x_4) \in \{0,1\}^4$. Each codeword must satisfy the two parity check constraints $x_1 \oplus x_2 \oplus x_3 = 0$ and $x_2 \oplus x_3 \oplus x_4 = 0$, as defined by the constraint functions ψ_a and ψ_b. (b) Construction of a fractional vertex, also known as a pseudocodeword, for the code shown in panel (a). The pseudocodeword has elements $\widetilde{\tau} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}$, which satisfy all the constraints (8.25) defining the LP relaxation. However, the vector $\widetilde{\tau}$ cannot be expressed as a convex combination of codewords, and so lies strictly outside the codeword polytope, which corresponds to the marginal polytope for this graphical model.

When expressed as a graphical model using the parity check functions (see Equation (3.20)), the code is not immediately recognizable as a pairwise MRF to which the first-order LP relaxation (8.5) can be applied. However, any Markov random field over discrete variables can be converted into pairwise form by the selective introduction of auxiliary variables. In particular, suppose that for each check $a \in F$, we introduce an auxiliary variable z_a taking values in the space $\{0,1\}^{|N(a)|}$. Defining pairwise interactions between z_a and each bit x_i for nodes $i \in N(a)$, we can use z_a as a device to enforce constraints over the subvector $(x_i, \ i \in N(a))$. See Appendix E.3 for further details on converting a general discrete graphical model to an equivalent pairwise form.

If we apply the first-order LP relaxation (8.5) to this pairwise MRF, the relevant variables consist of a pseudomarginal vector $\begin{bmatrix} 1 - \tau_i & \tau_i \end{bmatrix}$ for each $i \in V$, and for each check $a \in F$, a set of pseudomarginals $\{\tau_{a;J}, \ J \text{ an even-sized subset of } N(a)\}$. In this case, the pairwise marginalization conditions that define the set $\mathbb{L}(G)$ reduce to

$$\sum_{J \ni i} \tau_{a;J} = \tau_i, \quad \text{for each } a, \text{ and } i \in N(a), \tag{8.24}$$

along with the box constraints $\tau_i, \tau_{a;J} \in [0,1]$. It is also possible to remove the variables $\tau_{a;J}$ by Fourier–Motzkin elimination [24], so as to obtain an equivalent LP relaxation that is described only in terms of the vector $\begin{bmatrix} \tau_1 & \tau_2 & \dots & \tau_m \end{bmatrix}$. Doing so shows that the first-order relaxation (8.5), when applied to the coding problem, is characterized by the box constraints $\tau_i \in [0,1]$ for each $i \in V$, and for each $a \in F$, the *forbidden set* constraints

$$\sum_{k \in K} (1 - \tau_k) + \sum_{k \in N(a) \setminus K} \tau_k \geq 1 \quad \forall\, K \subset N(a) \text{ with } |K| \text{ odd.} \quad (8.25)$$

The interpretation of this inequality is very intuitive: it enforces that for each parity check $a \in F$, the subvector $\tau_{N(a)} = (\tau_i,\, i \in N(a))$ must be at Hamming distance at least one from any odd-parity configuration over the bits $N(a)$.

In practice, a binary codeword $x \in \mathbb{C}$ is transmitted through a channel, so that the user receives only a vector of noisy observations. As described in Example 3.6, channel transmission can be modeled in terms of a conditional distribution $p(y_i \mid x_i)$. Given a particular received sequence $(y_1, \dots, y_m)$ from the channel, define the vector $\theta \in \mathbb{R}^m$ of log likelihoods, with components $\theta_i = \log \frac{p(y_i|x_i=1)}{p(y_i|x_i=0)}$. The ML decoding problem corresponds to the integer program of maximizing the likelihood $\sum_{i \in V} \theta_i x_i$ over the discrete set of codewords $x \in \mathbb{C}$. It is well known to be computationally intractable [18] in a worst case sense.

The LP decoding algorithm of Feldman et al. [78] is based on maximizing the objective $\sum_{i \in V} \theta_i \tau_i$ subject to the box constraints $(\tau_1, \tau_2, \dots, \tau_m) \in [0,1]^m$ as well as the forbidden set constraints (8.25). Since its introduction [76, 78], the performance of this LP relaxation has been extensively studied [e.g., 50, 53, 62, 70, 77, 136, 228, 238, 239]. Not surprisingly, given the role of the constraint set $\mathbb{L}(G)$ in the Bethe variational problem, there are close connections between LP decoding and standard iterative algorithms like sum-product decoding [76, 78, 136]. Among other connections, the fractional extreme points of the first-order LP relaxation have a very specific interpretation as *pseudocodewords* of the underlying code, studied in earlier work on iterative decoding [83, 86, 257]. Figure 8.3(b) provides a concrete illustration of a pseudocodeword that arises when the relaxation is applied to the toy

code shown in Figure 8.3(b). Consider the vector $\widetilde{\tau} := (1, \frac{1}{2}, \frac{1}{2}, 0)$; it is easy to verify that it satisfies the box inequalities and the forbidden set constraints (8.25) that define the LP relaxation. (In fact, with a little more work, it can be shown that $\widetilde{\tau}$ is a vertex of the relaxed LP decoding polytope.) However, we claim that $\widetilde{\tau}$ does *not* belong to the marginal polytope for this graphical model — that is, it cannot be written as a convex combination of codewords. To see this fact, note that by taking a modulo two sum of the parity check constraints $x_1 \oplus x_2 \oplus x_3 = 0$ and $x_2 \oplus x_3 \oplus x_4 = 0$, we obtain that $x_1 \oplus x_4 = 0$ for any codeword. Therefore, for any vector μ in the codeword polytope, we must have $\mu_1 = \mu_4$, which implies that $\widetilde{\tau}$ lies outside the codeword polytope.

Apart from its interest in the context of error-control coding, the fractional vertex in Figure 8.3(b) also provides a counterexample regarding the *persistency* of fractional vertices of the polytope $\mathbb{L}(G)$. Recall from our discussion at the end of Section 8.4.1 that the first-order LP relaxation, when applied to integer programs with binary variables, has the strong persistency property [105], as summarized in Proposition 8.4. It is natural to ask whether strong persistency also holds for higher-order variables as well. Note that after conversion to a pairwise Markov random field (to which the first-order LP relaxation applies), the coding problem illustrated in Figure 8.3(a) includes nonbinary variables z_a and z_b at each of the factor nodes $a, b \in F$. We now claim that the fractional vertex $\widetilde{\tau}$ illustrated in Figure 8.3 constitutes a failure of strong persistency for the first-order LP relaxation with higher-order variables. Consider applying the LP decoder to the cost function $\theta = (2, 0, 0, -1)$; using the previously constructed $\widetilde{\tau} = (1, \frac{1}{2}, \frac{1}{2}, 0)$, a little calculation shows that $\langle \theta, \widetilde{\tau} \rangle = 2$. In contrast, the optimal codewords are $x^* = (1, 1, 0, 1)$ and $y^* = (1, 0, 1, 1)$, both with value $\langle \theta, x \rangle = \langle \theta, y^* \rangle = 1$. Thus, neither of the optimal integral solutions have $x_4 = 0$, even though $\widetilde{\tau}_4 = 0$ in the fractional vertex. Consequently, strong persistency is violated for this instance.

We conclude that relaxation (8.5) is strongly persistent only for pairwise Markov random fields over binary random variables, otherwise

known as binary quadratic programs. We now turn to an in-depth consideration of this particular case:

Example 8.4 (Binary Quadratic Programs and Combinatorial Problems). Recall the Ising model, as first introduced in Example 3.1: it is an exponential family over a vector X of binary random variables, which may take either "spin" values $\{-1,+1\}^m$, or zero-one values $\{0,1\}^m$. Note that the mapping $x_s \mapsto 2x_s - 1$ and its inverse $z_s \mapsto \frac{1}{2}(z_s + 1)$ may be used to convert freely back and forth from the $\{0,1\}$ form to the $\{-1,+1\}$ form.

Let us consider the $\{0,1\}$-case and the mode-finding problem associated with the canonical overcomplete set of potential functions — namely

$$\max_{x \in \{0,1\}^m} \langle \theta, \phi(x) \rangle = \max_{x \in \{0,1\}^m} \left\{ \sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t) \right\},$$
(8.26)

where

$$\theta_s(x_s) := \sum_{j=0}^{1} \theta_{s;j} \mathbb{I}_j[x_s] \quad \text{and} \quad \theta_{st}(x_s, x_t) := \sum_{j,k=0}^{1} \theta_{st;jk} \mathbb{I}_j[x_s] \mathbb{I}_k[x_t]$$

are weighted sums of indicator functions.

This problem is a binary quadratic program, and includes as special cases various types of classical problems in combinatorial optimization.

Independent set and vertex cover: Given an undirected graph $G = (V, E)$, an *independent set* I is a subset of vertices such that $(s,t) \notin E$ for all $s, t \in I$. Given a set of vertex weights $w_s \geq 0$, the maximum weight independent set (MWIS) problem is to find the independent set I with maximal weight, $w(I) := \sum_{s \in I} w_s$. This problem has applications in scheduling for wireless sensor networks, where the independent set constraints are imposed to avoid interference caused by having neighboring nodes transmit simultaneously.

To model this combinatorial problem as an instance of the binary quadratic program, let $X \in \{0,1\}^m$ be an indicator vector for membership in S, meaning that $X_s = 1$ if and only if $s \in S$. Then define

$\theta_s(x_s) = \begin{bmatrix} 0 & w_s \end{bmatrix}$, and the pairwise interaction

$$\theta_{st}(x_s, x_t) = \begin{bmatrix} 0 & 0 \\ 0 & -\infty \end{bmatrix}.$$

With these definitions, the binary quadratic program corresponds to the maximum weight independent set (MWIS) problem.

A related problem is that of finding a *vertex cover* — that is, a set C of vertices such that for any edge $(s,t) \in E$, at least one of s or t belongs to C — with minimum weight $w(C) = \sum_{s \in C} w_s$. This minimum weight vertex cover (MWVC) problem can be recast as a maximization problem: it is another special case of the binary QP with $\theta_s(x_s) = \begin{bmatrix} 0 & -w_s \end{bmatrix}$, and

$$\theta_{st}(x_s, x_t) = \begin{bmatrix} -\infty & 0 \\ 0 & 0 \end{bmatrix}.$$

Let us now consider how the first-order LP relaxation (8.5) specializes to these problems. Recall that the general LP relaxation is in terms of the two-vector of singleton pseudomarginals $\tau_s = \begin{bmatrix} \tau_{s;0} & \tau_{s;1} \end{bmatrix}$, and an analogous 2×2 matrix τ_{st} of pairwise pseudomarginals. In the independent set problem, the parameter setting $\theta_{st;11} = -\infty$ is tantamount to enforcing the constraint $\tau_{st;11} = 0$, so that the LP relaxation can be expressed purely in terms of a vector $\mu \in \mathbb{R}^m$ with elements $\mu_s := \tau_{s;1}$. After simplification, the constraints defining $\mathbb{L}(G)$ (see Proposition 4.1) can be reduced to $\mu_s \geq 0$ for all nodes $s \in V$, and $\mu_s + \mu_t \leq 1$ for all edges $(s,t) \in E$. Thus, for the MWIS problem, the first-order relaxation (8.5) reduces to the linear program

$$\max_{\mu \geq 0} \sum_{s \in V} w_s \mu_s \quad \text{such that } \mu_s + \mu_t \leq 1 \text{ for all } (s,t) \in E.$$

This LP relaxation is the classical one for the independent set problem [180, 236]. Sanghavi et al. [207] discuss some connections between the ordinary max-product algorithm and this LP relaxation, as well as to auction algorithms [22].

In a similar way, specializing the first-order LP relaxation (8.5) to the MWVC problem yields the linear program

$$\min_{\mu \geq 0} \sum_{s \in V} w_s \mu_s \quad \text{such that } \mu_s + \mu_t \geq 1 \text{ for all } (s,t) \in E,$$

which is another classical LP relaxation from the combinatorics literature [180].

MAX-CUT: Consider the following graph-theoretic problem: given a nonnegative weight $w_{st} \geq 0$ for each edge of an undirected graph $G = (V, E)$, find a partition (U, U^c) of the vertex set such that the associated weight

$$w(U, U^c) := \sum_{\{(s,t) \mid s \in U, t \in U^c\}} w_{st}$$

of edges across the partition is maximized. To model this MAX-CUT problem as an instance of the binary quadratic program, let $X \in \{0,1\}^m$ be an indicator vector for membership in U, meaning that $X_s = 1$ if and only if $s \in U$. Then define $\theta_s(x_s) = 0$ for all vertices, and define the pairwise interaction

$$\theta_{st}(x_s, x_t) = \begin{bmatrix} 0 & w_{st} \\ w_{st} & 0 \end{bmatrix}. \tag{8.27}$$

With these definitions, a little algebra shows that problem (8.26) is equivalent to the MAX-CUT problem, a canonical example of an NP-complete problem. As before, the first-order LP relaxation (8.5) can be specialized to this problem. In Section 9, we also describe the celebrated semidefinite program (SDP) relaxation for MAX-CUT due to Goemans and Williamson [98] (see Example 9.3).

Supermodular and submodular interactions: An important subclass of binary quadratic programs are those based on supermodular potential functions [158]. The interaction θ_{st} is *supermodular* if

$$\theta_{st}(1,1) + \theta_{st}(0,0) \geq \theta_{st}(1,0) + \theta_{st}(0,1), \tag{8.28}$$

and it is *submodular* if the function $g_{st}(x_s, x_t) = -\theta_{st}(x_s, x_t)$ is supermodular. Note that the MAX-CUT problem and the independent set problem both involve submodular potential functions, whereas the vertex cover problem involves supermodular potentials. It is well known that the class of *regular binary QPs* — meaning supermodular maximization problems or submodular minimization problems — can be solved in polynomial time. As a particular instance, consider the problem of finding the minimum s–t cut in a graph. This can be formulated

as a minimization problem in terms of the potential functions (8.27), with additional nonzero singleton potentials θ_s, yielding an instance of submodular minimization. It is easily solved by conversion to a maximum flow problem, using the classical Ford–Fulkerson duality theorem [22, 101]. However, the MAX-CUT, independent set, and vertex cover problems all fall outside the class of regular binary QPs, and indeed are canonical instances of intractable problems.

Among other results, Kolmogorov and Wainwright [138] establish that tree-reweighted max-product is exact for any regular binary QP. The same statement fails to hold for the ordinary max-product updates, since it fails on the regular binary QP discussed in Example 8.2. The exactness of tree-reweighted max-product stems from the tightness of the first-order LP relaxation (8.5) for regular binary QPs. This tightness can be established via results due to Hammer et al. [105] on the so-called roof dual relaxation in pseudo-Boolean optimization, which turns out to be equivalent to the tree-based relaxation (8.5) for the special case of binary variables.

Finally, we discuss a related class of combinatorial problems for which some recent work has studied message-passing and linear programming:

Example 8.5 (Maximum Weight Matching Problems). Given an undirected graph $G = (V, E)$, the matching problem is to find a subset F of edges, such that each vertex is adjacent to at most one edge $e \in F$. In the weighted variant, each edge is assigned a weight w_e, and the goal is to find the matching F that maximizes the weight function $\sum_{e \in F} w_e$. This maximum weight matching (MWM) problem is well known to be solvable in polynomial time for any graph [212]. For bipartite graphs, the MWM problem can be reduced to an especially simple linear program, which we derive here as a special case of the first-order relaxation (8.5).

In order to apply the first-order relaxation, it is convenient to first reformulate the matching problem as a mode-finding problem in an MRF described by a factor graph, and then convert the factor graph to a pairwise form, as in Example 8.3. We begin by associating with the

original graph $G = (V, E)$ a hypergraph $\widetilde{G}$, in which each edge $e \in E$ corresponds to a vertex of $\widetilde{G}$, and each vertex $s \in V$ corresponds to a hyperedge. The hyperedge indexed by s connects to all those vertices of $\widetilde{G}$ (edges of the original graph) in the set $E(s) = \{e \in E \mid s \in e\}$. Finally, we define a Markov random field over the hypergraph as follows. First, let each $e \in E$ be associated with a binary variable $x_e \in \{0, 1\}$, which acts as an indicator variable for whether edge e participates in the matching. We define the weight function $\theta_e(x_e) = w_e x_e$, where w_e is the weight specified for edge e in the matching problem. Second, given the subvector $x_{E(s)} = (x_e, \, e \in E(s))$, we define an associated interaction potential

$$\theta_s(x_{E(s)}) := \begin{cases} 0 & \text{if } \sum_{e \ni s} x_e \leq 1 \\ -\infty & \text{otherwise.} \end{cases}$$

With this definition, the mode-finding problem

$$\max_{x \in \{0,1\}^{|E|}} \left\{ \sum_{e \in E} \theta_e(x_e) + \sum_{s \in V} \theta(x_{E(s)}) \right\} \tag{8.29}$$

in the hypergraph is equivalent to the original matching problem.

This hypergraph-based mode-finding problem (8.29) can be converted to an equivalent mode-finding problem in a pairwise Markov random field by following the generic recipe described in Appendix E.3. Doing so and applying the first-order LP relaxation (8.5) to the resulting pairwise MRF yields the following LP relaxation of the maximum weight matching M^*:

$$M^* \leq \max_{\tau \in \mathbb{R}^{|E|}} \sum_{e \in E} w_e \tau_e$$

$$\text{s.t.} \quad x_e \geq 0 \quad \forall e \in E, \quad \text{and} \quad \sum_{e \ni s} x_e \leq 1 \quad \forall s \in V. \tag{8.30}$$

This LP relaxation is a classical one for the matching problem, known to be tight for any bipartite graph but loose for nonbipartite graphs [212]. A line of recent research has established close links between the LP relaxation and the ordinary max-product algorithm, including the case of bipartite weighted matching [15], bipartite

weighted *b*-matching [116], weighted matching on general graphs [206], and weighted *b*-matching on general graphs [13].

8.5   Higher-order LP Relaxations

The tree-based relaxation (8.17) can be extended to hypertrees of higher treewidth t, by using the hypertree-based outer bounds $\mathbb{L}_t(G)$ on marginal polytopes described in Section 4.2.3. This extension produces a sequence of progressively tighter LP relaxations, which we describe here. Given a hypergraph $G = (V, E)$, we use the shorthand notation $x_h := (x_i, \; i \in h)$ to denote the subvector of variables associated with hyperedge $h \in E$. We define interaction potentials $\theta_h(x_h) := \sum_J \theta_{h;J} \mathbb{I}[x_h = J]$ and consider the mode-finding problem of computing

$$x^* \in \arg \max_{x \in \mathcal{X}^m} \left\{ \sum_{h \in E} \theta_h(x_h) \right\}. \qquad (8.31)$$

Note that problem (8.31) generalizes the analogous problem for pairwise MRFs, a special case in which the hyperedge set consists of only vertices and ordinary edges.

Letting $t + 1$ denote the maximal cardinality of any hyperedge, the relaxation based on $\mathbb{L}_t(G)$ involves the collection of pseudomarginals $\{\tau_h \mid h \in E\}$ subject to local consistency constraints

$$\mathbb{L}_t(G) := \left\{ \tau \geq 0 \mid \sum_{x_h'} \tau_h(x_h') = 1 \ \forall \, h \in E, \quad \text{and} \right.$$

$$\left. \sum_{\{x_h' \mid x_g' = x_g\}} \tau_h(x_h') = \tau_g(x_g) \ \forall g \subset h \right\}. \qquad (8.32)$$

Following the same reasoning as in Section 8.4.1, we have the upper bound

$$\max_{x \in \mathcal{X}^m} \left\{ \sum_{h \in E} \theta_h(x_h) \right\} \leq \underbrace{\max_{\tau \in \mathbb{L}_t(G)} \sum_{h \in E} \left[\sum_{x_h} \tau_h(x_h) \theta_h(x_h) \right]}_{\max_{\tau \in \mathbb{L}_t(G)} \langle \tau, \theta \rangle}. \quad (8.33)$$

Notice that the relaxation (8.33) can be applied directly to a graphical model that involves higher-order interactions, obviating the need to convert higher-order interactions into pairwise form, as done in illustrating the first-order LP relaxation (see Example 8.3). In fact, in certain cases, this direct application can yield a tighter relaxation than that based on conversion to the pairwise case, as illustrated in Example 8.6. Of course, the relaxation (8.33) can also be applied directly to a pairwise Markov random field, which can always be embedded into a hypergraph with higher-order interactions. Thus, Equation (8.33) actually describes a sequence of relaxations, based on the nested constraint sets

$$\mathbb{L}_1(G) \supseteq \mathbb{L}_2(G) \supseteq \mathbb{L}_3(G) \supseteq \ldots \mathbb{L}_t(G) \supseteq \ldots \supseteq \mathbb{M}(G), \qquad (8.34)$$

with increasing accuracy as the interaction size t is increased. The tradeoff, of course, is that computational complexity of the relaxation also increases in the parameter t. The following result is an immediate consequence of our development thus far:

Proposition 8.6 (Hypertree Tightness). For any hypergraph G of treewidth t, the LP relaxation (8.33) based on the set $\mathbb{L}_t(G)$ is exact.

Proof. This assertion is equivalent to establishing that $\mathbb{L}_t(G) = \mathbb{M}(G)$ for any hypergraph G of treewidth t. Here $\mathbb{M}(G)$ denotes the marginal polytope, corresponding to marginal probability distributions $(\mu_h, \ h \in E)$ defined over the hyperedges that are globally consistent. First, the inclusion $\mathbb{L}_t(G) \supseteq \mathbb{M}(G)$ is immediate, since any set of globally consistent marginals must satisfy the local consistency conditions defining $\mathbb{L}_t(G)$.

In the reverse direction, consider a locally consistent set of pseudomarginals $\tau \in \mathbb{L}_t(G)$. Using these pseudomarginals, we may construct the functions φ_h defined previously (4.40) in our discussion of hypertree factorization. Using these quantities, let us define distribution

$$p_\tau(x_1, x_2, \ldots, x_m) := \prod_{h \in E} \varphi_h(x_h; \tau). \qquad (8.35)$$

This distribution is constructed according to the factorization princi-
ple (4.42) for hypertrees. Using the local normalization and marginal-
ization conditions that $(\tau_h, \; h \in E)$ satisfies by virtue of membership
in $\mathbb{L}_t(G)$, it can be verified that this this distribution is properly nor-
malized (i.e., $\sum_x p_\tau(x) = 1$). Moreover, for each hyperedge $h \in E$, this
distribution has marginal distribution τ_h, meaning that

$$\sum_{x_s, s \notin h} p_\tau(x_1, \ldots, x_m) = \tau_h(x_h) \qquad \text{for all } h \in E.$$

Again, this fact can be verified by direct calculation from the factoriza-
tion, or can be derived as a consequence of the junction tree theorem.
Consequently, the distribution (8.35) provides a certificate of the mem-
bership of τ in the marginal polytope $\mathbb{M}(G)$. $\hfill \square$

In the binary $\{0,1\}$ case, the sequence of relaxations (8.34) has been
proposed and studied previously by Hammer et al. [105], Boros et
al. [36], and Sherali and Adams [217], although without the connections
to the underlying graphical structure provided by Proposition 8.6.

**Example 8.6 (Tighter Relaxations for Higher-order Interac-
tions).** We begin by illustrating how higher-order LP relaxations yield
tighter constraints with a continuation of Example 8.3. Consider the
factor graph shown in Figure 8.4(a), corresponding to a hypergraph-
structured MRF of the form

$$p_\theta(x) \propto \exp\left\{ \sum_{i=1}^{4} \theta_i(x_i) + \theta_a(x_1, x_2, x_3) + \theta_b(x_2, x_3, x_4) \right\}, \qquad (8.36)$$

for suitable potential functions $\{\theta_i, i = 1, \ldots 4\}$, θ_a, and θ_b. In this exam-
ple, we consider two different procedures for obtaining an LP relaxation:

(a) First convert the hypergraph MRF (8.36) into a pairwise
MRF and then apply the first-order relaxation (8.5); or
(b) Apply the second-order relaxation based on $\mathbb{L}_2(G)$ directly
to the original hypergraph MRF.

After some algebraic manipulation, both relaxations can be expressed
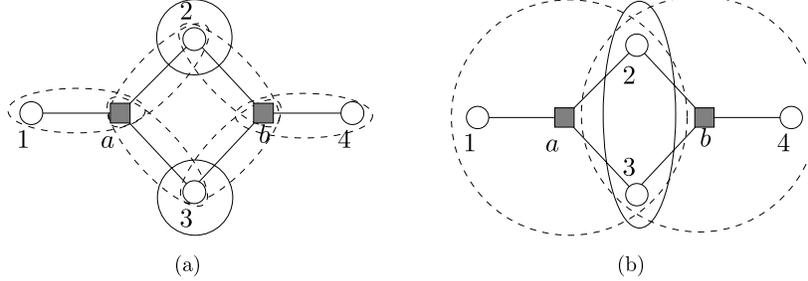purely in terms of the triplet pseudomarginals $\tau_{123}(x_1, x_2, x_3)$ and

Fig. 8.4 (a) A hypergraph-structured Markov random field with two sets of triplet interactions over $a = \{1,2,3\}$ and $b = \{2,3,4\}$. The first-order LP relaxation (8.5) applied to this graph first converts it into an equivalent pairwise MRF, and thus enforces only consistency only via the singleton marginals τ_2 and τ_3. (b) The second-order LP relaxation enforces additional consistency over the shared pairwise pseudomarginal τ_{23}, and is exact for this graph.

$\tau_{234}(x_2, x_3, x_4)$, and in particular their consistency on the overlap (x_2, x_3). The basic first-order LP relaxation (procedure (a)) imposes only the two marginalization conditions

$$\sum_{x_1', x_2'} \tau_{123}(x_1', x_2', x_3) = \sum_{x_2', x_4'} \tau_{234}(x_2', x_3, x_4'), \quad \text{and} \qquad (8.37a)$$

$$\sum_{x_1', x_3'} \tau_{123}(x_1', x_2, x_3') = \sum_{x_3', x_4'} \tau_{234}(x_2, x_3', x_4'), \qquad (8.37b)$$

which amount to ensuring that the *singleton* pseudomarginals τ_2 and τ_3 induced by τ_{123} and τ_{234} agree. Note, however, that the first-order relaxation imposes no constraint on the pairwise marginal(s) τ_{23} induced by the triplet.

In contrast, the second-order relaxation treats the triplets (x_1, x_2, x_3) and (x_2, x_3, x_4) exactly, and so addition to the singleton conditions (8.37), also requires agreement on the overlap — viz.

$$\sum_{x_1'} \tau_{123}(x_1', x_2, x_3) = \sum_{x_4'} \tau_{234}(x_2, x_3, x_4'). \qquad (8.38)$$

As a special case of Proposition 8.6, this second-order relaxation is tight, since the hypergraph in Figure 8.4(a) has treewidth two.

In Example 8.3, we considered the first-order LP relaxation applied to the MRF in Figure 8.4(a) for the special case of a linear code defined

over binary random variables $x \in \{0,1\}^4$. There we constructed the fractional vector

$$\widetilde{\tau} \; = \; \big(\tau_1(1), \tau_2(1), \tau_3(1), \tau_4(1)\big) \; = \; (1, \frac{1}{2}, \frac{1}{2}, 0),$$

and showed that it was a fractional vertex for the relaxed polytope. Although the vector $\widetilde{\tau}$ is permitted under the pairwise relaxation in Figure 8.4(a), we claim that it is forbidden by the second-order relaxation in Figure 8.4(b). Indeed, the singleton pseudomarginals $\widetilde{\tau}$ are consistent with triplet pseudomarginals τ_{123} and τ_{234} defined as follows: let τ_{123} assign mass $\frac{1}{2}$ to the configurations $(x_1, x_2, x_3) = (101)$ and (110), with zero mass elsewhere, and let τ_{234} assign mass $\frac{1}{2}$ to the configurations $(x_2, x_3, x_4) = (000)$ and (110), and zero elsewhere. By computing the induced marginals, it can be verified that τ_{123} and τ_{234} marginalize down to $\widetilde{\tau}$, and satisfy the first-order consistency condition (8.37) required by the first-order relaxation. However, the second-order relaxation also requires agreement over the overlap (x_2, x_3), as expressed by condition (8.38). On one hand, by definition of τ_{123}, we have

$$\sum_{x_1'} \tau_{123}(x_1', x_2, x_3) = \frac{1}{2}\mathbb{I}[(x_2, x_3) = (0,1)] + \frac{1}{2}\mathbb{I}[(x_2, x_3) = (1,0)],$$

whereas on the other hand, by definition of τ_{234}, we have

$$\sum_{x_1'} \tau_{234}(x_2, x_3, x_4') = \frac{1}{2}\mathbb{I}[(x_2, x_3) = (1,1)] + \frac{1}{2}\mathbb{I}[(x_2, x_3) = (1,1)].$$

Consequently, condition (8.38) is violated.

More generally, the second-order LP relaxation is exact for Figure 8.4(b), meaning that it is impossible to find any triplet pseudomarginals τ_{123} and τ_{234} that marginalize down to $\widetilde{\tau}$, and agree over the overlap (x_2, x_3).

Of course, the treewidth-two relaxation can be applied directly to MRFs with cost functions already in pairwise form. In explicit terms, the second-order LP relaxation applied to a pairwise MRF has the form

$$\max_{\tau} \left\{ \sum_{s \in V} \left[\sum_{x_s} \tau_s(x_s)\theta_s(x_s) \right] + \sum_{(s,t) \in E} \left[\sum_{x_s, x_t} \tau_{st}(x_s, x_t)\theta_{st}(x_s, x_t) \right] \right\},$$

$$(8.39)$$

subject to the constraints

$$\tau_s(x_s) = \sum_{x'_t, x'_u} \tau_{stu}(x_s, x'_t, x'_u) \qquad \forall (s,t,u) \ni s, \quad \forall s \in V \qquad (8.40a)$$

$$\tau_{st}(x_s, x_t) = \sum_{x'_u} \tau_{stu}(x_s, x_t, x'_u) \quad \forall (s,t,u) \ni (s,t), \ \forall (s,t) \in E. \quad (8.40b)$$

Assuming that all edges and vertices are involved in the cost function, this relaxation involves m singleton marginals, $\binom{m}{2}$ pairwise marginals, and $\binom{m}{3}$ triplet marginals.

Note that even though the triplet pseudomarginals τ_{stu} play no role in the cost function (8.39) itself, they nonetheless play a central role in the relaxation via the consistency conditions (8.40) that they impose. Among other implications, Equations (8.40a) and (8.40b) imply that the pairwise marginals must be consistent with the singleton marginals (i.e., $\sum_{x'_t} \tau_{st}(x_s, x'_t) = \tau_s(x_s)$). Since these pairwise conditions define the first-order LP relaxation (8.5), this fact confirms that the second-order LP relaxation is at least as good as the first-order one. In general, the triplet consistency also imposes additional constraints not ensured by pairwise consistency alone. For instance, Example 4.1 shows that the pairwise constraints are insufficient to characterize the marginal polytope of a single cycle on three nodes, whereas Proposition 8.6 implies that the triplet constraints (8.40) provide a complete characterization, since a single cycle on three nodes has treewidth two.

This technique — namely, introducing additional parameters in order to tighten a relaxation, such as the triplet pseudomarginals τ_{stu} — is known as a *lifting operation*. It is always possible, at least in principle, to project the lifted polytope down to the original space, thereby yielding an explicit representation of the tightened relaxation in the original space. This combination is known as a *lift-and-project* method [217, 150, 159]. In the case of the lifted relaxation based on the triplet consistency (8.40) defining $\mathbb{L}_2(G)$, the projection is from the full space down to the set of pairwise and singleton marginals.

Example 8.7 (Lift-and-project for Binary MRFs). Here we illustrate how to project the triplet consistency constraints (8.40) for any binary Markov random field, thereby deriving the so-called cycle

inequalities for the binary marginal polytope. (We presented these cycle inequalities earlier in Example 4.1; see Section 27.1 of Deza and Laurent [69] for a complete discussion.) In order to do so, it is convenient to work with a minimal set of pseudomarginal parameters: in particular, for a binary Markov random field, the seven numbers $\{\tau_{stu}, \tau_{st}, \tau_{su}, \tau_{tu}, \tau_s, \tau_t, \tau_u\}$ suffice to characterize the triplet pseudo-marginal over variables (X_s, X_t, X_u). Following a little algebra, it can be shown that the singleton (8.40a) and pairwise consistency (8.40b) conditions are equivalent to the inequalities:

$$\tau_{stu} \geq 0 \tag{8.41a}$$

$$\tau_{stu} \geq -\tau_s + \tau_{st} + \tau_{su} \tag{8.41b}$$

$$\tau_{stu} \leq 1 - \tau_s - \tau_t - \tau_u + \tau_{st} + \tau_{su} + \tau_{tu} \tag{8.41c}$$

$$\tau_{stu} \leq \tau_{st}, \tau_{su}, \tau_{tu}. \tag{8.41d}$$

Note that following permutations of the triplet (s, t, u), there are eight inequalities in total, since inequality (8.41b) has three distinct versions.

The goal of projection is to eliminate the variable τ_{stu} from the description, and obtain a set of constraints purely in terms of the singleton and pairwise pseudomarginal parameters. If we consider singleton, pairwise, and triplet pseudomarginals for all possible combinations, there are a total of $T = m + \binom{m}{2} + \binom{m}{3}$ possible pseudomarginal parameters. We would like to project this subset of $\mathbb{R}^T$ down to a lower-dimensional subset of $\mathbb{R}^L$, where $L = m + \binom{m}{2}$ is the total number of singleton and pairwise parameters. More specifically, we would like to determine the set $\Pi(\mathbb{L}_2(G)) \subset \mathbb{R}^L$ of pseudomarginal parameters of the form

$$\big\{ (\tau_s, \tau_t, \tau_u, \tau_{st}, \tau_{su}, \tau_{tu}) \mid \exists\, \tau_{stu} \text{ such that inequalities (8.41) hold} \big\},$$

corresponding to the projection of $\mathbb{L}_2(G)$ down to $\mathbb{R}^L$.

A classical technique for computing such projections is Fourier–Motzkin elimination [24, 271]. It is based on the following two steps: (a) first express all the inequalities so that the variable τ_{stu} to be eliminated appears on the left-hand side; and (b) then combine the ($\leq$) constraints with the ($\geq$) constraints in pairs, thereby yielding a new inequality in which the variable τ_{stu} no longer plays a role. Note that τ_{stu} appears

on the left-hand side of all the inequalities (8.41); hence, performing step (b) yields the following inequalities

$$\tau_{st}, \tau_{su}, \tau_{tu} \geq 0, \quad \text{combine (8.41a) and (8.41d)}$$
$$1 + \tau_{tu} - \tau_t - \tau_u \geq 0, \quad \text{combine (8.41b) and (8.41c)}$$
$$\tau_s - \tau_{su} \geq 0, \quad \text{combine (8.41b) and (8.41d)}$$
$$\tau_s + \tau_{tu} - \tau_{su} - \tau_{st} \geq 0, \quad \text{combine (8.41b) and (8.41d)}$$
$$1 - \tau_s - \tau_t - \tau_u + \tau_{st} + \tau_{su} + \tau_{tu} \geq 0, \quad \text{combine (8.41a) and (8.41c).}$$

The first three sets of inequalities should be familiar; in particular, they correspond to the constraints defining the first-order relaxation, in the special case of binary variables (see Example 4.1). Finally, the last two inequalities, and all permutations thereof, correspond to a known set of inequalities on the binary marginal polytope, usually referred to as the cycle inequalites.[3] In recent work, Sontag and Jaakkola [219] examined the use of these cycle inequalities, as well as their extensions to non-binary variables, in variational methods for approximate marginalization, and showed significantly more accurate results in many cases.

---

[3] To be clear, in the specified form, these inequalities are usually referred to as the triangle inequalities, for obvious reasons. Note that for a graph $G = (V, E)$ with $|V|$ variables, there are a total of $\binom{|V|}{3}$ such groups of triangle inequalities. However, if the graph G is not fully connected, then some of these triangle inequalities involve mean parameters μ_{uv} for pairs (u, v) *not* in the graph edge set. In order to obtain inequalities that involve only mean parameters μ_{st} for edges $(s, t) \in E$, one can again perform Fourier–Motzkin elimination, which leads to the so-called cycle inequalities. See Deza and Laurent [69] for more details.

9

Moment Matrices, Semidefinite Constraints, and Conic Programming Relaxation

Although the linear constraints that we have considered thus far yield a broad class of relaxations, many problems require a more expressive framework for imposing constraints on parameters. In particular, as we have seen at several points in the preceding sections, semidefinite constraints on moment matrices arise naturally within the variational approach — for instance, in our discussion of Gaussian mean parameters from Example 3.7. This section is devoted to a more systematic study of moment matrices, in particular their use in constructing hierarchies of relaxations based on conic programming. Moment matrices and conic programming provide a very expressive language for the design of variational relaxations. In particular, we will see that the LP relaxations considered in earlier sections are special cases of conic relaxations in which the underlying cone is the positive orthant. We will also see that moment matrices allow us to define a broad class of additional conic relaxations based on semidefinite programming (SDP) and second-order cone programming (SOCP).

The study of moment matrices and their properties has an extremely rich history (e.g., [3, 130]), particularly in the context of scalar random variables. The basis of our presentation is more recent work

[e.g., 147, 148, 150, 191] that applies to multivariate moment problems. While much of this work aims at general classes of problems in algebraic geometry, in our treatment we limit ourselves to considering marginal polytopes and we adopt the statistical perspective of imposing positive semidefiniteness on covariance and other moment matrices. The moment matrix perspective allows for a unified treatment of various relaxations of marginal polytopes. See Wainwright and Jordan [247] for additional material on the ideas presented here.

9.1 Moment Matrices and Their Properties

Given a random vector $Y \in \mathbb{R}^d$, consider the collection of its second-order moments: $\lambda_{st} = \mathbb{E}[Y_s Y_t]$, for $s, t = 1, \dots, d$. Using these moments, we can form the following symmetric $d \times d$ matrix:

$$N[\lambda] := \mathbb{E}[YY^T] = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \cdots & \lambda_{1n} \\ \lambda_{21} & \lambda_{22} & \cdots & \lambda_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \lambda_{n1} & \lambda_{n2} & \cdots & \lambda_{nn} \end{bmatrix}. \tag{9.1}$$

At first sight, this definition might seem limiting, because the matrix involves only second-order moments. However, given some random vector X of interest, we can expose any of its moments by defining $Y = f(X)$ for a suitable choice of function f, and then considering the associated second-order moment matrix (9.1) for Y. For instance, by setting $Y := \begin{bmatrix} 1 & X \end{bmatrix} \in \mathbb{R} \times \mathbb{R}^m$, the moment matrix (9.1) will include both first and second-order moments of X. Similarly, by including terms of the form $X_s X_t$ in the definition of Y, we can expose third moments of X. The significance of the moment matrix (9.1) lies in the following simple result:

Lemma 9.1 (Moment Matrices). Any valid moment matrix $N[\lambda]$ is positive semidefinite.

Proof. We must show that $a^T N[\lambda] a \geq 0$ for an arbitrary vector $a \in \mathbb{R}^d$. If λ is a valid moment vector, then it arises by taking expectations

under some distribution p. Accordingly, we can write

$$a^T N[\lambda] a = \mathbb{E}_p[a^T Y Y^T a] = \mathbb{E}_p[\|a^T Y\|^2],$$

which is clearly nonnegative.   □

Lemma 9.1 provides a *necessary* condition for a vector $\lambda = (\lambda_{st}, \ s,t = 1,\ldots,d)$ to define balid second-order moments. Such a condition is both necessary and sufficient for certain classical moment problems involving scalar random variables [e.g., 114, 130]. This condition is also necessary and sufficient for a multivariate Gaussian random vector, as discussed in Example 3.7.

9.1.1   Multinomial and Indicator Bases

Now, consider a discrete random vector $X \in \{0,1,\ldots,r-1\}^m$. In order to study moments associated with X, it is convenient to define some function bases. Our first basis involves multinomial functions over $(x_1,\ldots,x_m)$. Each multinomial is associated with a *multi-index*, meaning a vector $\alpha := (\alpha_1, \alpha_2,\ldots,\alpha_m)$ of nonnegative integers α_s. For each such multi-index, we define the *multinomial function*

$$x^\alpha := \prod_{s=1}^m x_s^{\alpha_s}, \tag{9.2}$$

following the convention that $x_t^0 = 1$ for any position t such that $\alpha_t = 0$.

For discrete random variables in $\mathcal{X}^m := \{0,1,\ldots,r-1\}^m$, it suffices[1] to consider multi-indices such that the maximum degree $\|\alpha\|_\infty := \max_{s=1,\ldots,m} \alpha_s$ is less than or equal to $r-1$. Consequently, our multinomial basis involves a total of r^m multinomial functions. We define the Hamming norm $\|\alpha\|_0 := \operatorname{card}\{i = 1,\ldots,m \mid \alpha_i \neq 0\}$, which counts the number of nonzero elements in the multi-index α. For each integer $k = 1,\ldots,m$, we then define the multi-index set

$$\mathcal{I}_k := \{\alpha \ | \ | \ \|\alpha\|_0 \leq k \}. \tag{9.3}$$

---

[1] Indeed, for any variable $x \in \mathcal{X} = \{0,1,\ldots,r-1\}$, note that there holds $\prod_{j=0}^{r-1}(x-j) = 0$ A simple rearrangement of this relation yields an expression for x^r as a polynomial of degree $r-1$, which implies that any monomial x^i with $i \geq r$ can be expressed as a linear combination of lower-order monomials.

The nested sequence of multi-index sets

$$\mathcal{I}_1 \subset \mathcal{I}_2 \subset \ldots \subset \mathcal{I}_m \tag{9.4}$$

describes a hierarchy of models, which can be associated with hypergraphs with increasing sizes of hyperedges. To calculate the cardinality of $\mathcal{I}_k$, observe that for each $i = 0, \ldots, k$, there are $\binom{m}{i}$ possible subsets of size i. Moreover, for each member of each such subset, there are $(r-1)$ possible choices of the index value, so that $\mathcal{I}_k$ has $\sum_{i=0}^{k} \binom{m}{i} (r-1)^i$ elements in total. The total number of all possible multi-indices (with $\|\alpha\|_\infty \leq r-1$) is given by $|\mathcal{I}_m| = \sum_{i=0}^{m} \binom{m}{i}(r-1)^i = r^m$.

Our second basis is a generalization of the standard overcomplete potentials (3.34), based on indicator functions for events of the form $\{X_s = j\}$ as well as their higher-order analogs. Recall the $\{0,1\}$-valued indicator function $\mathbb{I}_{j_s}(x_s)$ for the event $\{X_s = j_s\}$. Using these node-based functions, we define, for each global configuration $J = (j_1, \ldots, j_m) \in \{0, 1, \ldots, r-1\}^m$, the indicator function

$$\mathbb{I}_J(x) := \prod_{s=1}^{m} \mathbb{I}_{j_s}(x_s) = \begin{cases} 1 & \text{if } (x_1, \ldots, x_m) = (j_1, \ldots, j_m) \\ 0 & \text{otherwise.} \end{cases} \tag{9.5}$$

In our discussion thus far, we have defined these function bases over the full vector $(x_1, \ldots, x_m)$; more generally, we can also consider the same bases for subvectors $(x_1, \ldots, x_k)$. Overall, for any integer $k \leq m$, we have two function bases: the r^k vector of multinomial functions

$$\mathfrak{M}_k(x_1, \ldots, x_k) := \left\{ \prod_{s=1}^{k} x_s^{\alpha_s} \mid \alpha \in \{0, 1, \ldots, r-1\}^k \right\},$$

and the r^k vector of indicator functions

$$\mathfrak{I}_k(x_1, \ldots, x_k) := \left\{ \mathbb{I}_J(x) \mid J \in \{0, 1, \ldots, r-1\}^k \right\},$$

The following elementary lemma shows that these two bases are in one-to-one correspondence:

Lemma 9.2 For each $k = 2, 3, \ldots, m$, there is an invertible $r^k \times r^k$ matrix B such that

$$\mathfrak{M}_k(x) = B \, \mathfrak{I}_k(x) \quad \text{for all } x \in \mathcal{X}^k. \tag{9.6}$$

Proof. Our proof is based on explicitly constructing the matrix B. For each $s = 1, \ldots, k$ and $j \in \{0, 1, 2, \ldots, r - 1\}$, consider the following identities between the scalar indicator functions $\mathbb{I}_j(x_s)$ and monomials x_s^j:

$$\mathbb{I}_j(x_s) = \prod_{\ell \neq j} \frac{x_s - \ell}{j - \ell}, \quad \text{and} \quad x_s^j = \sum_{\ell=0}^{r-1} \ell^j \, \mathbb{I}_\ell(x_s). \tag{9.7}$$

Using the second identity multiple times (once for each $s \in \{1, \ldots, k\}$), we obtain that for each $\alpha \in \mathcal{I}_k$,

$$x^\alpha = \prod_{s=1}^{k} x_s^{\alpha_s} = \prod_{s=1}^{k} \left[\sum_{\ell=0}^{r-1} \ell^{\alpha_s} \, \mathbb{I}_\ell(x_s) \right].$$

By expanding the product on the right-hand side, we obtain an expression for multinomial x^α as a linear combination of the indicator functions $\{\mathbb{I}_J(x), | \; J \in \mathcal{X}^k\}$. Conversely, for each $J \in \{0, 1, \ldots, r - 1\}^k$, we have

$$\mathbb{I}_J(x) := \prod_{s=1}^{k} \mathbb{I}_{j_s}(x_s) = \prod_{s=1}^{k} \left[\prod_{\ell \neq j_s} \frac{x_s - \ell}{j_s - \ell} \right].$$

As before, by expanding the product on the right-hand side, we obtain an expression for the indicator function $\mathbb{I}_J$ as a linear combination of the monomials $\{x^\alpha, \alpha \in \mathcal{I}_k\}$. Thus, there is an invertible linear transformation between the indicator functions $\{\mathbb{I}_J(x), J \in \mathcal{X}^k\}$ and the monomials $\{x^\alpha, \alpha \in \mathcal{I}_k\}$. We let $B \in \mathbb{R}^{r^k \times r^k}$ denote the invertible matrix that carries out this transformation. $\qquad \square$

These bases are convenient for different purposes, and Lemma 9.2 allows us to move freely between them. The mean parameters associated with the indicator basis $\mathfrak{I}$ are readily interpretable as probabilities — viz. $\mathbb{E}[\mathbb{I}_J(X)] = \mathbb{P}[X = J]$. However, the multinomial basis is convenient for defining hierarchies of semidefinite relaxations.

9.1.2   Marginal Polytopes for Hypergraphs

We now turn to the use of these function bases in defining marginal polytopes for general hypergraphs. Given a hypergraph $G = (V, E)$

in which the maximal hyperedge has cardinality k, we may consider the multinomial Markov random field $p_\theta(x) \propto \exp\left\{\sum_{\alpha \in \mathcal{I}(G)} \theta_\alpha x^\alpha\right\}$. Here $\mathcal{I}(G) \subseteq \mathcal{I}_k$ corresponds to those multi-indices associated with the hyperedges of G. For instance, if G includes the triplet hyperedge $\{s, t, u\}$, then $\mathcal{I}(G)$ must include the set of r^3 multi-indices of the form

$$(0,\, 0, \ldots,\, \alpha_s,\, \alpha_t,\, \alpha_u,\, \ldots, 0) \quad \text{for some } (\alpha_s, \alpha_t, \alpha_u) \in \{0, 1, \ldots, r-1\}^3.$$

As an important special case, for each integer $t = 1, 2, \ldots, m$, we also define the multi-index sets $\mathcal{I}_t := \mathcal{I}(K_{m,t})$, where $K_{m,t}$ denotes the hypergraph on m nodes that includes all hypergraphs up to size t. For instance, the hypergraph $K_{m,2}$ is simply the ordinary complete graph on m nodes, including all $\binom{m}{2}$ edges.

Let $\mathcal{P}(\mathcal{X}^m)$ denote the set of all distributions p supported on $\mathcal{X}^m = \{0, 1, \ldots, r-1\}^m$. For any multi-index $\alpha \in \mathbb{Z}_+^m$ and distribution $p \in \mathcal{P}(\mathcal{X}^m)$, let

$$\mu_\alpha := \mathbb{E}_p[X^\alpha] = \mathbb{E}_p\left[\prod_{i=1}^m X_i^{\alpha_i}\right] \tag{9.8}$$

denote the associated mean parameter or moment. (We simply write $\mathbb{E}[X^\alpha]$ when the underlying distribution p is understood from the context.) We use $\mathbb{M}(G)$ to denote the marginal polytope associated with hypergraph G — that is, $\mathbb{M}(G)$ is the set

$$\left\{\mu \in \mathbb{R}^{|\mathcal{I}(G)|} \;\middle|\; \exists\, p \in \mathcal{P}(\mathcal{X}^m) \text{ such that } \mu_\alpha = \mathbb{E}_p[X^\alpha] \quad \forall \alpha \in \mathcal{I}(G)\right\}. \tag{9.9}$$

To be precise, it should be noted that our choice of notation is slightly inconsistent with previous sections, where we defined marginal polytopes in terms of the indicator functions $\mathbb{I}_j(x_s)$ and $\mathbb{I}_j[x_s]\mathbb{I}_k[x_t]$. However, by the one-to-one correspondence between these indicator functions and the multinomials $\{x^\alpha\}$ from Lemma 9.2, the corresponding marginal polytopes are isomorphic objects, regardless of the underlying potential functions chosen.

9.2 Semidefinite Bounds on Marginal Polytopes

We now describe the Lasserre sequence [148, 150] of semidefinite outer bounds on the marginal polytope $\mathbb{M}(G)$. This sequence is defined in

terms of a hierarchy of moment matrices which generate a nested sequence of semidefinite outer bounds on any marginal polytope.

9.2.1   Lasserre Sequence

To define the relevant moment matrices, for each $t = 1, 2, \ldots, m$, consider the $|\mathcal{I}_t| \times |\mathcal{I}_t|$ matrix of moments $N_t[\mu]$ defined by the $|\mathcal{I}_t|$-dimensional random vector $Y := (X^\alpha, \ \alpha \in \mathcal{I}_t)$. Each row and column of $N_t[\mu]$ is associated with some multi-index $\alpha \in \mathcal{I}_t$, and its entries are specified as follows:

$$\left(N_t[\mu]\right)_{\alpha\beta} := \mu_{\alpha+\beta} = \mathbb{E}[X^\alpha X^\beta]. \tag{9.10}$$

As a particular example, Figure 9.1(a) provides an illustration of the matrix $N_3[\mu]$ for the special case of binary variables $(r = 2)$ on three nodes $(m = 3)$, so that the overall matrix is eight-dimensional.

(a)

(b)

Fig. 9.1 Moment matrices and minors defining the Lasserre sequence of semidefinite relaxations for a triplet $(m = 3)$ of binary variables $(r = 2)$. (a) Full matrix $N_3[\mu]$. (b) Shaded region: 7×7 principal minor $N_2[\mu]$ constrained by the Lasserre relaxation at order 1.

The shaded region in Figure 9.1(b) shows the matrix $N_2[\mu]$ for this same example; note that it has $|\mathcal{I}_2| = 7$ rows and columns.

Let us make some clarifying comments regarding these moment matrices. First, so as to simplify notation in this binary case, we have used μ_1 as a shorthand for the first-order moment $\mu_{1,0,0}$, with similar shorthand for the other first-order moments μ_2 and μ_3. Similarly, the quantity μ_{12} is shorthand for the second-order moment $\mu_{1,1,0} = \mathbb{E}[X_1^1 X_2^1 X_3^0]$, and the quantity μ_{123} denotes the triplet moment $\mu_{1,1,1} = \mathbb{E}[X_1 X_2 X_3]$. Second, in both matrices, the element in the upper left-hand corner is

$$\mu_\emptyset \; = \; \mu_{0,0,0} \; = \; \mathbb{E}[X^0],$$

which is always equal to one. Third, in calculating the form of these moment matrices, we have repeatedly used the fact that $X_i^2 = X_i$ for any binary variable to simplify moment calculations. For instance, in computing the $(8,7)$ element of $N_3[\mu]$, we write

$$\mathbb{E}[(X_1 X_2 X_3)(X_1 X_3)] \; = \; \mathbb{E}[X_1 X_2 X_3] \; = \; \mu_{123}.$$

We now describe how the moment matrices $N_t[\mu]$ induce outer bounds on the marginal polytope $\mathbb{M}(G)$. Given a hypergraph G, let $t(G)$ denote the smallest integer such that all moments associated with $\mathbb{M}(G)$ appear in the moment matrix $N_{t(G)}[\mu]$. (For example, for an ordinary graph with maximal cliques of size two, such as a 2D lattice, we have $t(G) = 1$, since all moments of orders one and two appear in the matrix $N_1[\mu]$.) For each integer $t = t(G), \ldots, m$, we can define the mapping $\Pi_G : \mathbb{R}^{|\mathcal{I}_t|} \to \mathbb{R}^{|\mathcal{I}(G)|}$ that maps any vector $\mu \in \mathbb{R}^{\mathcal{I}_t}$ to the indices $\{\alpha \in \mathcal{I}(G)\}$. Then for each $t = t(G), 2, \ldots$, define the semidefinite constraint set

$$\mathbb{S}_t(G) := \Pi_G[\{\mu \in \mathbb{R}^{|\mathcal{I}_t|} \mid N_t[\mu] \succeq 0\}]. \tag{9.11}$$

As a consequence of Lemma 9.1, each set $\mathbb{S}_t(G)$ is an outer bound on the marginal polytope $\mathbb{M}(G)$, and by definition of the matrices $N_t[\mu]$, these outer bounds form a nested sequence

$$\mathbb{S}_1(G) \; \supseteq \; \mathbb{S}_2(G) \; \supseteq \; \mathbb{S}_3(G) \; \supseteq \cdots \supseteq \mathbb{M}(G). \tag{9.12}$$

This sequence is known as the Lasserre sequence of relaxations [147, 150]. Lovász and Schrijver [159] describe a related class of relaxations, also based on semidefinite constraints.

Example 9.1 (First-order Semidefinite Bound on $\mathbb{M}(K_3)$). As an illustration of the power of semidefinite bounds, recall Example 4.3, in which we considered the fully connected graph K_3 on three nodes. In terms of the 6D vector of sufficient statistics $(X_1, X_2, X_3, X_1 X_2, X_2 X_3, X_1 X_3)$, we considered the pseudomarginal vector

$$\begin{pmatrix} \tau_1, & \tau_2, & \tau_3, & \tau_{12}, & \tau_{23}, & \tau_{13} \end{pmatrix} = \begin{pmatrix} 0.5, & 0.5, & 0.5, & 0.4, & 0.4, & 0.1 \end{pmatrix}.$$

In Example 4.3, we used the cycle inequalities, as derived in Example 8.7, to show that τ does *not* belong to $\mathbb{M}(K_3)$. Here we provide an alternative and arguably more direct proof of this fact based on semidefinite constraints. In particular, the moment matrix $N_1[\tau]$ associated with these putative mean parameters has the form:

$$N_1[\tau] = \begin{bmatrix} 1 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.4 & 0.1 \\ 0.5 & 0.4 & 0.5 & 0.4 \\ 0.5 & 0.1 & 0.4 & 0.5 \end{bmatrix}.$$

A simple calculation shows that this matrix $N_1[\tau]$ is not positive definite, whence $\tau \notin \mathbb{S}_1(K_3)$, so that by the inclusions (9.12), we conclude that $\tau \notin \mathbb{M}(K_3)$.

9.2.2   Tightness of Semidefinite Outer Bounds

Given the nested sequence (9.12) of outer bounds on the marginal polytope $\mathbb{M}(G)$, we now turn to a natural question: what is the minimal required order (if any) for which $\mathbb{S}_t(G)$ provides an exact characterization of the marginal polytope? It turns out that for an arbitrary hypergraph, $t = m$ is the minimal required (although see Section 9.3 and Proposition 9.5 for sharper guarantees based on treewidth). This property of finite termination for semidefinite relaxations in a general

setting was proved by Lasserre [147], and also by Laurent [149, 150] using different methods. We provide here one proof of this exactness:

Proposition 9.3 (Tightness of Semidefinite Constraints). For any hypergraph G with m vertices, the semidefinite constraint set $\mathbb{S}_m(G)$ provides an exact description of the associated marginal polytope $\mathbb{M}(G)$.

Proof. Although the result holds more generally [147], we prove it here for the case of binary random variables. The inclusion $\mathbb{M}(G) \subseteq \mathbb{S}_m(G)$ is an immediate consequence of Lemma 9.1, so that it remains to establish the reverse inclusion. In order to do so, it suffices to consider a vector $\tau \in \mathbb{R}^{2^m}$, with elements indexed by indicator vectors $\alpha \in \{0,1\}^m$ of subsets of $\{1,\ldots,m\}$. In particular, element τ_α represents a candidate moment associated with the multinomial $x^\alpha = \prod_{s=1}^m x_s^{\alpha_s}$. Suppose that the matrix $N_m[\tau] \in \mathbb{R}^{2^m \times 2^m}$ is positive semidefinite; we need to show that τ is a valid moment vector, meaning that there is some distribution p supported on $\{0,1\}^m$ such that $\tau_\alpha = \mathbb{E}_p[X^\alpha]$ for all $\alpha \in \mathcal{I}_m$.

Lemma 9.2 shows that the multinomial basis $\mathfrak{M}_m$ and the indicator basis $\mathfrak{I}_m$ are related by an invertible linear transformation $B \in \mathbb{R}^{2^m \times 2^m}$. In the binary case, each multi-index α can be associated with a subset of $\{1, 2, \ldots, m\}$, and we have the partial order $\alpha \subseteq \beta$ defined by inclusion of these subsets. It is straightforward to verify that the matrix B arises from the inclusion-exclusion principle, and so has entries

$$B(\alpha, \beta) = \begin{cases} (-1)^{|\beta \backslash \alpha|} & \text{if } \alpha \subseteq \beta \\ 0 & \text{otherwise.} \end{cases} \tag{9.13}$$

Moreover, the inverse matrix B^{-1} has entries

$$B^{-1}(\alpha, \beta) = \begin{cases} 1 & \text{if } \alpha \subseteq \beta \\ 0 & \text{otherwise.} \end{cases} \tag{9.14}$$

See Appendix E.1 for more details on these matrices, which arise as a special case of the Möbius inversion formula.

In Appendix E.2, we prove that B diagonalizes $N_m[\tau]$, so that we have $B N_m[\tau] B^T = D$ for some diagonal matrix D with a nonnegative

entries, and $\text{trace}(D) = 1$. Thus, we can define a probability distribution p supported on $\{0,1\}^m$ with elements $p(\alpha) = D_\alpha \geq 0$. Since B is invertible, we have $N_m[\tau] = B^{-1}DB^{-T}$. Focusing on the αth diagonal entry, the definition of $N_m[\tau]$ implies that $(N_m[\tau])_{\alpha\alpha} = \tau_\alpha$. Consequently, we have

$$
\begin{aligned}
\tau_\alpha &= \sum_{(\beta,\gamma)\in\{0,1\}^m\times\{0,1\}^m} B^{-1}(\alpha,\beta)DB^{-T}(\gamma,\alpha) \\
&= \sum_{\beta\in\{0,1\}^m} D(\beta,\beta)B^{-1}(\alpha,\beta)B^{-T}(\beta,\alpha),
\end{aligned}
$$

using the fact that D is diagonal. Using the form (9.14) of B^{-1}, we conclude

$$
\tau_\alpha = \sum_{\alpha\subseteq\beta} p(\beta) = \mathbb{E}_p[X^\alpha],
$$

which provides an explicit demonstration of the global realizability of τ_α. □

This result shows that imposing a semidefinite constraint on the largest possible moment matrix $N_m[\mu]$ is sufficient to fully characterize all marginal polytopes. Moreover, the $t = m$ condition in Proposition 9.3 is not only sufficient for exactness, but also necessary in a worst case setting — that is, for any $t < m$, there exists some hypergraph G such that $\mathbb{M}(G)$ is strictly contained within $\mathbb{S}_t(G)$. The following example illustrates both the sufficiency and necessity of Proposition 9.3:

Example 9.2   ((In)exactness of Semidefinite Constraints). Consider a pair of binary random variables $(X_1, X_2) \in \{0,1\}^2$. With respect to the monomials (X_1, X_2, X_1X_2), the marginal polytope $\mathbb{M}(K_2)$ consists of three moments (μ_1, μ_2, μ_{12}). Figure 9.2(a) provides an illustration of this 3D polytope; as derived previously in Example 3.8, this set is characterized by the four constraints

$$
\mu_{12} \geq 0, \quad 1 + \mu_1 + \mu_2 - \mu_{12} \geq 0, \quad \text{and} \quad \mu_s - \mu_{12} \geq 0 \quad \text{for } s = 1,2.
$$
$$\tag{9.15}$$

The first-order semidefinite constraint set $\mathbb{S}_1(K_2)$ is defined by the semidefinite moment matrix constraint

$$N_1[\mu] = \begin{bmatrix} 1 & \mu_1 & \mu_2 \\ \mu_1 & \mu_1 & \mu_{12} \\ \mu_2 & \mu_{12} & \mu_2 \end{bmatrix} \succeq 0. \tag{9.16}$$

In order to deduce the various constraints implied by this semidefinite inequality, we make use of an extension of Sylvester's criterion for assessing positive semidefiniteness: a square matrix is positive semidefinite if and only if the determinant of any principal submatrix is nonnegative (see Horn and Johnson [114], p. 405). So as to facilitate both our calculation and subsequent visualization, it is convenient to focus on the intersection of both the marginal polytope and the constraint set $\mathbb{S}_1(K_2)$ with the hyperplane $\mu_1 = \mu_2$. Stepping through the requirements of Sylvester's criterion, positivity of the $(1,1)$ element is trivial, and the remaining singleton principal submatrices imply that $\mu_1 \geq 0$ and $\mu_2 \geq 0$. With a little calculation, it can be seen the $(1,2)$ and $(1,3)$ principal submatrices yield the interval constraints $\mu_1, \mu_2 \in [0,1]$. The $(2,3)$ principal submatrix yields $\mu_1\mu_2 \geq \mu_{12}^2$, which after setting $\mu_1 = \mu_2$ reduces to $\mu_1 \geq |\mu_{12}|$. Finally, we turn to nonnegativity of the full determinant: after setting $\mu_1 = \mu_2$ and some algebraic manipulation, we obtain the the constraint

$$\mu_{12}^2 - (2\mu_1^2)\mu_{12} + (2\mu_1^3 - \mu_1^2) = [\mu_{12} - \mu_1]\,[\mu_{12} - \mu_1\,(2\mu_1 - 1)] \leq 0.$$

Since $\mu_1 \geq |\mu_{12}|$ from before, this quadratic inequality implies the pair of constraints

$$\mu_{12} \leq \mu_1, \quad \text{and} \quad \mu_{12} \geq \mu_1\,(2\mu_1 - 1). \tag{9.17}$$

The gray area in Figure 9.2(b) shows the intersection of the 3D marginal polytope $\mathbb{M}(K_2)$ from panel (a) with the hyperplane $\mu_1 = \mu_2$. The intersection of the semidefinite constraint set $\mathbb{S}_1(K_2)$ with this same hyperplane is characterized by the interval inclusion $\mu_1 \in [0,1]$ and the two inequalities in Equation (9.17). Note that the semidefinite constraint set is an outer bound on $\mathbb{M}(K_2)$, but that it includes points that are clearly not valid marginals. For instance, it can be verified that $(\mu_1, \mu_2, \mu_{12}) = (\frac{1}{4}, \frac{1}{4}, -\frac{1}{8})$ corresponds to a positive semidefinite $N_1[\mu]$, but this vector certainly does not belong to $\mathbb{M}(K_2)$.
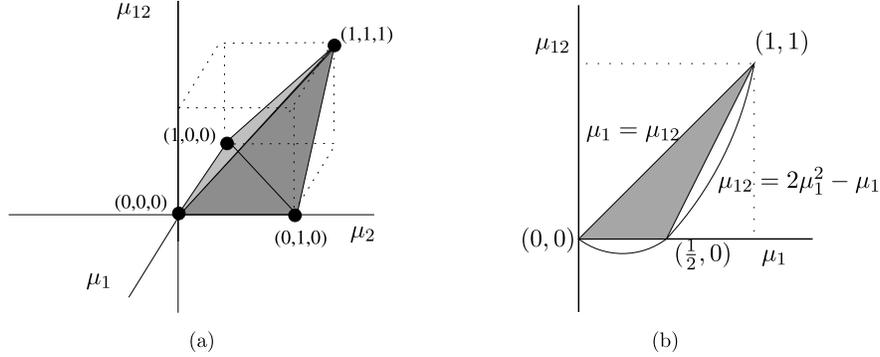
Fig. 9.2 (a) The marginal polytope $\mathbb{M}(K_2)$ for a pair $(X_1, X_2) \in \{0,1\}^2$ of random variables; it is a a polytope with four facets contained within the cube $[0,1]^3$. (b) Nature of the semidefinite outer bound $\mathbb{S}_1$ on the marginal polytope $\mathbb{M}(K_2)$. The gray area shows the cross-section of the binary marginal polytope $\mathbb{M}(K_2)$ corresponding to intersection with the hyperplane $\mu_1 = \mu_2$. The intersection of $\mathbb{S}_1$ with this same hyperplane is defined by the inclusion $\mu_1 \in [0,1]$, the linear constraint $\mu_{12} \leq \mu_1$, and the quadratic constraint $\mu_{12} \geq 2\mu_1^2 - \mu_1$. Consequently, there are points belonging to $\mathbb{S}_1$ that lie strictly outside $\mathbb{M}(K_2)$.

In this case, if we move up one more step to the semidefinite outer bound $\mathbb{S}_2(K_2)$, then Proposition 9.3 guarantees that the description should be exact. To verify this fact, note that $\mathbb{S}_2(K_2)$ is based on imposing positive semidefiniteness of the moment matrix

$$
N_2[\mu] = \begin{bmatrix} 1 & \mu_1 & \mu_2 & \mu_{12} \\ \mu_1 & \mu_1 & \mu_{12} & \mu_{12} \\ \mu_2 & \mu_{12} & \mu_2 & \mu_{12} \\ \mu_{12} & \mu_{12} & \mu_{12} & \mu_{12} \end{bmatrix}.
$$

Positivity of the diagonal element $(4,4)$ gives the constraint $\mu_{12} \geq 0$. The determinant of 2×2 submatrix formed by rows and columns 3 and 4 (referred to as the $(3,4)$ subminor for short) is given by $\mu_{12}[\mu_2 - \mu_{12}]$. Combined with the constraint $\mu_{12} \geq 0$, the nonnegativity of this determinant yields the inequality $\mu_2 - \mu_{12} \geq 0$. By symmetry, the $(2,4)$ subminor gives $\mu_1 - \mu_{12} \geq 0$. Finally, calculating the determinant of $N_2[\mu]$ yields

$$
\det N_2[\mu] = \mu_{12} \left[\mu_1 - \mu_{12} \right] \left[\mu_1 - \mu_{12} \right] \left[1 + \mu_{12} - \mu_1 - \mu_2 \right]. \quad (9.18)
$$

The constraint $\det N_2[\mu] \geq 0$, in conjunction with the previous constraints, implies the inequality $1 + \mu_{12} - \mu_1 - \mu_2 \geq 0$. In fact, the

quantities $\{\mu_{12}, \mu_1 - \mu_{12}, \mu_2 - \mu_{12}, 1 + \mu_{12} - \mu_1 - \mu_2\}$ are the eigenvalues of $N_2[\mu]$, so positive semidefiniteness of $N_2[\mu]$ is equivalent to nonnegativity of these four quantities. The positive semidefiniteness of $\mathbb{S}_2(K_2)$ thus recovers the four inequalities (9.15) that define $\mathbb{M}(K_2)$, thereby providing an explicit confirmation of Proposition 9.3.

From a practical point of view, however, the consequences of Proposition 9.3 are limited, because $N_m[\mu]$ is a $|\mathcal{I}_m| \times |\mathcal{I}_m|$ matrix, where $|\mathcal{I}_m| = r^m$ is exponentially large in the number of variables m. Applications of SDP relaxations are typically based on solving a semidefinite program involving the constraint $N_t[\mu] \succeq 0$ for some t substantially smaller than the total number m of variables. We illustrate with a continuation of the MAX-CUT problem, introduced earlier in Section 8.4.4.

Example 9.3 (First-order SDP relaxation of MAX-CUT). As described previously in Example 8.4, the MAX-CUT problem arises from a graph-partitioning problem, and is computationally intractable (NP-complete) for general graphs $G = (V, E)$. Here, we describe a natural semidefinite programming relaxation obtained by applying the first-order semidefinite relaxation $\mathbb{S}_1(G)$. In the terminology of graphical models, the MAX-CUT problem is equivalent to computing the mode of a binary pairwise Markov random field (i.e., an Ising model) with a very specific choice of interactions. More specifically, it corresponds to solving the binary quadratic program

$$\max_{x \in \{0,1\}^m} \sum_{(s,t) \in E} w_{st}\{x_s x_t + (1 - x_s)(1 - x_t)\}, \qquad (9.19)$$

where $w_{st} \geq 0$ is a weight associated with each edge (s,t) of the graph G. By Theorem 8.1, this integer program can be represented as an equivalent linear program. In particular, introducing the mean parameters $\mu_s = \mathbb{E}[X_s]$ for each node $s \in V$, and $\mu_{st} = \mathbb{E}[X_s X_t]$ for each edge $(s,t) \in E$, we can rewrite the binary quadratic program (9.19) as

$$\max_{\mu \in \mathbb{M}(G)} \sum_{(s,t) \in E} w_{st}\{2\mu_{st} + 1 - \mu_s - \mu_t\},$$

where $\mathbb{M}(G)$ is the marginal polytope associated with the ordinary graph. Although originally described in somewhat different terms, the

relaxation proposed by Goemans and Williamson [98] amounts to replacing the marginal polytope $\mathbb{M}(G)$ with the first-order semidefinite outer bound $\mathbb{S}_1(G)$, thereby obtaining a semidefinite programming relaxation. Solving this SDP relaxation yields a moment matrix, which can be interpreted as the covariance matrix of a Gaussian random vector in m-dimensions. Sampling from this Gaussian distribution and then randomly rounding the resulting m-dimensions yields a randomized algorithm for approximating the maximum cut. Remarkably, Goemans and Williamson [98] show that in the worst case, the expected cut value obtained in this way is at most ≈ 0.878 less than the optimal cut value. Related work by Nesterov [181] provides similar worst-case guarantees for applications of the first-order SDP relaxation to general classes of binary quadratic programs.

In addition to the MAX-CUT problem, semidefinite programming relaxations based on the sequence $\mathbb{S}_t(G), t = 1, 2, \ldots$ have been studied for various other classes of combinatorial optimization problems, including graph coloring, graph partitioning and satisfiability problems. Determining when the additional computation required to impose higher-order semidefinite constraints does (or does not) yield more accurate approximations is an active and lively area of research.

9.3   Link to LP Relaxations and Graphical Structure

Recall from our previous discussion that the complexity of a given marginal polytope $\mathbb{M}(G)$ depends very strongly on the structure of the (hyper)graph G. One consequence of the junction tree theorem is that marginal polytopes associated with hypertrees are straightforward to characterize (see Propositions 2.1 and 8.6). This simplicity is also apparent in the context of semidefinite characterizations. In fact, the LP relaxations discussed in Section 8.5, which are tight for hypertrees of appropriate widths, can be obtained by imposing semidefinite constraints on particular minors of the overall moment matrix $N_m[\mu]$, as we illustrate here.

Given a hypergraph $G = (V, E)$ and the associated subset of multi-indices $\mathcal{I}(G)$, let $\mathbb{M}(G)$ be the $|\mathcal{I}(G)|$-dimensional marginal polytope

defined by the sufficient statistics $(x^\alpha, \alpha \in \mathcal{I}(G))$. For each hyperedge $h \in E$, let $\mathcal{I}(h)$ be the subset of multi-indices with nonzero components only in the elements of h, and let $\mathbb{M}(\{h\})$ be the $|\mathcal{I}(h)|$-dimensional marginal polytope associated with the subvector of multinomials $(x^\alpha, \alpha \in \mathcal{I}(h))$. Letting the maximal hyperedge have cardinality $t + 1$, we can then rewrite the hypertree-based LP relaxation from Section 8.5 in the form

$$\mathbb{L}_t(G) = \bigcap_{h \in E} \{\mu \in \mathbb{R}^{|\mathcal{I}(G)|} \mid \Pi_h(\mu) \in \mathbb{M}(\{h\})\}, \qquad (9.20)$$

where $\Pi_h : \mathbb{R}^{|\mathcal{I}(G)|} \to \mathbb{R}^{|\mathcal{I}(h)|}$ is the standard coordinate projection.

In general, imposing semidefiniteness on moment matrices produces constraint sets that are convex but nonpolyhedral (i.e., with curved boundaries; see Figure 9.2(b) for an illustration). In certain cases, however, a semidefinite constraint actually reduces to a set of linear inequalities, so that the associated constraint set is a polytope. We have already seen one instance of this phenomenon in Proposition 9.3, where by a basis transformation (between the indicator functions $\mathbb{I}_j(x_s)$ and the monomials $x_s^{\alpha_s}$), we showed that a semidefinite constraint on the full matrix $N_m[\mu]$ is equivalent to a total of r^m linear inequalities.

In similar fashion, it turns out that the LP relaxation $\mathbb{L}_t(G)$ can be described in terms of semidefiniteness constraints on a subset of minors from the full moment matrix $N_m[\mu]$. In particular, for each hyperedge $h \in E$, let $N_{\{h\}}[\mu]$ denote the $|\mathcal{I}(h)| \times |\mathcal{I}(h)|$ minor of $N_m[\mu]$, and define the semidefinite constraint set

$$\mathbb{S}(\{h\}) = \{\mu \in \mathbb{R}^{|\mathcal{I}_m|} \mid N_{\{h\}}[\mu] \succeq 0\}. \qquad (9.21)$$

As a special case of Proposition 9.3 for $m = |h|$, we conclude that the projection of this constraint set down to the mean parameters indexed by $\mathcal{I}(h)$ — namely, the set $\Pi_h(\mathbb{S}(\{h\}))$ — is equal to the local marginal polytope $\mathbb{M}(\{h\})$. We have thus established the following:

Proposition 9.4 Given a hypergraph G with maximal hyperedge size $|h| = t + 1$, an equivalent representation of the polytope $\mathbb{L}_t(G)$ is in

terms of the family of hyperedge-based semidefinite constraints

$$\mathbb{L}_t(G) = \bigcap_{h \in E} \mathbb{S}(\{h\}), \tag{9.22}$$

obtained by imposing semidefiniteness on certain minors of the full moment matrix $N_m[\mu]$. This relaxation is tight when G is a hypertree of width t.

Figure 9.3 provides an illustration of the particular minors that are constrained by the relaxation $\mathbb{L}_t(G)$ in the special case of the single cycle K_3 on $m = 3$ nodes with binary variables (see Figure 8.1). Each panel in Figure 9.3 shows the full 8×8 moment matrix $N_3[\mu]$ associated with this graph; the shaded portions in each panel demarcate the minors $N_{\{12\}}[\mu]$ and $N_{\{13\}}[\mu]$ constrained in the first-order

$$\begin{bmatrix}
1 & \mu_1 & \mu_2 & \mu_3 & \mu_{12} & \mu_{23} & \mu_{13} & \mu_{123} \\
\mu_1 & \mu_1 & \mu_{12} & \mu_{13} & \mu_{12} & \mu_{123} & \mu_{13} & \mu_{123} \\
\mu_2 & \mu_{12} & \mu_2 & \mu_{23} & \mu_{12} & \mu_{23} & \mu_{123} & \mu_{123} \\
\mu_3 & \mu_{13} & \mu_{23} & \mu_3 & \mu_{123} & \mu_{23} & \mu_{13} & \mu_{123} \\
\mu_{12} & \mu_{12} & \mu_{12} & \mu_{123} & \mu_{12} & \mu_{123} & \mu_{123} & \mu_{123} \\
\mu_{23} & \mu_{123} & \mu_{23} & \mu_{23} & \mu_{123} & \mu_{23} & \mu_{123} & \mu_{123} \\
\mu_{13} & \mu_{13} & \mu_{123} & \mu_{13} & \mu_{123} & \mu_{123} & \mu_{13} & \mu_{123} \\
\mu_{123} & \mu_{123} & \mu_{123} & \mu_{123} & \mu_{123} & \mu_{123} & \mu_{123} & \mu_{123}
\end{bmatrix}$$

(a)

$$\begin{bmatrix}
1 & \mu_1 & \mu_2 & \mu_3 & \mu_{12} & \mu_{23} & \mu_{13} & \mu_{123} \\
\mu_1 & \mu_1 & \mu_{12} & \mu_{13} & \mu_{12} & \mu_{123} & \mu_{13} & \mu_{123} \\
\mu_2 & \mu_{12} & \mu_2 & \mu_{23} & \mu_{12} & \mu_{23} & \mu_{123} & \mu_{123} \\
\mu_3 & \mu_{13} & \mu_{23} & \mu_3 & \mu_{123} & \mu_{23} & \mu_{13} & \mu_{123} \\
\mu_{12} & \mu_{12} & \mu_{12} & \mu_{123} & \mu_{12} & \mu_{123} & \mu_{123} & \mu_{123} \\
\mu_{23} & \mu_{123} & \mu_{23} & \mu_{23} & \mu_{123} & \mu_{23} & \mu_{123} & \mu_{123} \\
\mu_{13} & \mu_{13} & \mu_{123} & \mu_{13} & \mu_{123} & \mu_{123} & \mu_{13} & \mu_{123} \\
\mu_{123} & \mu_{123} & \mu_{123} & \mu_{123} & \mu_{123} & \mu_{123} & \mu_{123} & \mu_{123}
\end{bmatrix}$$

(b)

Fig. 9.3 Shaded regions correspond to the 4×4 minors $N_{\{12\}}[\mu]$ in panel (a), and $N_{\{13\}}[\mu]$ in panel (b) are constrained by the Sherali–Adams relaxation of order 2. Also constrained is the minor $N_{\{23\}}[\mu]$ (not shown).

LP relaxation $\mathbb{L}(G)$. In addition, this LP relaxation also constrains the minor $N_{\{23\}}[\mu]$, not shown here.

Thus, the framework of moment matrices allows us to understand both the hierarchy of LP relaxations defined by the hypertree-based polytopes $\mathbb{L}_t(G)$, as well as the SDP hierarchy based on the nonpolyhedral sets $\mathbb{S}_t(G)$. It is worth noting that at least for general graphs with $m \geq 3$ vertices, this hierarchy of LP and SDP relaxations are mutually incomparable, in that neither one dominates the other at a fixed level of the hierarchy. We illustrate this mutual incomparability in the following example:

Example 9.4 (Mutual Incomparability of LP/SDP Relaxations). For the single cycle K_3 on $m = 3$ vertices, neither the first-order LP relaxation $\mathbb{L}_1(G)$ nor the first-order semidefinite relaxation $\mathbb{S}_1(K_3)$ are exact. At the same time, these two relaxations are mutually incomparable, in that neither dominates the other. In one direction, Example 9.1 provides an instance of a pseudomarginal vector τ that belongs to $\mathbb{L}_1(K_3)$, but violates the semidefinite constraint defining $\mathbb{S}_1(K_3)$. In the other direction, we need to construct a pseudomarginal vector τ that satisfies the semidefinite constraint $N_1[\tau] \succeq 0$, but violates at least one of the linear inequalities defining $\mathbb{L}_1(K_3)$. Consider the pseudomarginal τ with moment matrix $N_1[\tau]$ of the form:

$$
N_1[\tau] = \begin{bmatrix} 1 & \tau_1 & \tau_2 & \tau_3 \\ \tau_1 & \tau_1 & \tau_{12} & \tau_{13} \\ \tau_2 & \tau_{12} & \tau_2 & \tau_{23} \\ \tau_3 & \tau_{13} & \tau_{23} & \tau_3 \end{bmatrix} = \begin{bmatrix} 1 & 0.75 & 0.75 & 0.75 \\ 0.75 & 0.75 & 0.45 & 0.5 \\ 0.75 & 0.45 & 0.75 & 0.5 \\ 0.75 & 0.5 & 0.5 & 0.75 \end{bmatrix}.
$$

Calculation shows that $N_1[\tau]$ is positive definite, yet the inequality $1 + \tau_{12} - \tau_1 - \tau_2 \geq 0$ from the conditions (9.15), necessary for membership in $\mathbb{L}_1(K_3)$, is not satisfied. Hence, the constructed vector τ belongs to $\mathbb{S}_1(K_3)$ but does not belong to $\mathbb{L}_1(K_3)$.

Of course, for hypergraphs of low treewidth, the LP relaxation is definitely advantageous, in that it is tight once the order of relaxation t hits the hypergraph treewidth — that is, the equality $\mathbb{L}_t(G) = \mathbb{M}(G)$ holds for any hypergraph of treewidth t, as shown in Proposition 8.6.

In contrast, the semidefinite relaxation $\mathbb{S}_t(G)$ is not tight for a general hypergraph of width t; as concrete examples, see Example 9.2 for the failure of $\mathbb{S}_1(G)$ for the graph $G = K_2$, which has treewidth $t = 1$, and Example 9.4 for the failure of $\mathbb{S}_2(G)$ for the graph $G = K_3$, which has treewidth $t = 2$. However, if the order of semidefinite relaxation is raised to *one order higher* than the treewidth, then the semidefinite relaxation is tight:

Proposition 9.5 For a random vector $X \in \{0, 1, \dots, r-1\}^m$ that is Markov with respect to a hypergraph G, we always have

$$\mathbb{S}_{t+1}(G) \subseteq \mathbb{L}_t(G),$$

where the inclusion is strict unless G is a hypertree of width t. For any such hypertree, the equality $\mathbb{S}_{t+1}(G) = \mathbb{M}(G)$ holds.

Proof. Each maximal hyperedge h in a hypergraph G with treewidth t has cardinality $|h| = t + 1$. Note that the moment matrix $N_{t+1}[\mu]$ constrained by the relaxation $\mathbb{S}_{t+1}(G)$ includes as minors the $|\mathcal{I}(h)| \times |\mathcal{I}(h)|$ matrix $N_{\{h\}}[\mu]$, for every hyperedge $h \in E$. This fact implies that $\mathbb{S}_{t+1}(G) \subseteq \mathbb{S}(\{h\})$ for each hyperedge $h \in E$, where the set $\mathbb{S}(\{h\})$ was defined in Equation (9.21). Hence, the asserted inclusion follows from Proposition 9.4. For hypergraphs of treewidth t, the equality $\mathbb{S}_{t+1}(G) = \mathbb{M}(G)$ follows from this inclusion, and the equivalence between $\mathbb{L}_t(G)$ and $\mathbb{M}(G)$ for hypergraphs of width t, as asserted in Proposition 8.6.   □

9.4   Second-order Cone Relaxations

In addition to the LP and SDP relaxations discussed thus far, another class of constraints on marginal polytopes are based on the second-order cone (see Appendix A.2). The resulting second-order cone programming (SOCP) relaxations of the mode-finding problem as well as of general nonconvex optimization problems have been studied by various researchers [134, 146]. In the framework of moment matrices, second-order cone constraints can be understood as a particular weakening of a positive semidefiniteness constraint, as we describe here.

The semidefinite relaxations that we have developed are based on enforcing that certain moment matrices are positive semidefinite. All of these constraints can be expressed in terms of moment matrices of the form

$$N[\mu] := \mathbb{E}\left[\begin{bmatrix}1\\Y\end{bmatrix}\begin{bmatrix}1 & Y\end{bmatrix}\right], \tag{9.23}$$

where the random vector $Y = f(X_1,\ldots,X_m)$ is a suitable function of the vector X. The class of SOCP relaxations are based on the following equivalence:

Lemma 9.6 For any moment matrix (9.23), the positive semidefiniteness constraint $N[\mu] \succeq 0$ holds if and only if

$$\langle\!\langle U^T U, \mathbb{E}[YY^T]\rangle\!\rangle \geq \|U\mathbb{E}[Y]\|^2 \quad \text{for all } U \in \mathbb{R}^{d\times d}. \tag{9.24}$$

Proof. Using the Schur complement formula [114], a little calculation shows that the condition $N[\mu] \succeq 0$ is equivalent to requiring that the covariance matrix $\Sigma := \mathbb{E}[YY^T] - \mathbb{E}[Y]\mathbb{E}[Y]$ be positive semidefinite. A symmetric matrix Σ is positive semidefinite if and only if its Frobenius inner product $\langle\!\langle \Sigma, \Lambda \rangle\!\rangle := \text{trace}(\Sigma\Lambda)$ with any other positive semidefinite matrix $\Lambda \in \mathcal{S}_+^d$ is nonnegative [114]. (This fact corresponds to the *self-duality* of the positive semidefinite cone [39].) By the singular value decomposition [114], any $\Lambda \in \mathcal{S}_+^d$ can be written[2] as $\Lambda = U^T U$ for some matrix $U \in \mathbb{R}^{d\times d}$, so that the constraint $\langle\!\langle \Sigma, \Lambda \rangle\!\rangle \geq 0$ can be rewritten as $\langle\!\langle U^T U, \mathbb{E}[YY^T]\rangle\!\rangle \geq \|U\mathbb{E}[Y]\|^2$, as claimed.　□

An inequality of the form (9.24) corresponds to a *second-order cone constraint* on the elements of the moment matrix $N[\mu]$; see Appendix A.2 for background on this cone. The second-order cone programming approach is based on imposing the constraint (9.24) only for a subset of matrices U, so that SOCPs are weaker than the associated SDP relaxation. The benefit of this weakening is that many SOCPs

---

[2] We can obtain rank-deficient Λ by allowing U to have some zero rows.

can be solved with lower computational complexity than semidefinite programs [39]. Kim and Kojima [134] study SOCP relaxations for fairly general classes of nonconvex quadratic programming problems. Kumar et al. [146] study the use of SOCPs for mode-finding in pairwise Markov random fields, in particular using matrices U that are locally defined on the edges of the graph, as well as additional linear inequalities, such as the cycle inequalities in the binary case (see Example 8.7). In later work, Kumar et al. [145] showed that one form of their SOCP relaxation is equivalent to a form of the quadratic programming (QP) relaxation proposed by Lafferty and Ravikumar [198]. Kumar et al. [145] also provide a cautionary message, by demonstrating that certain classes of SOCP constraints fail to improve upon the first-order tree-based LP relaxation (8.5). An example of such redundant SOCP constraints are those in which the matrix $U^T U$ has nonzero entries only in pairs of elements, corresponding to a single edge, or more generally, is associated with a subtree of the original graph on which the MRF is defined. This result can be established using moment matrices by the following reasoning: any SOC constraint (9.24) that constrains only elements associated with some subtree of the graph is redundant with the first-order LP constraints (8.5), since the LP constraints ensure that any moment vector μ is consistent on any subtree embedded within the graph.

Overall, we have seen that a wide range of conic programming relaxations, including linear programming, second-order cone programming, and semidefinite programming, can be understood in a unified manner in terms of multinomial moment matrices. An active area of current research is devoted to understanding the properties of these types of relaxations when applied to particular classes of graphical models.

10

Discussion

The core of this survey is a general set of variational principles for the problems of computing marginal probabilities and modes, applicable to multivariate statistical models in the exponential family. A fundamental object underlying these optimization problems is the set of realizable mean parameters associated with the exponential family; indeed, the structure of this set largely determines whether or not the associated variational problem can be solved exactly in an efficient manner. Moreover, a large class of well-known algorithms for both exact and approximate inference — including belief propagation or the sum-product algorithm, expectation propagation, generalized belief propagation, mean field methods, the max-product algorithm and linear programming relaxations, as well as generalizations thereof — can be derived and understood as methods for solving various forms, either exact or approximate, of these variational problems. The variational perspective also suggests convex relaxations of the exact principle, and so is a fertile source of new approximation algorithms.

Many of the algorithms described in this survey are already standard tools in various application domains, as described in more detail in Section 2.4. While such empirical successes underscore the promise

of variational approaches, a variety of theoretical questions remain to be addressed. One important direction to pursue is obtaining *a priori* guarantees on the accuracy of a given variational method for particular subclasses of problems. Past and ongoing work has studied the performance of linear programming relaxations of combinatorial optimization problems (e.g., [135, 236, 78, 48]), many of them particular cases of the first-order LP relaxation (8.5). It would also be interesting to study higher-order analogs of the sequence of LP relaxations described in Section 8. For certain special classes of graphical models, particularly those based on "locally tree-like" graphs, recent and ongoing work (e.g., [7, 14, 169, 200]) has provided some encouraging results on the accuracy of the sum-product algorithm in computing approximations of the marginal distributions and the cumulant function. Similar questions apply to other types of more sophisticated variational methods, including the various extensions of sum-product (e.g., Kikuchi methods or expectation-propagation) discussed in this survey.

We have focused in this survey on regular exponential families where the parameters lie in an open convex set. This class covers a broad class of graphical models, particularly undirected graphical models, where clique potentials often factor into products over parameters and sufficient statistics. However, there are also examples in which nonlinear constraints are imposed on the parameters in a graphical model. Such constraints, which often arise in the setting of directed graphical models, require the general machinery of *curved exponential families* [74]. Although there have been specialized examples of variational methods applied to such families [122], there does not yet exist a general treatment of variational methods for curved exponential families.

Another direction that requires further exploration is the study of variational inference for distributions outside of the exponential family. In particular, it is of interest to develop variational methods for the stochastic processes that underlie nonparametric Bayesian modeling. Again, special cases of variational inference have been presented for such models — see in particular the work of Blei and Jordan [31] on variational inference for Dirichlet processes — but there is as of yet no general framework.

Still more broadly, there are many general issues in statistical inference that remain to be investigated from a variational perspective. In particular, the treatment of multivariate point estimation and interval estimation in frequentist statistical inference often requires treating nuisance variables differently from parameters. Moreover, it is often necessary to integrate over some variables (e.g., random effects) while maximizing over others. Our presentation has focused on treating marginalization separately from maximization; further research will be needed to find ways to best combine these operations. Also, although marginalization is natural from a Bayesian perspective, our focus on the exponential family is limiting from this perspective, where it is relatively rare for the joint probability distribution of all random variables (data, latent variables and parameters) to follow an exponential family distribution. Further work on finding variational approximations for non-exponential-family contributions to joint distributions, including noninformative and robust prior distributions, will be needed for variational inference methods to make further inroads into Bayesian inference.

Finally, it should be emphasized that the variational approach provides a set of techniques that are complementary to Monte Carlo methods. One important program of research, then, is to characterize the classes of problems for which variational methods (or conversely, Monte Carlo methods) are best suited, and moreover to develop a theoretical characterization of the trade-offs in complexity versus accuracy inherent to each method. Indeed, there are various emerging connections between the mixing properties of Markov chains for sampling from graphical models, and the contraction or correlation decay conditions that are sufficient to ensure convergence of the sum-product algorithm.

Acknowledgments

A

Background Material

In this appendix, we provide some basic background on graph theory, and convex sets and functions.

A.1 Background on Graphs and Hypergraphs

Here, we collect some basic definitions and results on graphs and hypergraphs; see the standard books [17, 34, 35] for further background. A graph $G = (V, E)$ consists of a set $V = \{1, 2, \ldots, m\}$ of vertices, and a set $E \subset V \times V$ of edges. By definition, a graph does not contain self-loops (i.e., $(s, s) \notin E$ for all vertices $s \in V$), nor does it contain multiple copies of the same edge. (These features are permitted only in the extended notion of a multigraph.) For a *directed graph*, the ordering of the edge matters, meaning that (s, t) is distinct from (t, s), whereas for an *undirected graph*, the quantities (s, t) and (t, s) refer to the same edge. A *subgraph* F of a given graph $G = (V, E)$ is a graph $(V(F), E(F))$ such that $V(F) \subseteq V$ and $E(F) \subseteq E$. Given a subset $S \subseteq V$ of the vertex set of a graph $G = (V, E)$, the *vertex-induced subgraph* is the subgraph $F[S] = (S, E(S))$ with edge-set $E(S) := \{(s, t) \in E \mid (s, t) \in E\}$. Given a subset of edges $E' \subseteq E$, the *edge-induced subgraph* is $F(E') = (V(E'), E')$, where $V(E') = \{s \in V \mid (s, t) \in E'\}$.

A *path* P is a graph $P = (V(P), E(P))$ with vertex set $V(P) := \{v_0, v_1, v_2, \ldots, v_k\}$, and edge set

$$E(P) := \{(v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k)\}.$$

We say that the path P joins vertex v_0 to vertex v_k. Of particular interest are paths that are subgraphs of a given graph $G = (V, E)$, meaning that $V(P) \subseteq V$ and $E(P) \subseteq E$. A *cycle* is a graph $C = (V(C), E(C))$ with $V(C) = \{v_0, v_1, \ldots, v_k\}$ with $k \geq 2$ and edge set $E(C) = \{(v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k), (v_k, v_0)\}$. An undirected graph is *acyclic* if it contains no cycles. A graph is *bipartite* if its vertex set can be partitioned as a disjoint union $V = V_A \cup V_B$ (where $\cup$ denotes disjoint union), such that $(s, t) \in E$ implies that $s \in V_A$ and $t \in V_B$ (or vice versa).

A *clique* of a graph is a subset S of vertices that are all joined by edges (i.e., $(s, t) \in E$ for all $s, t \in S$). A *chord* of a cycle C on the vertex set $V(C) = \{v_0, v_1, \ldots, v_k\}$ is an edge (v_i, v_j) that is *not* part of the cycle edge set $E(C) = \{(v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k), (v_k, v_0)\}$. A cycle C of length four or greater contained within a larger graph G is *chordless* if the graph's edge set contains no chords for the cycle C. A graph is *triangulated* if it contains no cycles of length four or greater that are chordless. A *connected component* of a graph is a subset of vertices $S \subseteq V$ such that for all $s, t \in S$, there exists a path contained within the graph G joining s to t. We say that a graph G is *singly connected*, or simply *connected*, if it consists of a single connected component. A *tree* is an acyclic graph with a single connected component; it can be shown by induction that any tree on m vertices must have $m - 1$ edges. More generally, a *forest* is an acyclic graph consisting of one or more connected components.

A *hypergraph* is $G = (V, E)$ is a natural generalization of a graph: it consists of a vertex set $V = \{1, 2, \ldots, m\}$, and a set E of *hyperedges*, which each hyperedge $h \in E$ is a particular subset of V. (Thus, an ordinary graph is the special case in which $|h| = 2$ for each hyperedge.) The *factor graph* associated with a hypergraph G is a bipartite graph (V', E') with vertex set V' corresponding to the union $V \cup E$ of the hyperedge vertex set V, and the set of hyperedges E, and an edge set

E' that includes the edge (s, h), where $s \in V$ and $h \in E$, if and only if the hyperedge h includes vertex s.

A.2   Basics of Convex Sets and Functions

Here, we collect some basic definitions and results on convex sets and functions. Rockafellar [203] is a standard reference on convex analysis; see also the books by Hiriart-Urruty and Lemaréchal [112, 113], Boyd and Vandenberghe [39], and Bertsekas [23].

A.2.1   Convex Sets and Cones

A set $C \subseteq \mathbb{R}^d$ is *convex* if for all $x, y \in C$ and $\alpha \in [0,1]$, the set C also contains the point $\alpha x + (1 - \alpha)y$. Equivalently, the set C must contain the entire line segment joining any two of its elements. Note that convexity is preserved by various operations on sets:

- The *intersection* $\cap_{j \in \mathcal{J}} C_j$ of a family $\{C_j\}_{j \in \mathcal{J}}$ of convex sets remains convex.
- The *Cartesian product* $C_1 \otimes C_2 \otimes \cdots \otimes C_k$ of a family of convex sets is convex.
- The image $f(C)$ of a convex set C under any affine mapping $f(x) = Ax + b$ is a convex set.
- The *closure* $\overline{C}$ and *interior* C° of a convex set C are also convex sets.

Note that the union of convex sets is not convex, in general.

A *cone* K is a set such that for any $x \in K$, the ray $\{\lambda x \mid \lambda > 0\}$ also belongs to K. A *convex cone* is a cone that is also convex. (To appreciate the distinction, the union of two distinct rays is a cone, but not convex in general.) Important examples of convex cones include:

- A *subspace* $\{y \in \mathbb{R}^d \mid Ay = 0\}$ for some matrix $A \in \mathbb{R}^{m \times d}$.
- The *nonnegative orthant*

$$\{y \in \mathbb{R}^d \mid y_1 \geq 0, y_2 \geq 0, \ldots, y_d \geq 0\}.$$

- The *conical hull* of a collection of vectors $\{x_1, \ldots, x_n\}$:

$$\left\{ y \in \mathbb{R}^d \mid y = \sum_{i=1}^{n} \lambda_i x_i \quad \text{with } \lambda_i \geq 0, \ i = 1, \ldots, n \right\}. \quad \text{(A.1)}$$

- The *second-order cone* $\{(y, t) \in \mathbb{R}^d \times \mathbb{R} \mid \|y\|_2 \leq t\}$.
- The cone $\mathcal{S}_+^d$ of *symmetric positive semidefinite matrices*

$$\mathcal{S}_+^d = \{X \in \mathbb{R}^{d \times d} \mid X = X^T, \ X \succeq 0\}. \quad \text{(A.2)}$$

A.2.2   Convex and Affine Hulls

A *linear combination* of elements $x_1, x_2, \ldots, x_k$ from a set S is a sum $\sum_{i=1}^{k} \alpha_i x_i$, for arbitrary scalars $\alpha_i \in \mathbb{R}$. An *affine combination* is a linear combination with the further restriction that $\sum_{i=1}^{k} \alpha_i = 1$, whereas a *convex combination* is a linear combination with the further restrictions $\sum_{i=1}^{k} \alpha_i = 1$ and $\alpha_i \geq 0$ for all $i = 1, \ldots k$. The *affine hull* of a set S, denoted by $\text{aff}(S)$, is the smallest set that contains all affine combinations. Similarly, the *convex hull* of a set S, denoted by $\text{conv}(S)$, is the smallest set that contains all its convex combinations. Note that $\text{conv}(S)$ is a convex set, by definition.

A.2.3   Affine Hulls and Relative Interior

For any $\epsilon > 0$ and vector $z \in \mathbb{R}^d$, define the Euclidean ball

$$\mathbb{B}_\epsilon(z) := \{y \in \mathbb{R}^d \mid \|y - z\| < \epsilon\}. \quad \text{(A.3)}$$

The *interior* of a convex set $C \subseteq \mathbb{R}^d$, denoted by C°, is given by

$$C^\circ := \{z \in C \mid \exists \, \epsilon > 0 \text{ s.t. } \mathbb{B}_\epsilon(z) \subset C\}. \quad \text{(A.4)}$$

The relative interior is defined similarly, except that the interior is taken with respect to the affine hull of C, denoted $\text{aff}(C)$. More formally, the *relative interior* of C, denoted $\text{ri}(C)$, is given by

$$\text{ri}(C) := \{z \in C \mid \exists \epsilon > 0 \text{ s.t. } \mathbb{B}_\epsilon(z) \cap \text{aff}(C) \subset C\}. \quad \text{(A.5)}$$

To illustrate the distinction, note that the interior of the convex set $[0, 1]$, when viewed as a subset of $\mathbb{R}^2$, is empty. In contrast, the affine

hull of $[0, 1]$ is the real line, so that the relative interior of $[0, 1]$ is the open interval $(0, 1)$.

A key property of any convex set C is that its relative interior is always nonempty [203]. A convex set $C \subseteq \mathbb{R}^d$ is *full-dimensional* if its affine hull is equal to $\mathbb{R}^d$. For instance, the interval $[0, 1]$, when viewed as a subset of $\mathbb{R}^2$, is not full-dimensional, since its affine hull is only the real line. For a full-dimensional convex set, the notion of interior and relative interior coincide.

A.2.4 Polyhedra and Their Representations

A *polyhedron* P is a set that can be represented as the intersection of a finite number of half-spaces

$$P = \{x \in \mathbb{R}^d \mid \langle a_j, x \rangle \leq b_j \quad \forall j \in \mathcal{J}\}, \qquad (A.6)$$

where for each $j \in \mathcal{J}$, the pair $(a_j, b_j) \in \mathbb{R}^d \times \mathbb{R}$ parameterizes a particular half-space.

We refer to a bounded polyhedron as a *polytope*. Given a polyhedron P, we say that $x \in P$ is an *extreme point* if it is not possible to write $x = \lambda y + (1 - \lambda)z$ for some $\lambda \in (0, 1)$ and $y, z \in P$. We say that x is a *vertex* if there exists some $c \in \mathbb{R}^d$ such that $\langle c, x \rangle > \langle c, y \rangle$ for all $y \in P$ with $y \neq x$. For a polyhedron, x is a vertex if and only if it is an extreme point. A consequence of the Minkowski–Weyl theorem is that any nonempty polytope can be written as the convex hull of its extreme points. This convex hull representation is dual to the half-space representation. Conversely, the convex hull of any finite collection of vectors is a polytope, and so has a half-space representation (A.6).

A.2.5 Convex Functions

It is convenient to allow convex functions to take the value $+\infty$, particularly in performing dual calculations. More formally, an *extended real-valued function* f on $\mathbb{R}^d$ takes values in the extended reals $\mathbb{R}_* := \mathbb{R} \cup \{+\infty\}$. A function $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is *convex* if for all $x, y \in \mathbb{R}^d$ and $\alpha \in [0, 1]$, we have

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y). \qquad (A.7)$$

The function is *strictly convex* if the inequality (A.7) holds strictly for all $x \neq y$ and $\alpha \in (0,1)$.

The *domain* of a extended real-valued convex function f is the set

$$\text{dom}(f) := \{x \in \mathbb{R}^d \mid f(x) < +\infty\}. \tag{A.8}$$

Note that the domain is always a convex subset of $\mathbb{R}^d$. Throughout, we restrict our attention to *proper* convex functions, meaning that $\text{dom}(f)$ is nonempty.

A key consequence of the definition (A.7) is *Jensen's inequality*, which says that for any convex combination $\sum_{i=1}^{k} \alpha_i x_i$ of points $\{x_i\}$ in the domain of f, we have $f\left(\sum_{i=1}^{k} \alpha_i x_i\right) \leq \sum_{i=1}^{k} \alpha_i f(x_i)$.

Convexity of functions is preserved by various operations. In particular, given a collection of convex functions $\{f_j \ j \in \mathcal{J}\}$:

- any linear combination $\sum_{j \in \mathcal{J}} \alpha_j f_j$ with $\alpha_i \geq 0$ is a convex function;
- the pointwise supremum $f(x) := \sup_{j \in \mathcal{J}} f_j(x)$ is also a convex function.

The convexity of a function can also be defined in terms of its *epigraph*, namely the subset of $\mathbb{R}^d \times \mathbb{R}$ given by

$$\text{epi}(f) := \left\{(x,t) \in \mathbb{R}^d \times \mathbb{R} \mid f(x) \leq t\right\}, \tag{A.9}$$

as illustrated in Figure A.1. Many properties of convex functions can be stated in terms of properties of this epigraph. For instance, it is a straightforward exercise to show that the function f is convex if and only if its epigraph is a convex subset of $\mathbb{R}^d \times \mathbb{R}$. A convex function is *lower semi-continuous* if $\lim_{y \to x} f(y) \geq f(x)$ for all x in its domain. It can be shown that a convex function f is lower semi-continuous if and only if its epigraph is a closed set; in this case, f is said to be a *closed convex* function.

A.2.6   Conjugate Duality

Given a convex function $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ with non-empty domain, its conjugate dual is a new extended real-valued function
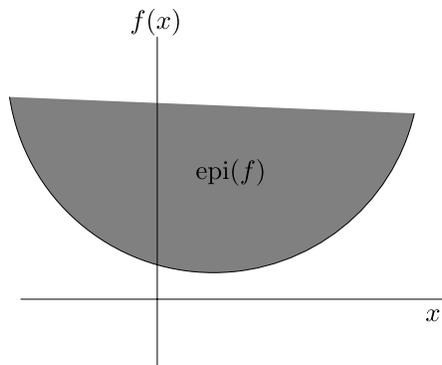
$f(x)$

epi(f)

x

Fig. A.1 The epigraph of a convex function is a subset of $\mathbb{R}^d \times \mathbb{R}$, given by $\text{epi}(f) = \{(x,t) \in \mathbb{R}^d \times \mathbb{R} \mid f(x) \le t\}$. For a convex function, this set is always convex.

$f^* : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$, defined as

$$
\begin{aligned}
f^*(y) &:= \sup_{x \in \mathbb{R}^d} \big\{ \langle y, x \rangle - f(x) \big\} \\
&= \sup_{x \in \text{dom}(f)} \big\{ \langle y, x \rangle - f(x) \big\},
\end{aligned}
\tag{A.10}
$$

where the second equality follows by definition of the domain of f. Note that f^* is always a convex function, since it is the pointwise supremum of a family of convex functions. Geometrically, this operation can be interpreted as computing the intercept of the supporting hyperplane to $\text{epi}(f)$ with normal vector $(y, -1) \in \mathbb{R}^d \times \mathbb{R}$, as illustrated in Figure A.2.

It is also possible to take the conjugate dual of this dual function thereby yielding a new function known as the biconjugate

$$
(f^*)^*(z) = \sup_{y \in \mathbb{R}^d} \big\{ \langle z, y \rangle - f^*(y) \big\}.
\tag{A.11}
$$

An important property of conjugacy is that whenever f is a closed convex function, then the biconjugate f^{**} is equal to the original function f.

For closed, convex, and differentiable functions f that satisfy additional technical conditions, the conjugate pair (f, f^*) induce a one-to-one correspondence between $\text{dom}(f)$ and $\text{dom}(f^*)$, based on the gradient mappings ∇f and ∇f^*. In particular, we say
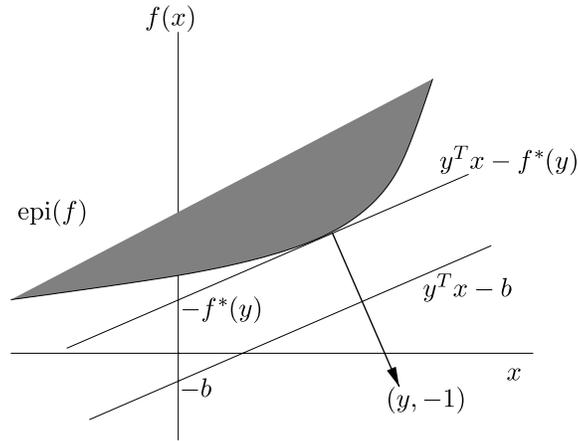
Fig. A.2 Interpretation of the conjugate dual function in terms of supporting hyperplanes to the epigraph. The negative value of the dual function $-f^*(y)$ corresponds to the intercept of the supporting hyperplane to epi(f) with normal vector $(y, -1) \in \mathbb{R}^d \times \mathbb{R}$.

that a convex function f is *essentially smooth* if it has a nonempty domain $C = \mathrm{dom}(f)$, f is differentiable throughout C°, and $\lim_{i \to \infty} \nabla f(x^i) = +\infty$ for any sequence $\{x^i\}$ contained in C, and converging to a boundary point of C. Note that a convex function with domain $\mathbb{R}^d$ is always essentially smooth, since its domain has no boundary. The property of essential smoothness is also referred to as *steepness* in some statistical treatments [43].

A function f is of *Legendre type* if it is strictly convex, differentiable on the interior C° of its domain, and essentially smooth. Letting $D = \mathrm{dom}(f^*)$, it can be shown that (f, C°) is a convex function of Legendre type if and only if (f^*, D°) is a convex function of Legendre type. In this case, the gradient mapping ∇f is one-to-one from C° onto the open convex set D°, continuous in both directions, and moreover $\nabla f^* = (\nabla f)^{-1}$. See Section 26 of Rockafellar [203] for further details on these types of Legendre correspondences between f and its dual f^*.

B

Proofs and Auxiliary Results: Exponential Families and Duality

---

In this section, we provide the proofs of Theorems 3.3 and 3.4 from Section 3, as well as some auxiliary results of independent interest.

B.1   Proof of Theorem 3.3

We prove the result first for a minimal representation, and then discuss its extension to the overcomplete case. Recall from Appendix A.2.3 that a convex subset of $\mathbb{R}^d$ is full-dimensional if its affine hull is equal to $\mathbb{R}^d$. We first note that $\mathcal{M}$ is a full-dimensional convex set if and only if the exponential family representation is minimal. Indeed, the representation is *not* minimal if and only if there exists some vector $a \in \mathbb{R}^d$ and constant $b \in \mathbb{R}$ such that $\langle a, \phi(x) \rangle = b$ holds ν-a.e. By definition of $\mathcal{M}$, this equality holds if and only if $\langle a, \mu \rangle = b$ for all $\mu \in \mathcal{M}$, which is equivalent to $\mathcal{M}$ not being full-dimensional.

Thus, if we restrict attention to minimal exponential families for the time-being, the set $\mathcal{M}$ is full-dimensional. Our proof makes use of the following properties of a full-dimensional convex set [see 112, 203]: (a) its interior $\mathcal{M}^\circ$ is nonempty, and the interior of the closure $[\overline{\mathcal{M}}]^\circ$ is equal to the interior $\mathcal{M}^\circ$; and (b) the interior $\mathcal{M}^\circ$ contains the zero

vector 0 if and only if for all nonzero $\gamma \in \mathbb{R}^d$, there exists some $\mu \in \mathcal{M}$ with $\langle \gamma, \mu \rangle > 0$.

We begin by establishing the inclusion $\nabla A(\Omega) \subseteq \mathcal{M}^\circ$. By shifting the potential ϕ by a constant vector if necessary, it suffices to consider the case $0 \in \nabla A(\Omega)$. Let $\theta^0 \in \Omega$ be the associated canonical parameter satisfying $\nabla A(\theta^0) = 0$. We prove that for all nonzero directions $\gamma \in \mathbb{R}^d$, there is some $\mu \in \mathcal{M}$ such that $\langle \gamma, \mu \rangle > 0$, which implies $0 \in \mathcal{M}^\circ$ by property (b). For any $\gamma \in \mathbb{R}^d$, the openness of Ω ensures the existence of some $\delta > 0$ such that $(\theta^0 + \delta\gamma) \in \Omega$. Using the strict convexity and differentiability of A on Ω and the fact that $\nabla A(\theta^0) = 0$ by assumption, we have

$$A(\theta^0 + \delta\gamma) > A(\theta^0) + \langle \nabla A(\theta^0), \delta\gamma \rangle = A(\theta^0).$$

Similarly, defining $\mu^\delta := \nabla A(\theta^0 + \delta\gamma)$, we can write

$$A(\theta^0) > A(\theta^0 + \delta\gamma) + \langle \mu^\delta, -\delta\gamma \rangle.$$

These two inequalities in conjunction imply that

$$\delta \langle \mu^\delta, \gamma \rangle > A(\theta^0 + \delta\gamma) - A(\theta^0) > 0.$$

Since $\mu^\delta \in \nabla A(\Omega) \subseteq \mathcal{M}$ and $\gamma \in \mathbb{R}^d$ was arbitrary, this establishes that $0 \in \mathcal{M}^\circ$.

Next we show that $\mathcal{M}^\circ \subseteq \nabla A(\Omega)$. As in the preceding argument, we may take $0 \in \mathcal{M}^\circ$ without loss of generality. Then, we must establish the existence of $\theta \in \Omega$ such that $\nabla A(\theta) = 0$. By convexity, it is equivalent to show that $\inf_{\theta \in \Omega} A(\theta)$ is attained. To establish the attainment of this infimum, we prove that A has no directions of recession, meaning that $\lim_{n \to +\infty} A(\theta^n) = +\infty$ for all sequences $\{\theta^n\}$ such that $\|\theta^n\| \to +\infty$.

For an arbitrary nonzero direction $\gamma \in \mathbb{R}^d$ and $\epsilon > 0$, consider the set $H_{\gamma,\epsilon} := \{x \in \mathcal{X}^m \mid \langle \gamma, \phi(x) \rangle \geq \epsilon\}$. Since $0 \in \mathcal{M}^\circ$, this set must have positive measure under ν for all sufficiently small $\epsilon > 0$. Otherwise, the inequality $\langle \gamma, \phi(x) \rangle \leq 0$ would hold ν-a.e., which implies that $\langle \gamma, \mu \rangle \leq 0$ for all $\mu \in \overline{\mathcal{M}}$. By the convexity of $\mathcal{M}$, this inequality would imply that $0 \notin [\overline{\mathcal{M}}]^\circ = \mathcal{M}^\circ$, which contradicts our starting assumption.

For an arbitrary $\theta^0 \in \Omega$, we now write

$$A(\theta^0 + t\gamma) \geq \log \int_{H_{\gamma,\epsilon}} \exp\left\{\langle \theta^0 + t\gamma, \phi(x) \rangle\right\} \nu(dx)$$

$$\geq t\epsilon + \underbrace{\log \int_{H_{\gamma,\epsilon}} \exp\left\{\langle \theta^0, \phi(x) \rangle\right\} \nu(dx)}_{C(\theta^0)} .$$

Note that we must have $C(\theta^0) > -\infty$, because

$$\exp\{\langle \theta^0, \phi(x) \rangle\} > 0 \quad \text{for all } x \in \mathcal{X}^m,$$

and $\nu(H_{\gamma,\epsilon}) > 0$. Hence, we conclude that $\lim_{t \to +\infty} A(\theta^0 + t\gamma) = +\infty$ for all directions $\gamma \in \mathbb{R}^d$, showing that A has no directions of recession.

Finally, we discuss the extension to the overcomplete case. For any overcomplete vector ϕ of sufficient statistics, let φ be a set of potential functions in an equivalent minimal representation. In particular, a collection φ can be specified by eliminating elements of ϕ until no affine dependencies remain. Let ∇A_φ and ∇A_ϕ be the respective mean parameter mappings associated with φ and ϕ, with the sets $\mathcal{M}_\varphi$ and $\mathcal{M}_\phi$ similarly defined. By the result just established, ∇A_φ is onto the interior of $\mathcal{M}_\varphi$. By construction of φ, each member in the relative interior of $\mathcal{M}_\phi$ is associated with a unique element in the interior of $\mathcal{M}_\varphi$. We conclude that the mean parameter mapping ∇A_ϕ is onto the relative interior of $\mathcal{M}_\phi$.

B.2 Proof of Theorem 3.4

We begin with proof of part (a), which we divide into three cases.

Case (i) $\mu \in \mathcal{M}^\circ$: In this case, Theorem 3.3 guarantees that the inverse image $(\nabla A)^{-1}(\mu)$ is nonempty. Any point in this inverse image attains the supremum in Equation (3.42). In a minimal representation, there is only one optimizing point, whereas there is an affine subset for an overcomplete representation. Nonetheless, for any $\theta(\mu) \in (\nabla A)^{-1}(\mu)$, the value of the optimum is given by

$$A^*(\mu) = \langle \theta(\mu), \mu \rangle - A(\theta(\mu)).$$

By definition (3.2) of the entropy, we have

$$-H(p_{\theta(\mu)}) = \mathbb{E}_{\theta(\mu)}[\langle \theta(\mu), \phi(X) \rangle - A(\theta(\mu))]$$
$$= \langle \theta(\mu), \mu \rangle - A(\theta(\mu)),$$

where the final equality uses linearity of expectation, and the fact that $\mathbb{E}_{\theta(\mu)}[\phi(X)] = \mu$.

Case (ii) $\mu \notin \overline{\mathcal{M}}$: Let $\operatorname{dom} A^* = \{\mu \in \mathbb{R}^d \mid A^*(\mu) < +\infty\}$ denote the effective domain of A^*. With this notation, we must prove that $\overline{\mathcal{M}} \supseteq \operatorname{dom} A^*$. In order to do so, we use Corollary 26.4.1 of Rockafellar [203] which asserts that if A is essentially smooth and lower semi-continuous, then we have the inclusions $[\operatorname{dom} A^*]^\circ \subseteq \nabla A(\Omega) \subseteq \operatorname{dom} A^*$, or equivalently

$$[\operatorname{dom} A^*]^\circ \subseteq \mathcal{M}^\circ \subseteq \operatorname{dom} A^*,$$

since $\nabla A(\Omega) = \mathcal{M}^\circ$ from Theorem 3.3. Since both $\mathcal{M}$ and $\operatorname{dom} A^*$ are convex sets, taking closures in these inclusions yields that $\overline{\operatorname{dom} A^*} = \overline{[\mathcal{M}^\circ]} = \overline{\mathcal{M}}$, where the second equality follows by the convexity of $\mathcal{M}$. Therefore, by definition of the effective domain, $A^*(\mu) = +\infty$ for any $\mu \notin \overline{\mathcal{M}}$.

It remains to verify that A is both lower semi-continuous, and essentially smooth (see Appendix A.2.5 for the definition). Recall from Proposition 3.1 that A is differentiable; hence, it is lower semi-continuous on its domain. To establish that A is essentially smooth, let θ^b be a boundary point, and let $\theta^0 \in \Omega$ be arbitrary. Since the set Ω is convex and open, the line $\theta^t := t\theta^b + (1-t)\theta^0$ is contained in Ω for all $t \in [0,1)$ (see Theorem 6.1 of Rockafellar [203]). Using the differentiability of A on Ω and its convexity (see Proposition 3.1(b)), for any $t < 1$, we can write

$$A(\theta^0) \geq A(\theta^t) + \langle \nabla A(\theta^t), \theta^0 - \theta^t \rangle.$$

Re-arranging and applying the Cauchy–Schwartz inequality to the inner product term yields that

$$A(\theta^t) - A(\theta^0) \leq \|\theta^t - \theta^0\| \, \|\nabla A(\theta^t)\|.$$

Now as $t \to 1^-$, the left-hand side tends to infinity by the lower semi-continuity of A, and the regularity of the exponential family.

Consequently, the right-hand side must also tend to infinity; since $\|\theta^t - \theta^0\|$ is bounded, we conclude that $\|\nabla A(\theta^t)\| \to +\infty$, which shows that A is essentially smooth.

Case (iii) $\mu \in \overline{\mathcal{M}} \backslash \mathcal{M}^\circ$: Since A^* is defined as a conjugate function, it is lower semi-continuous. Therefore, the value of $A^*(\mu)$ for any boundary point $\mu \in \overline{\mathcal{M}} \backslash \mathcal{M}^\circ$ is determined by the limit over a sequence approaching μ from inside $\mathcal{M}^\circ$, as claimed.

We now turn to the proof of part (b). From Proposition 3.1, A is convex and lower semi-continuous, which ensures that $(A^*)^* = A$, and part (a) shows that $\overline{\mathrm{dom}\, A^*} = \overline{\mathcal{M}}$. Consequently, we can write

$$
\begin{aligned}
A(\theta) &= \sup_{\mu \in \overline{\mathcal{M}}} \big\{ \langle \theta, \mu \rangle - A^*(\mu) \big\} \\
&= \sup_{\mu \in \mathcal{M}} \big\{ \langle \theta, \mu \rangle - A^*(\mu) \big\},
\end{aligned}
\tag{B.1}
$$

since it is inconsequential whether we take the supremum over $\mathcal{M}$ or its closure $\overline{\mathcal{M}}$.

Last, we prove part (c) of the theorem. For a minimal representation, Proposition 3.2 and Theorem 3.3 guarantee that the gradient mapping ∇A is a bijection between Θ and $\mathcal{M}^\circ$. On this basis, it follows that the gradient mapping ∇A^* also exists and is bijective [203], whence the supremum (B.1) is attained at a unique point whenever $\theta \in \Omega$. The analogous statement for an overcomplete representation can be proved via reduction to a minimal representation.

B.3   General Properties of $\mathcal{M}$ and A^*

In this section, we state and prove some auxiliary results of independent interest on general properties of the dual function A^*, and the associated set $\mathcal{M}$ of valid mean parameters.

B.3.1   Properties of $\mathcal{M}$

From its definition, it is clear that $\mathcal{M}$ is always a convex set. Other more specific properties of $\mathcal{M}$ turn out to be determined by the properties of the exponential family. Recall from Appendix A.2.3 that a convex set

$\mathcal{M} \subseteq \mathbb{R}^d$ is full-dimensional if its affine hull is equal to $\mathbb{R}^d$. With this notion, we have the following:

Proposition B.1   The set $\mathcal{M}$ has the following properties:

(a) $\mathcal{M}$ is full-dimensional if and only if the exponential family is minimal.

(b) $\mathcal{M}$ is bounded if and only if $\Theta = \mathbb{R}^d$ and A is globally Lipschitz on $\mathbb{R}^d$.

Proof. (a) The representation is *not* minimal if and only if there exists some vector $a \in \mathbb{R}^d$ and constant $b \in \mathbb{R}$ such that $\langle a, \phi(x) \rangle = b$ holds ν-a.e. By definition of $\mathcal{M}$, this equality holds if and only if $\langle a, \mu \rangle = b$ for all $\mu \in \mathcal{M}$, which is equivalent to $\mathcal{M}$ not being full-dimensional.

(b) The recession function A_∞ is the support function $\sup_{\mu \in \mathcal{M}} \langle \mu, \theta \rangle$. Therefore, the set $\mathcal{M}$ is bounded if and only if $A_\infty(\theta)$ is finite for all $\theta \in \mathbb{R}^d$. The recession function A_∞ is finite-valued if and only if A is Lipschitz and hence finite on all of $\mathbb{R}^d$ (see Proposition 3.2.7 in Hiriart-Urruty and Lemaréchal [112]).   □

The necessity of the condition $\Omega = \mathbb{R}^d$ for $\mathcal{M}$ to be bounded is clear from the boundary behavior of ∇A given in Proposition 3.1. However, the additional global Lipschitz condition is also necessary, as demonstrated by the Poisson family (see Table 3.2). In this case, we have $\Theta = \mathbb{R}$ yet the set of mean parameters $\mathcal{M} = (0, +\infty)$ is unbounded. This unboundedness occurs because the function $A(\theta) = \exp(\theta)$, while finite on $\mathbb{R}$, is not globally Lipschitz.

B.3.2   Properties of A^*

Despite the fact that A^* is not given in closed form, a number of properties can be inferred from its variational definition (3.42). For instance, an immediate consequence is that A^* is always convex. More specific properties of A^* depend on the nature of the exponential family. Recall from Appendix A.2.6 the definition of an essentially smooth convex function.

Proposition B.2 The dual function A^* is always convex and lower semi-continuous. Moreover, in a minimal and regular exponential family:

(a) A^* is differentiable on $\mathcal{M}^\circ$, and $\nabla A^*(\mu) = (\nabla A)^{-1}(\mu)$.
(b) A^* is strictly convex and essentially smooth.

Proof. The convexity and lower semi-continuity follow because A^* is the supremum of collection of functions linear in μ. Since both A and A^* are lower semi-continuous, the dual A^* has these properties if and only if A is strictly convex and essentially smooth (see our discussion of Legendre duality in Appendix A.2.6, and Theorem 26.3 in Rockafellar [203]). For a minimal representation, A is strictly convex by Proposition 3.1, and from the proof of Theorem 3.4(a), it is also essentially smooth, so that the stated result follows. $\qquad\square$

This result is analogous to Proposition 3.1, in that the conditions stated ensure the essential smoothness of the dual function A^* in a minimal representation. The boundary behavior of ∇A^* can be verified explicitly for the examples shown in Table 3.2, for which we have closed form expressions for A^*. For instance, in the Bernoulli case, we have $\mathcal{M} = [0,1]$ and $|\nabla A^*(\mu)| = |\log[(1-\mu)/\mu]|$, which tends to infinity as $\mu \to 0^+$ or $\mu \to 1^-$. Similarly, in the Poisson case, we have $\mathcal{M} = (0, +\infty)$ and $|\nabla A^*(\mu)| = |\log\mu|$, which tends to infinity as μ tends to the boundary point 0.

B.4 Proof of Theorem 4.2(b)

For any MRF defined by a tree T, both components of the Bethe variational problem (4.16) are exact:

(a) by Proposition 4.1, the set $\mathbb{L}(T)$ is equivalent to the marginal polytope $\mathbb{M}(T)$, and
(b) the Bethe entropy H_{Bethe} is equivalent to the negative dual function $-A^*$.

Consequently, for the tree-structured problem, the Bethe variational principle is equivalent to a special case of the conjugate duality between A and A^* from Theorem 3.4. Therefore, the value of the optimized Bethe problem is equal to the cumulant function value $A(\theta)$, as claimed. Finally, Theorem 3.4(c) implies that the optimum τ^* corresponds to the exact marginal distributions of the tree-structured MRF. Strict convexity implies that the solution is unique.

B.5   Proof of Theorem 8.1

We begin by proving assertion (8.4a). Let $\mathcal{P}$ be the space of all densities p, taken with respect to the base measure ν associated with the given exponential family. On the one hand, for any $p \in \mathcal{P}$, we have $\int \langle \theta, \phi(x) \rangle p(x) \nu(dx) \leq \max_{x \in \mathcal{X}^m} \langle \theta, \phi(x) \rangle$, whence

$$\sup_{p \in \mathcal{P}} \int \langle \theta, \phi(x) \rangle p(x) \nu(dx) \leq \max_{x \in \mathcal{X}^m} \langle \theta, \phi(x) \rangle. \tag{B.2}$$

Since the support of ν is $\mathcal{X}^m$, equality is achieved in the inequality (B.2) by taking a sequence of densities $\{p^n\}$ converging to a delta function $\delta_{x^*}(x)$, where $x^* \in \arg\max_x \langle \theta, \phi(x) \rangle$. Finally, by linearity of expectation and the definition of $\mathcal{M}$, we have $\sup_{p \in \mathcal{P}} \int \langle \theta, \phi(x) \rangle p(x) \nu(dx) = \sup_{\mu \in \mathcal{M}} \langle \theta, \mu \rangle$, which establishes the claim (8.4a).

  We now turn to the claim (8.4b). By Proposition 3.1, the function A is lower semi-continuous. Therefore, for all $\theta \in \Omega$, the quantity $\lim_{\beta \to +\infty} A(\beta\theta)/\beta$ is equivalent to the recession function of A, which we denote by A_∞ (see Corollary 8.5.2 of Rockafellar [203]). Hence, it suffices to establish that $A_\infty(\theta)$ is equal to $\sup_{\mu \in \mathcal{M}} \langle \theta, \mu \rangle$. Using the lower semi-continuity of A and Theorem 13.3 of Rockafellar [203], the recession function of A corresponds to the support function of the effective domain of its dual. By Theorem 3.4, we have $\overline{\mathrm{dom}\,A^*} = \overline{\mathcal{M}}$, whence $A_\infty(\theta) = \sup_{\mu \in \bar{\mathcal{M}}} \langle \theta, \mu \rangle$. Finally, the supremum is not affected by taking the closure.

C

Variational Principles for Multivariate Gaussians

In this appendix, we describe how the general variational representation (3.45) specializes to the multivariate Gaussian case. For a Gaussian Markov random field, as described in Example 3.7, computing the mapping $(\theta, \Theta) \mapsto (\mu, \Sigma)$ amounts to computing the mean vector $\mu \in \mathbb{R}^d$, and the covariance matrix $\Sigma - \mu\mu^T$. These computations are typically carried out through the so-called normal equations [126]. In this section, we derive such normal equations for Gaussian inference from the principle (3.45).

C.1 Gaussian with Known Covariance

We first consider the case of $X = (X_1, \ldots, X_m)$ with unknown mean $\mu \in \mathbb{R}^m$, but *known covariance* matrix Λ, which we assume to be strictly positive definite (hence invertible). Under this assumption, we may alternatively use the precision matrix P, corresponding to the inverse covariance Λ^{-1}. This collection of models can be written as an m-dimensional exponential family with respect to the base measure

$$\nu(dx) = \frac{\exp\left(-\frac{1}{2}x^T P x\right)}{\sqrt{(2\pi)^m |P^{-1}|}} dx,$$

with densities of the form $p_\theta(x) = \exp\left\{\sum_{i=1}^m \theta_i x_i - A(\theta)\right\}$. With this notation, the well-known *normal equations* for computing the Gaussian mean vector $\mu = \mathbb{E}[X]$ are given by $\mu = P^{-1}\theta$.

Let us now see how these normal equations are a consequence of the variational principle (3.45). We first compute the cumulant function for the specified exponential family as follows:

$$
\begin{aligned}
A(\theta) &= \log \int_{\mathbb{R}^m} \exp\left\{\sum_{i=1}^m \theta_i x_i\right\} \; \frac{\exp\left(-\frac{1}{2}x^T P x\right)}{\sqrt{(2\pi)^m |P^{-1}|}} dx \\
&= \frac{1}{2}\theta^T P^{-1}\theta \\
&= \frac{1}{2}\theta^T \Lambda\theta,
\end{aligned}
$$

where have completed the square in order to compute the integral.[1] Consequently, we have $A(\theta) < +\infty$ for all $\theta \in \mathbb{R}^m$, meaning that $\Omega = \mathbb{R}^m$. Moreover, the mean parameter $\mu = \mathbb{E}[X]$ takes values in all of $\mathbb{R}^m$, so that $\mathcal{M} = \mathbb{R}^m$. Next, a straightforward calculation yields that the conjugate dual of A is given by $A^*(\mu) = \frac{1}{2}\mu^T P\mu$. Consequently, the variational principle (3.45) takes the form:

$$
A(\theta) = \sup_{\mu \in \mathbb{R}^m} \left\{\langle \theta, \mu \rangle - \frac{1}{2}\mu^T P\mu\right\},
$$

which is an unconstrained quadratic program. Taking derivatives to find the optimum yields that $\mu(\theta) = P^{-1}\theta$.

Thus, we have shown that the normal equations for Gaussian inference are a consequence of the variational principle (3.45) specialized to this particular exponential family. Moreover, by the Hammersley–Clifford theorem [25, 102, 106], the precision matrix P has the same sparsity pattern as the graph adjacency matrix (i.e., for all $i \neq j$, $P_{ij} \neq 0$ implies that $(i,j) \in E$). Consequently, the matrix-inverse-vector problem $P^{-1}\theta$ can often be solved very quickly by specialized methods that exploit the graph structure of P (see, for instance, Golub and van Loan [99]). Perhaps the best-known example is the case of a tri-diagonal matrix P, in which case the original Gaussian vector X is Markov with respect to a chain-structured graph, with edges $(i, i + 1)$

---

[1] Note that since $[\nabla^2 A(\theta)]_{ij} = \mathrm{cov}(X_i, X_j)$, this calculation is consistent with Proposition 3.1 (see Equation (3.41b)).

for $i = 1, \ldots, m - 1$. More generally, the Kalman filter on trees [262] can be viewed as a fast algorithm for solving the matrix-inverse-vector system of normal equations. For graphs without cycles, there are also local iterative methods for solving the normal equations. In particular, as we discuss in Example 5.3 in Section 5, Gaussian mean field theory leads to the Gauss–Jacobi updates for solving the normal equations.

C.2   Gaussian with Unknown Covariance

We now turn to the more general case of a Gaussian with unknown covariance (see Example 3.3). Its mean parameterization is specified by the vector $\mu = \mathbb{E}[X] \in \mathbb{R}^m$ and the second-order moment matrix $\Sigma = \mathbb{E}[XX^T] \in \mathbb{R}^{m \times m}$. The set $\overline{\mathcal{M}}$ of valid mean parameters is completely characterized by the positive semidefiniteness (PSD) constraint $\Sigma - \mu\mu^T \succeq 0$ (see Example 3.7). Turning to the form of the dual function, given a set of valid mean parameters $(\mu, \Sigma) \in \mathcal{M}^\circ$, we can associate them with a Gaussian random vector X with mean $\mu \in \mathbb{R}^m$ and covariance matrix $\Sigma - \mu\mu^T \succ 0$. The entropy of such a multivariate Gaussian takes the form [57]:

$$h(X) = -A^*(\mu, \Sigma) = \frac{1}{2} \log \det \left[\Sigma - \mu\mu^T \right] + \frac{m}{2} \log 2\pi e.$$

By the Schur complement formula [114], we may rewrite the negative dual function as

$$-A^*(\mu, \Sigma) = \frac{1}{2} \log \det \begin{bmatrix} 1 & \mu^T \\ \mu & \Sigma \end{bmatrix} + \frac{m}{2} \log 2\pi e. \qquad \text{(C.1)}$$

Since the log-determinant function is strictly concave on the cone of positive semidefinite matrices [39], this representation demonstrates the convexity of A^* in an explicit way.

These two ingredients allow us to write the variational principle (3.45) specialized to the multivariate Gaussian with unknown covariance as follows: the value of cumulant function $A(\theta, \Theta)$ is equal to

$$\sup_{(\mu, \Sigma), \Sigma - \mu\mu^T \succ 0} \left\{ \langle \theta, \mu \rangle + \frac{1}{2} \langle\!\langle \Theta, \Sigma \rangle\!\rangle + \frac{1}{2} \log \det \begin{bmatrix} 1 & \mu^T \\ \mu & \Sigma \end{bmatrix} + \frac{1}{2} \log 2\pi e \right\}.$$
$$\text{(C.2)}$$

This problem is an instance of a well-known class of concave maximization problems, known (for obvious reasons) as a log-determinant problem [235]. In general, log-determinant problems can be solved efficiently using standard methods for convex programming, such as interior point methods. However, this specific log-determinant problem (C.2) actually has a closed-form solution: in particular, it can be verified by a Lagrangian reformulation that the optimal pair (μ^*, Σ^*) is specified by the relations

$$\Sigma^* - \mu^*(\mu^*)^T = -[\Theta]^{-1}, \quad \text{and} \quad \mu^* = -[\Theta]^{-1}\,\theta. \qquad \text{(C.3)}$$

Note that these relations, which emerge as the optimality conditions for the log-determinant problem (C.2), are actually familiar statements. Recall from our exponential representation of the Gaussian density (3.12) that $-\Theta$ is the precision matrix. Consequently, the first equality in Equation (C.3) confirms that the covariance matrix is the inverse of the precision matrix, whereas the second equality corresponds to the normal equations for the mean μ of a Gaussian. Thus, as a special case of the general variational principle (3.45), we have re-derived the familiar equations for Gaussian inference.

C.3   Gaussian Mode Computation

In Section C.2, we showed that for a multivariate Gaussian with unknown covariance, the general variational principle (3.45) corresponds to a log-determinant optimization problem. In this section, we show that the mode-finding problem for a Gaussian is a semidefinite program (SDP), another well-known class of convex optimization problems [234, 235]. Consistent with our development in Section 8.1, this SDP arises as the "zero-temperature limit" of the log-determinant problem. Recall that for the multivariate Gaussian, the mean parameters are the mean vector $\mu = \mathbb{E}[X]$ and second-moment matrix $\Sigma = \mathbb{E}[XX^T]$, and the set $\overline{\mathcal{M}}$ of valid mean parameters is characterized by the semidefinite constraint $\Sigma - \mu\mu^T \succeq 0$, or equivalently (via the Schur complement formula [114]) by a linear matrix inequality

$$\Sigma - \mu\mu^T \succeq 0 \iff \begin{bmatrix} 1 & \mu^T \\ \mu & \Sigma \end{bmatrix} \succeq 0. \qquad \text{(C.4)}$$

Consequently, by applying Theorem 8.1, we conclude that

$$\max_{x \in \mathcal{X}} \left\{ \langle \theta, x \rangle + \frac{1}{2} \langle\!\langle \Theta,\, xx^T \rangle\!\rangle \right\} = \max_{\Sigma - \mu\mu^T \succeq 0} \left\{ \langle \theta, \mu \rangle + \frac{1}{2} \langle\!\langle \Theta,\, \Sigma \rangle\!\rangle \right\} \quad \text{(C.5)}$$

The problem on the left-hand side is simply a quadratic program, with optimal solution corresponding to the mode $x^* = -[\Theta]^{-1}\theta$ of the Gaussian density. The problem on the right-hand side is a semidefinite program [234], as it involves an objective function that is linear in the matrix-vector pair (μ, Σ), with a constraint defined by the linear matrix inequality (C.4).

In order to demonstrate how the optimum of this semidefinite program (SDP) recovers the Gaussian mode x^*, we begin with the fact [114] that a matrix B is positive semidefinite if and only if $\langle\!\langle B,\, C \rangle\!\rangle := \text{trace}(BC) \geq 0$ for all other positive semidefinite matrices C. (This property corresponds to the self-duality of the cone $\mathcal{S}_+^m$.) Applying this fact to the matrices $B = \Sigma - \mu\mu^T \succeq 0$ and $C = -\Theta \succ 0$ yields the inequality $\langle\!\langle \Theta,\, \Sigma \rangle\!\rangle \leq \langle\!\langle \Theta,\, \mu\mu^T \rangle\!\rangle$, which in turn implies that

$$\langle \theta, \mu \rangle + \frac{1}{2} \langle\!\langle \Theta,\, \Sigma \rangle\!\rangle \leq \langle \theta, \mu \rangle + \frac{1}{2} \langle\!\langle \Theta,\, \mu\mu^T \rangle\!\rangle. \qquad \text{(C.6)}$$

Observe that the right-hand side is a quadratic program in the vector $\mu \in \mathbb{R}^m$, with its maximum attained at $\mu^* = -[\Theta]^{-1}\theta$. Consequently, if we maximize the left-hand side over the set $\Sigma - \mu\mu^T \succeq 0$, the bound will be achieved at μ^* and $\Sigma^* = \mu^*(\mu^*)^T$. The interpretation is the optimal solution corresponds to a degenerate Gaussian, specified by mean parameters (μ^*, Σ^*), with a zero covariance matrix, so that all mass concentrates on its mean $\mu^* = x^*$.

In practice, it is not necessary to solve the semidefinite program (C.5) in order to compute the mode of a Gaussian problem. For multivariate Gaussians, the mode (or mean) can be computed by more direct methods, including Kalman filtering [126], corresponding to the sum-product algorithm on trees, by numerical methods such as conjugate gradient [67], by the max-product/min-sum algorithm (see Section 8.3), both of which solve the quadratic program in an iterative manner, or by iterative algorithms based on tractable subgraphs [161, 223]. However, the SDP-based formulation (C.6) provides valuable perspective on the use of semidefinite relaxations for integer programming problems, as discussed in Section 9.

D

Clustering and Augmented Hypergraphs

In this appendix, we elaborate on various techniques use to transform hypergraphs so as to apply the Kikuchi approximation and related cluster variational methods. The most natural strategy is to construct variational relaxations directly using the original hypergraph. The Bethe approximation is of this form, corresponding to the case when the original structure is an ordinary graph. Alternatively, it can be beneficial to develop approximations based on an *augmented hypergraph* $G = (V, E)$ built on top of the original hypergraph. A natural way to define new hypergraphs is by defining new hyperedges over clusters of the original vertices; a variety of such clustering methods have been discussed in the literature [e.g., 167, 188, 268, 269].

D.1 Covering Augmented Hypergraphs

Let $G' = (V, E')$ denote the original hypergraph, on which the undirected graphical model or Markov random field is defined. We consider an augmented hypergraph $G = (V, E)$, defined over the same set of vertices but with a hyperedge set E that includes all the original hyperedges E'. We say that the original hypergraph G' is *covered* by the augmented hypergraph G. A desirable feature of this requirement is

that any Markov random field (MRF) defined by G' can also be viewed as an MRF on a covering hypergraph G, simply by setting $\theta_h = 0$ for all $h \in E \backslash E'$.

Example D.1 (Covering Hypergraph). To illustrate, suppose that the original hypergraph G' is simply an ordinary graph — namely, the 3×3 grid shown in Figure D.1(a). As illustrated in panel (b), we cluster the nodes into groups of four, which is known as Kikuchi 4-plaque clustering in statistical physics [268, 269]. We then form the augmented hypergraph G shown in panel (c), with hyperedge set $E := V \cup E' \cup \big\{\{1,2,4,5\},\{2,3,5,6\},\{4,5,7,8\},\{5,6,8,9\}\big\}$. The darkness of the boxes in this diagram reflects the depth of the hyperedges in the poset diagram. This hypergraph covers the original (hyper)graph, since it includes as hyperedges all edges and vertices of the original 3×3 grid.

As emphasized by Yedidia et al. [269], it turns out to be important to ensure that every hyperedge (including vertices) in the original hypergraph G' is counted exactly once in the augmented hypergraph G. More specifically, for a given hyperedge $h \in E$, consider the set $\mathcal{A}^+(h) := \{f \in E \mid f \supseteq h\}$ of hyperedges in E that contain h. Let us recall the definition (4.46) of the overcounting numbers associated with the hypergraph G. In particular, these overcounting numbers are defined in terms of the Möbius function associated with G, viewed as a poset, in the following way:

$$c(f) := \sum_{e \in \mathcal{A}^+(f)} \omega(f,e), \tag{D.1}$$

The *single counting criterion* requires that for all vertices or hyperedges of the original graph — that is, for all $h' \in E' \cup V$— there holds

$$\sum_{f \in \mathcal{A}^+(h')} c(f) = 1. \tag{D.2}$$

In words, the sum of the overcounting numbers over all hyperedges f that contain h' must be equal to one.
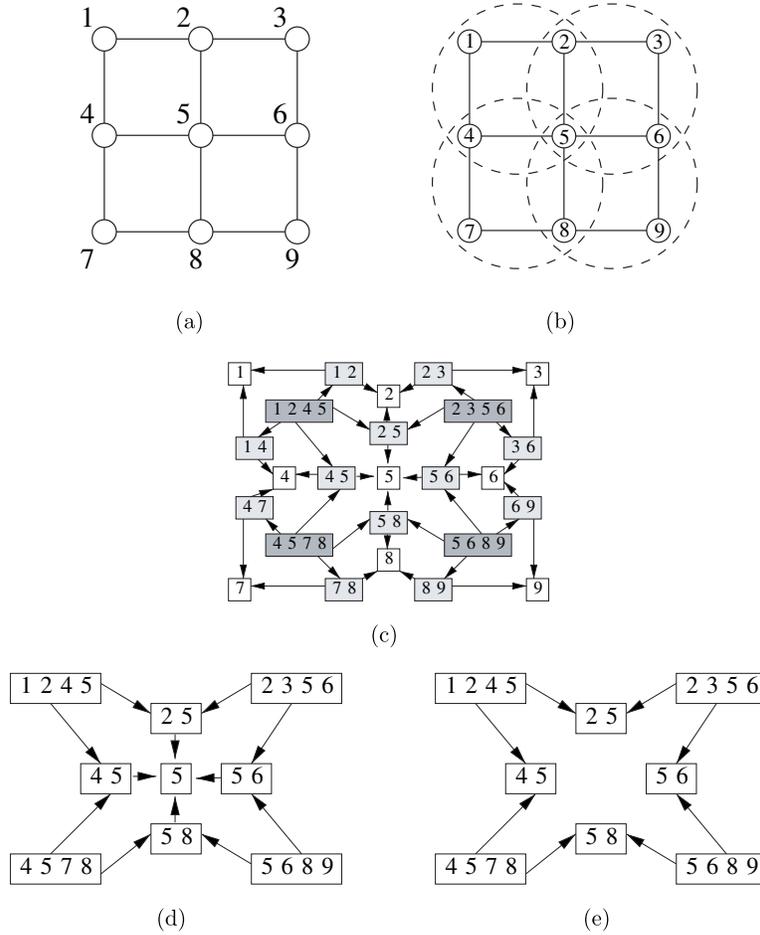
Fig. D.1 Constructing new hypergraphs via clustering and the single counting criterion.
(a) Original (hyper)graph G' is a 3×3 grid. Its hyperedge set E' consists of the union of
the vertex set with the (ordinary) edge set. (b) Nodes are clustered into groups of four.
(c) A covering hypergraph G formed by adjoining these 4-clusters to the original hyperedge
set E'. Darkness of the boxes indicates depth of the hyperedges in the poset representation.
(d) An augmented hypergraph constructed by the Kikuchi method. (e) A third augmented
hypergraph that fails the single counting criterion for (5).

Example D.2 (Single Counting). To illustrate the single counting
criterion, we consider two additional hypergraphs that can be con-
structed from the 3×3 grid of Figure D.1(a). The vertex set and edge
set of the grid form the original hyperedge set E'. The hypergraph

in panel (d) is constructed by the Kikuchi method described by Yedidia et al. [268, 269]. In this construction, we include the four clusters, all of their pairwise intersections, and all the intersections of intersections (only $\{5\}$ in this case). The hypergraph in panel (e) includes only the hyperedges of size four and two; that is, it omits the hyperedge $\{5\}$.

Let us focus first on the hypergraph (e), and understand why it violates the single counting criterion for hyperedge $\{5\}$. Viewed as a poset, all of the maximal hyperedges (of size four) in this hypergraph have a counting number of $c(h) = \omega(h, h) = 1$. Any hyperedge f of size two has two parents, each with an overcounting number of 1, so that $c(f) = 1 - (1 + 1) = -1$. The hyperedge $\{5\}$ is a member of the original hyperedge set E' (of the 3×3 grid), but not of the augmented hypergraph. It is included in all the hyperedges, so that $\mathcal{A}^+(\{5\}) = E$ and $\sum_{h \in \mathcal{A}^+(\{5\})} c(h) = 0$. Thus, the single criterion condition fails to hold for hypergraph (e). In contrast, it can be verified that for the hypergraphs in panels (c) and (d), the single counting condition holds for all hyperedges $h' \in E'$.

There is another interesting fact about hypergraphs (c) and (d). If we eliminate from hypergraph (c) all hyperedges that have zero overcounting numbers, the result is hypergraph (d). To understand this reduction, consider for instance the hyperedge $\{1, 4\}$ which appears in (c) but not in (d). Since it has only one parent that is a maximal hyperedge, we have $c(\{1, 4\}) = 0$. In a similar fashion, we see that $c(\{1, 2\}) = 0$. These two equalities together imply that $c(\{1\}) = 0$, so that we can eliminate hyperedges $\{1, 2\}$, $\{1, 4\}$, and $\{1\}$ from hypergraph (c). By applying a similar argument to the remaining hyperedges, we can fully reduce hypergraph (c) to hypergraph (d).

It turns out that if the augmented hypergraph G covers the original hypergraph G', then the single counting criterion is always satisfied. Implicit in this definition of covering is that the hyperedge set E' of the original hypergraph includes the vertex set, so that Equation (D.2) should hold for the vertices. The proof is quite straightforward.

Lemma D.1 (Single Counting). For any $h \in E$, the associated over-counting numbers satisfy the identity $\sum_{e \in \mathcal{A}^+(h)} c(e) = 1$, which can be written equivalently as $c(h) = 1 - \sum_{e \in \mathcal{A}(h)} c(e)$.

Proof. From the definition of $c(h)$, we have the identity:

$$\sum_{h \in \mathcal{A}^+(g)} c(h) = \sum_{h \in \mathcal{A}^+(g)} \sum_{f \in \mathcal{A}^+(h)} \omega(h, f). \tag{D.3}$$

Considering the double sum on the right-hand side, we see that for a fixed $d \in \mathcal{A}^+(g)$, there is a term $\omega(d, e)$ for each e such that $g \subseteq e \subseteq d$. Using this observation, we can write

$$\sum_{h \in \mathcal{A}^+(g)} \sum_{f \in \mathcal{A}^+(h)} \omega(h, f) = \sum_{d \in \mathcal{A}^+(g)} \sum_{\{e \mid g \subseteq e \subseteq d\}} \omega(e, d)$$

$$\overset{(a)}{=} \sum_{d \in \mathcal{A}^+(g)} \delta(d, g)$$

$$\overset{(b)}{=} 1.$$

Here equality (a) follows from the definition of the Möbius function (see Appendix E.1), and $\delta(d, g)$ is the Kronecker delta function, from which equality (b) follows. □

Thus, the construction that we have described, in which the hyper-edges (including all vertices) of the original hypergraph G' are covered by G and the partial ordering is set inclusion, ensures that the single counting criterion is always satisfied. We emphasize that there is a broad range of other possible constructions [e.g., 268, 269, 188, 167].

D.2   Specification of Compatibility Functions

Our final task is to specify how to assign the compatibility functions associated with the original hypergraph $G' = (V, E')$ with the hyperedges of the augmented hypergraph $G = (V, E)$. It is convenient to use the notation $\psi'_g(x_g) := \exp\{\theta_g(x_g)\}$ for the compatibility functions of the original hypergraph, corresponding to terms in the product (4.48). We can extend this definition to all hyperedges in E by

setting $\psi'_h(x_h) \equiv 1$ for any hyperedge $h \in E \backslash E'$. For each hyperedge $h \in E$, we then define a new compatibility function ψ_h as follows:

$$\psi_h(x_h) := \psi'_h(x_h) \prod_{g \in \mathcal{S}(h)} \psi'_g(x_g), \tag{D.4}$$

where $\mathcal{S}(h) := \{g \in E' \backslash E \mid g \subset h\}$ is the set of hyperedges in $E' \backslash E$ that are subsets of h. To illustrate this definition, consider the Kikuchi construction of Figure D.1(d), which is an augmented hypergraph for the 3×3 grid in Figure D.1(a). For the hyperedge (25), we have $\mathcal{S}(25) = \{(2)\}$, so that $\psi_{25} = \psi'_{25}\psi'_2$. On the other hand, for the hyperedge (1245), we have $\psi'_{1245} \equiv 1$ (since (1245) appears in E but not in E'), and $\mathcal{S}(1245) = \{(1),(12),(14)\}$. Accordingly, Equation (D.4) yields $\psi_{1245} = \psi'_1 \psi'_{12} \psi'_{14}$. More generally, using the definition (D.4), it is straightforward to verify that the equivalence

$$\prod_{h \in E} \psi_h(x_h) = \prod_{g \in E'} \psi'_g(x_g)$$

holds, so that we have preserved the structure of the original MRF.

E

Miscellaneous Results

This appendix collects together various miscellaneous results on graphical models and posets.

E.1 Möbius Inversion

This appendix provides brief overview of the Möbius function associated with a partially ordered set. We refer the reader to Chapter 3 of Stanley [221] for a thorough treatment. A *partially ordered set* or *poset* $\mathcal{P}$ consists of a set along with a binary relation on elements of the set, which we denote by $\subseteq$, satisfying reflexivity ($g \subseteq g$ for all $g \in \mathcal{P}$), anti-symmetry ($g \subseteq h$ and $h \subseteq g$ implies $g = h$), and transitivity ($f \subseteq g$ and $g \subseteq h$ implies $f \subseteq g$). Although posets can be defined more generally, for our purposes it suffices to restrict attention to finite posets.

The *zeta function* of a poset is a mapping $\zeta : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ defined by

$$\zeta(g, h) = \begin{cases} 1 & \text{if } g \subseteq h, \\ 0 & \text{otherwise.} \end{cases} \tag{E.1}$$

The Möbius function $\omega : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ arises as the multiplicative inverse of this zeta function. It can be defined in a recursive fashion, by first

specifying $\omega(g,g) = 1$ for all $g \in \mathcal{P}$, and $\omega(g,h) = 0$ for all $h \subsetneq g$. Once $\omega(g,f)$ has been defined for all f such that $g \subseteq f \subset h$, we then define:

$$\omega(g,h) = -\sum_{\{f \mid g \subseteq f \subset h\}} \omega(g,f). \tag{E.2}$$

With this definition, it can be seen that ω and ζ are multiplicative inverses, in the sense that

$$\sum_{f \in \mathcal{P}} \omega(g,f)\zeta(f,h) = \sum_{\{f \mid g \subseteq f \subseteq h\}} \omega(g,f) = \delta(g,h), \tag{E.3}$$

where $\delta(g,h)$ is the Kronecker delta.

An especially simple example of a poset is the set $\mathcal{P}$ of all subsets of $\{1,\ldots,m\}$, using set inclusion as the partial order. In this case, a straightforward calculation shows that the Möbius function is given by $\omega(g,h) = (-1)^{|h \backslash g|} \mathbb{I}[g \subseteq h]$. The inverse formula (E.3) then corresponds to the assertion

$$\sum_{\{f \mid g \subseteq f \subseteq h\}} (-1)^{|f \backslash g|} = \delta(g,h). \tag{E.4}$$

We now state the Möbius inversion formula for functions defined on a poset:

Lemma E.1 (Möbius Inversion Formula). Given functions real-valued functions Υ and Ω defined on a poset $\mathcal{P}$, then

$$\Omega(h) = \sum_{g \subseteq h} \Upsilon(g) \qquad \text{for all } h \in \mathcal{P} \tag{E.5}$$

if and only if

$$\Upsilon(h) = \sum_{g \subseteq h} \Omega(g)\omega(g,h) \qquad \text{for all } h \in \mathcal{P}. \tag{E.6}$$

E.2   Auxiliary Result for Proposition 9.3

In this appendix, we prove that the matrix $B(\alpha,\beta) = (-1)^{|\beta \backslash \alpha|} \mathbb{I}[\alpha \subseteq \beta]$ diagonalizes the matrix $N_m[\tau]$, for any vector $\tau \in \mathbb{R}^{2^m}$. For any pair

α, β, we calculate

$$\sum_{\gamma,\delta} B(\alpha,\gamma)\big(N_m[\tau]\big)_{\gamma,\delta} B(\delta,\beta) = \sum_{\{\gamma \mid \alpha \subseteq \gamma\}} (-1)^{|\gamma \backslash \alpha|} \sum_{\{\delta \mid \delta \subseteq \beta\}} \tau_{\gamma \cup \delta}(-1)^{|\beta \backslash \delta|}$$

Suppose that the event $\{\gamma \subseteq \beta\}$ does not hold; then there is some element $i \in \gamma \backslash \beta$. Consequently. we can write

$$\sum_{\{\delta \mid \beta \subseteq \delta\}} \tau_{\gamma \cup \delta}(-1)^{|\delta \backslash \beta|} = \sum_{\{\delta \mid \beta \subseteq \delta, i \notin \delta\}} \tau_{\gamma \cup \delta}\big\{(-1)^{|\delta \backslash \beta|} + (-1)^{|\{\delta \cup \{i\}\} \backslash \beta|}\big\}$$
$$= 0.$$

Otherwise, if $\gamma \subseteq \beta$, then we have

$$\sum_{\{\delta \mid \beta \subseteq \delta\}} \tau_{\gamma \cup \delta}(-1)^{|b \backslash \beta|} = \sum_{\{\delta \mid \beta \subseteq \delta\}} \tau_\delta \, (-1)^{|\delta \backslash \beta|} =: \, \Psi(\beta). \qquad \text{(E.7)}$$

Therefore, we have

$$\sum_{\gamma,\delta} B(\alpha,\gamma)\big(N_m[\tau]\big)_{\gamma\delta} B(\delta,\beta) = \Psi(\beta) \sum_{\{\gamma \mid \alpha \subseteq \gamma\}} (-1)^{|\gamma \backslash \alpha|} \mathbb{I}[\gamma \subseteq \beta]$$
$$= \Psi(\beta) \sum_{\{\gamma \mid \alpha \subseteq \gamma \subseteq \beta\}} (-1)^{|\gamma \backslash \alpha|}$$
$$= \begin{cases} \Psi(\beta) & \text{if } \alpha = \beta \\ 0 & \text{otherwise,} \end{cases}$$

where the final inequality uses Equation (E.4). Thus, we have shown that the matrix $BN_m[\tau]B^T$ is diagonal with entries $\Psi(\beta)$. If $N_m[\tau] \succeq 0$, then by the invertibility of B, we are guaranteed that $\Psi(\beta) \geq 0$. It remains to show that $\sum_\beta \Psi(\beta) = 1$. Recall that $\tau_\emptyset = 1$ by definition. From the definition (E.7), we have

$$\sum_\beta \Psi(\beta) = \sum_\beta \sum_{\{\delta \mid \beta \subseteq \delta\}} \tau_\delta \, (-1)^{|\delta \backslash \beta|}$$
$$= \sum_\gamma \tau_\gamma \Big\{ \sum_{\{\beta \mid \emptyset \subseteq \beta \subseteq \gamma\}} (-1)^{|\gamma \backslash \beta|} \Big\}$$
$$= \tau_\emptyset$$
$$= 1,$$

where we have again used Equation (E.4) in moving from the third to fourth lines.

E.3   Conversion to a Pairwise Markov Random Field

In this appendix, we describe how any Markov random field with discrete random variables can be converted to an equivalent pairwise form (i.e., with interactions only between pairs of variables). To illustrate the general principle, it suffices to show how to convert a compatibility function ψ_{123} defined on a triplet $\{x_1, x_2, x_3\}$ of random variables into a pairwise form. To do so, we introduce an auxiliary node A, and associate with it random variable z that takes values in the Cartesian product space $\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$. In this way, each configuration of z can be identified with a triplet (z_1, z_2, z_3). For each $s \in \{1, 2, 3\}$, we define a pairwise compatibility function ψ_{As}, corresponding to the interaction between z and x_s, by $\psi_{As}(z, x_s) := [\psi_{123}(z_1, z_2, z_3)]^{1/3} \mathbb{I}[z_s = x_s]$. (The purpose of the $1/3$ power is to incorporate ψ_{123} with the correct exponent.) With this definition, it is straightforward to verify that the equivalence

$$\psi_{123}(x_1, x_2, x_3) = \sum_z \prod_{s=1}^{3} \psi_{As}(z, x_s)$$

holds, so that our augmented model faithfully captures the interaction defined on the triplet $\{x_1, x_2, x_3\}$.

References

[1] A. Agresti, *Categorical Data Analysis*. New York: Wiley, 2002.

[2] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Transactions on Information Theory*, vol. 46, pp. 325–343, 2000.

[3] N. I. Akhiezer, *The Classical Moment Problem and Some Related Questions in Analysis*. New York: Hafner Publishing Company, 1966.

[4] S. Amari, "Differential geometry of curved exponential families — curvatures and information loss," *Annals of Statistics*, vol. 10, no. 2, pp. 357–385, 1982.

[5] S. Amari and H. Nagaoka, *Methods of Information Geometry*. Providence, RI: AMS, 2000.

[6] G. An, "A note on the cluster variation method," *Journal of Statistical Physics*, vol. 52, no. 3, pp. 727–734, 1988.

[7] A. Bandyopadhyay and D. Gamarnik, "Counting without sampling: New algorithms for enumeration problems wiusing statistical physics," in *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms*, 2006.

[8] O. Banerjee, L. El Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data," *Journal of Machine Learning Research*, vol. 9, pp. 485–516, 2008.

[9] F. Barahona and M. Groetschel, "On the cycle polytope of a binary matroid," *Journal on Combination Theory, Series B*, vol. 40, pp. 40–62, 1986.

[10] D. Barber and W. Wiegerinck, "Tractable variational structures for approximating graphical models," in *Advances in Neural Information Processing Systems*, pp. 183–189, Cambridge, MA: MIT Press, 1999.

[11] O. E. Barndorff-Nielsen, *Information and Exponential Families*. Chichester, UK: Wiley, 1978.

[12] R. J. Baxter, *Exactly Solved Models in Statistical Mechanics*. New York: Academic Press, 1982.

[13] M. Bayati, C. Borgs, J. Chayes, and R. Zecchina, "Belief-propagation for weighted b-matchings on arbitrary graphs and its relation to linear programs with integer solutions," Technical Report arXiv: 0709 1190, Microsoft Research, 2007.

[14] M. Bayati and C. Nair, "A rigorous proof of the cavity method for counting matchings," in *Proceedings of the Allerton Conference on Control, Communication and Computing*, Monticello, IL, 2007.

[15] M. Bayati, D. Shah, and M. Sharma, "Maximum weight matching for max-product belief propagation," in *International Symposium on Information Theory*, Adelaide, Australia, 2005.

[16] M. J. Beal, "Variational algorithms for approximate Bayesian inference," PhD thesis, Gatsby Computational Neuroscience Unit, University College, London, 2003.

[17] C. Berge, *The Theory of Graphs and its Applications*. New York: Wiley, 1964.

[18] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Transactions on Information Theory*, vol. 24, pp. 384–386, 1978.

[19] U. Bertele and F. Brioschi, *Nonserial Dynamic Programming*. New York: Academic Press, 1972.

[20] D. P. Bertsekas, *Dynamic Programming and Stochastic Control*. Vol. 1. Belmont, MA: Athena Scientific, 1995.

[21] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1995.

[22] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Methods*. Belmont, MA: Athena Scientific, 1998.

[23] D. P. Bertsekas, *Convex Analysis and Optimization*. Belmont, MA: Athena Scientific, 2003.

[24] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*. Belmont, MA: Athena Scientific, 1997.

[25] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistical Society, Series B*, vol. 36, pp. 192–236, 1974.

[26] J. Besag, "Statistical analysis of non-lattice data," *The Statistician*, vol. 24, no. 3, pp. 179–195, 1975.

[27] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society, Series B*, vol. 48, no. 3, pp. 259–279, 1986.

[28] J. Besag and P. J. Green, "Spatial statistics and Bayesian computation," *Journal of the Royal Statistical Society, Series B*, vol. 55, no. 1, pp. 25–37, 1993.

[29] H. A. Bethe, "Statistics theory of superlattices," *Proceedings of Royal Society London, Series A*, vol. 150, no. 871, pp. 552–575, 1935.

[30] P. J. Bickel and K. A. Doksum, *Mathematical statistics: basic ideas and selected topics*. Upper Saddle River, N.J.: Prentice Hall, 2001.

[31] D. M. Blei and M. I. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian Analysis*, vol. 1, pp. 121–144, 2005.

[32] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[33] H. Bodlaender, "A tourist guide through treewidth," *Acta Cybernetica*, vol. 11, pp. 1–21, 1993.

[34] B. Bollobás, *Graph Theory: An Introductory Course*. New York: Springer-Verlag, 1979.

[35] B. Bollobás, *Modern Graph Theory*. New York: Springer-Verlag, 1998.

[36] E. Boros, Y. Crama, and P. L. Hammer, "Upper bounds for quadratic 0-1 maximization," *Operations Research Letters*, vol. 9, pp. 73–79, 1990.

[37] E. Boros and P. L. Hammer, "Pseudo-boolean optimization," *Discrete Applied Mathematics*, vol. 123, pp. 155–225, 2002.

[38] J. Borwein and A. Lewis, *Convex Analysis*. New York: Springer-Verlag, 1999.

[39] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[40] X. Boyen and D. Koller, "Tractable inference for complex stochastic processes," in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pp. 33–42, San Francisco, CA: Morgan Kaufmann, 1998.

[41] A. Braunstein, M. Mézard, and R. Zecchina, "Survey propagation: An algorithm for satisfiability," Technical Report, arXiv:cs.CC/02122002 v2, 2003.

[42] L. M. Bregman, "The relaxation method for finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, pp. 191–204, 1967.

[43] L. D. Brown, *Fundamentals of Statistical Exponential Families*. Hayward, CA: Institute of Mathematical Statistics, 1986.

[44] C. B. Burge and S. Karlin, "Finding the genes in genomic DNA," *Current Opinion in Structural Biology*, vol. 8, pp. 346–354, 1998.

[45] Y. Censor and S. A. Zenios, *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford: Oxford University Press, 1988.

[46] M. Cetin, L. Chen, J. W. Fisher, A. T. Ihler, R. L. Moses, M. J. Wainwright, and A. S. Willsky, "Distributed fusion in sensor networks," *IEEE Signal Processing Magazine*, vol. 23, pp. 42–55, 2006.

[47] D. Chandler, *Introduction to Modern Statistical Mechanics*. Oxford: Oxford University Press, 1987.

[48] C. Chekuri, S. Khanna, J. Naor, and L. Zosin, "A linear programming formulation and approximation algorithms for the metric labeling problem," *SIAM Journal on Discrete Mathematics*, vol. 18, no. 3, pp. 608–625, 2005.

[49] L. Chen, M. J. Wainwright, M. Cetin, and A. Willsky, "Multitarget-multisensor data association using the tree-reweighted max-product algorithm," in *SPIE Aerosense Conference*, Orlando, FL, 2003.

[50] M. Chertkov and V. Y. Chernyak, "Loop calculus helps to improve belief propagation and linear programming decoding of LDPC codes," in *Proceedings of the Allerton Conference on Control, Communication and Computing*, Monticello, IL, 2006.

[51] M. Chertkov and V. Y. Chernyak, "Loop series for discrete statistical models on graphs," *Journal of Statistical Mechanics*, p. P06009, 2006.

[52] M. Chertkov and V. Y. Chernyak, "Loop calculus helps to improve belief propagation and linear programming decodings of low density parity check codes," in *Proceedings of the Allerton Conference on Control, Communication and Computing*, Monticello, IL, 2007.

[53] M. Chertkov and M. G. Stepanov, "An efficient pseudo-codeword search algorithm for linear programming decoding of LDPC codes," Technical Report arXiv:cs.IT/0601113, Los Alamos National Laboratories, 2006.

[54] S. Chopra, "On the spanning tree polyhedron," *Operations Research Letters*, vol. 8, pp. 25–29, 1989.

[55] S. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pp. 151–158, 1971.

[56] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1990.

[57] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: John Wiley and Sons, 1991.

[58] G. Cross and A. Jain, "Markov random field texture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 25–39, 1983.

[59] I. Csiszár, "A geometric interpretation of Darroch and Ratcliff's generalized iterative scaling," *Annals of Statistics*, vol. 17, no. 3, pp. 1409–1413, 1989.

[60] I. Csiszár and G. Tusnády, "Information geometry and alternating minimization procedures," *Statistics and Decisions, Supplemental Issue 1*, pp. 205–237, 1984.

[61] J. N. Darroch and D. Ratcliff, "Generalized iterative scaling for log-linear models," *Annals of Mathematical Statistics*, vol. 43, pp. 1470–1480, 1972.

[62] C. Daskalakis, A. G. Dimakis, R. M. Karp, and M. J. Wainwright, "Probabilistic analysis of linear programming decoding," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3565–3578, 2008.

[63] A. d'Aspremont, O. Banerjee, and L. El Ghaoui, "First order methods for sparse covariance selection," *SIAM Journal on Matrix Analysis and its Applications*, vol. 30, no. 1, pp. 55–66, 2008.

[64] J. Dauwels, H. A. Loeliger, P. Merkli, and M. Ostojic, "On structured-summary propagation, LFSR synchronization, and low-complexity trellis decoding," in *Proceedings of the Allerton Conference on Control, Communication and Computing*, Monticello, IL, 2003.

[65] A. P. Dawid, "Applications of a general propagation algorithm for probabilistic expert systems," *Statistics and Computing*, vol. 2, pp. 25–36, 1992.

[66] R. Dechter, *Constraint Processing*. San Francisco, CA: Morgan Kaufmann, 2003.

[67] J. W. Demmel, *Applied Numerical Linear Algebra*. Philadelphia, PA: SIAM, 1997.

[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*, vol. 39, pp. 1–38, 1977.

[69] M. Deza and M. Laurent, *Geometry of Cuts and Metric Embeddings*. New York: Springer-Verlag, 1997.

[70] A. G. Dimakis and M. J. Wainwright, "Guessing facets: Improved LP decoding and polytope structure," in *International Symposium on Information Theory*, Seattle, WA, 2006.

[71] M. Dudik, S. J. Phillips, and R. E. Schapire, "Maximum entropy density estimation with generalized regularization and an application to species distribution modeling," *Journal of Machine Learning Research*, vol. 8, pp. 1217–1260, 2007.

[72] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, eds., *Biological Sequence Analysis.* Cambridge, UK: Cambridge University Press, 1998.

[73] J. Edmonds, "Matroids and the greedy algorithm," *Mathematical Programming*, vol. 1, pp. 127–136, 1971.

[74] B. Efron, "The geometry of exponential families," *Annals of Statistics*, vol. 6, pp. 362–376, 1978.

[75] G. Elidan, I. McGraw, and D. Koller, "Residual belief propagation: Informed scheduling for asynchronous message-passing," in *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, Arlington, VA: AUAI Press, 2006.

[76] J. Feldman, D. R. Karger, and M. J. Wainwright, "Linear programming-based decoding of turbo-like codes and its relation to iterative approaches," in *Proceedings of the Allerton Conference on Control, Communication and Computing*, Monticello, IL, 2002.

[77] J. Feldman, T. Malkin, R. A. Servedio, C. Stein, and M. J. Wainwright, "LP decoding corrects a constant fraction of errors," *IEEE Transactions on Information Theory*, vol. 53, no. 1, pp. 82–89, 2007.

[78] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Transactions on Information Theory*, vol. 51, pp. 954–972, 2005.

[79] J. Felsenstein, "Evolutionary trees from DNA sequences: A maximum likelihood approach," *Journal of Molecular Evolution*, vol. 17, pp. 368–376, 1981.

[80] S. E. Fienberg, "Contingency tables and log-linear models: Basic results and new developments," *Journal of the American Statistical Association*, vol. 95, no. 450, pp. 643–647, 2000.

[81] M. E. Fisher, "On the dimer solution of planar Ising models," *Journal of Mathematical Physics*, vol. 7, pp. 1776–1781, 1966.

[82] G. D. Forney, Jr., "The Viterbi algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268–277, 1973.

[83] G. D. Forney, Jr., R. Koetter, F. R. Kschischang, and A. Reznick, "On the effective weights of pseudocodewords for codes defined on graphs with cycles," in *Codes, Systems and Graphical Models*, pp. 101–112, New York: Springer, 2001.

[84] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *International Journal on Computer Vision*, vol. 40, no. 1, pp. 25–47, 2000.

[85] B. J. Frey, R. Koetter, and N. Petrovic, "Very loopy belief propagation for unwrapping phase images," in *Advances in Neural Information Processing Systems*, pp. 737–743, Cambridge, MA: MIT Press, 2001.

[86] B. J. Frey, R. Koetter, and A. Vardy, "Signal-space characterization of iterative decoding," *IEEE Transactions on Information Theory*, vol. 47, pp. 766–781, 2001.

[87] R. G. Gallager, *Low-Density Parity Check Codes*. Cambridge, MA: MIT Press, 1963.

[88] A. E. Gelfand and A. F. M. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American Statistical Association*, vol. 85, pp. 398–409, 1990.

[89] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

[90] H. O. Georgii, *Gibbs Measures and Phase Transitions*. New York: De Gruyter, 1988.

[91] Z. Ghahramani and M. J. Beal, "Propagation algorithms for variational Bayesian learning," in *Advances in Neural Information Processing Systems*, pp. 507–513, Cambridge, MA: MIT Press, 2001.

[92] Z. Ghahramani and M. I. Jordan, "Factorial hidden Markov models," *Machine Learning*, vol. 29, pp. 245–273, 1997.

[93] W. Gilks, S. Richardson, and D. Spiegelhalter, eds., *Markov Chain Monte Carlo in Practice*. New York: Chapman and Hall, 1996.

[94] A. Globerson and T. Jaakkola, "Approximate inference using planar graph decomposition," in *Advances in Neural Information Processing Systems*, pp. 473–480, Cambridge, MA: MIT Press, 2006.

[95] A. Globerson and T. Jaakkola, "Approximate inference using conditional entropy decompositions," in *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, San Juan, Puerto Rico, 2007.

[96] A. Globerson and T. Jaakkola, "Convergent propagation algorithms via oriented trees," in *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, Arlington, VA: AUAI Press, 2007.

[97] A. Globerson and T. Jaakkola, "Fixing max-product: Convergent message passing algorithms for MAP, LP-relaxations," in *Advances in Neural Information Processing Systems*, pp. 553–560, Cambridge, MA: MIT Press, 2007.

[98] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM*, vol. 42, pp. 1115–1145, 1995.

[99] G. Golub and C. Van Loan, *Matrix Computations*. Baltimore: Johns Hopkins University Press, 1996.

[100] V. Gómez, J. M. Mooij, and H. J. Kappen, "Truncating the loop series expansion for BP," *Journal of Machine Learning Research*, vol. 8, pp. 1987–2016, 2007.

[101] D. M. Greig, B. T. Porteous, and A. H. Seheuly, "Exact maximum a posteriori estimation for binary images," *Journal of the Royal Statistical Society B*, vol. 51, pp. 271–279, 1989.

[102] G. R. Grimmett, "A theorem about random fields," *Bulletin of the London Mathematical Society*, vol. 5, pp. 81–84, 1973.

[103] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*. Berlin: Springer-Verlag, 1993.

[104] M. Grötschel and K. Truemper, "Decomposition and optimization over cycles in binary matroids," *Journal on Combination Theory, Series B*, vol. 46, pp. 306–337, 1989.

[105] P. L. Hammer, P. Hansen, and B. Simeone, "Roof duality, complementation, and persistency in quadratic 0-1 optimization," *Mathematical Programming*, vol. 28, pp. 121–155, 1984.

[106] J. M. Hammersley and P. Clifford, "Markov fields on finite graphs and lattices," Unpublished, 1971.

[107] M. Hassner and J. Sklansky, "The use of Markov random fields as models of texture," *Computer Graphics and Image Processing*, vol. 12, pp. 357–370, 1980.

[108] T. Hazan and A. Shashua, "Convergent message-passing algorithms for inference over general graphs with convex free energy," in *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, Arlington, VA: AUAI Press, 2008.

[109] T. Heskes, "On the uniqueness of loopy belief propagation fixed points," *Neural Computation*, vol. 16, pp. 2379–2413, 2004.

[110] T. Heskes, "Convexity arguments for efficient minimization of the Bethe and Kikuchi free energies," *Journal of Artificial Intelligence Research*, vol. 26, pp. 153–190, 2006.

[111] T. Heskes, K. Albers, and B. Kappen, "Approximate inference and constrained optimization," in *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, pp. 313–320, San Francisco, CA: Morgan Kaufmann, 2003.

[112] J. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms.* Vol. 1, New York: Springer-Verlag, 1993.

[113] J. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis.* Springer-Verlag: New York, 2001.

[114] R. A. Horn and C. R. Johnson, *Matrix Analysis.* Cambridge, UK: Cambridge University Press, 1985.

[115] J. Hu, H.-A. Loeliger, J. Dauwels, and F. Kschischang, "A general computation rule for lossy summaries/messages with examples from equalization," in *Proceedings of the Allerton Conference on Control, Communication and Computing*, Monticello, IL, 2005.

[116] B. Huang and T. Jebara, "Loopy belief propagation for bipartite maximum weight b-matching," in *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, San Juan, Puerto Rico, 2007.

[117] X. Huang, A. Acero, H.-W. Hon, and R. Reddy, *Spoken Language Processing.* New York: Prentice Hall, 2001.

[118] A. Ihler, J. Fisher, and A. S. Willsky, "Loopy belief propagation: Convergence and effects of message errors," *Journal of Machine Learning Research*, vol. 6, pp. 905–936, 2005.

[119] E. Ising, "Beitrag zur theorie der ferromagnetismus," *Zeitschrift für Physik*, vol. 31, pp. 253–258, 1925.

[120] T. S. Jaakkola, "Tutorial on variational approximation methods," in *Advanced Mean Field Methods: Theory and Practice*, (M. Opper and D. Saad, eds.), pp. 129–160, Cambridge, MA: MIT Press, 2001.

[121] T. S. Jaakkola and M. I. Jordan, "Improving the mean field approximation via the use of mixture distributions," in *Learning in Graphical Models*, (M. I. Jordan, ed.), pp. 105–161, MIT Press, 1999.

[122] T. S. Jaakkola and M. I. Jordan, "Variational probabilistic inference and the QMR-DT network," *Journal of Artificial Intelligence Research*, vol. 10, pp. 291–322, 1999.

[123] E. T. Jaynes, "Information theory and statistical mechanics," *Physical Review*, vol. 106, pp. 620–630, 1957.

[124] J. K. Johnson, D. M. Malioutov, and A. S. Willsky, "Lagrangian relaxation for MAP estimation in graphical models," in *Proceedings of the Allerton Conference on Control, Communication and Computing*, Monticello, IL, 2007.

[125] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, pp. 183–233, 1999.

[126] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Englewood Cliffs, NJ: Prentice Hall, 2000.

[127] H. Kappen and P. Rodriguez, "Efficient learning in Boltzmann machines using linear response theory," *Neural Computation*, vol. 10, pp. 1137–1156, 1998.

[128] H. J. Kappen and W. Wiegerinck, "Novel iteration schemes for the cluster variation method," in *Advances in Neural Information Processing Systems*, pp. 415–422, Cambridge, MA: MIT Press, 2002.

[129] D. Karger and N. Srebro, "Learning Markov networks: Maximum bounded tree-width graphs," in *Symposium on Discrete Algorithms*, pp. 392–401, 2001.

[130] S. Karlin and W. Studden, *Tchebycheff Systems, with Applications in Analysis and Statistics*. New York: Interscience Publishers, 1966.

[131] R. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, pp. 85–103, New York: Plenum Press, 1972.

[132] P. W. Kastelyn, "Dimer statistics and phase transitions," *Journal of Mathematical Physics*, vol. 4, pp. 287–293, 1963.

[133] R. Kikuchi, "The theory of cooperative phenomena," *Physical Review*, vol. 81, pp. 988–1003, 1951.

[134] S. Kim and M. Kojima, "Second order cone programming relaxation of non-convex quadratic optimization problems," Technical report, Tokyo Institute of Technology, July 2000.

[135] J. Kleinberg and E. Tardos, "Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields," *Journal of the ACM*, vol. 49, pp. 616–639, 2002.

[136] R. Koetter and P. O. Vontobel, "Graph-covers and iterative decoding of finite length codes," in *Proceedings of the 3rd International Symposium on Turbo Codes*, pp. 75–82, Brest, France, 2003.

[137] V. Kolmogorov, "Convergent tree-reweighted message-passing for energy minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1568–1583, 2006.

[138] V. Kolmogorov and M. J. Wainwright, "On optimality properties of tree-reweighted message-passing," in *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pp. 316–322, Arlington, VA: AUAI Press, 2005.

[139] N. Komodakis, N. Paragios, and G. Tziritas, "MRF optimization via dual decomposition: Message-passing revisited," in *International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007.

[140] V. K. Koval and M. I. Schlesinger, "Two-dimensional programming in image analysis problems," *USSR Academy of Science, Automatics and Telemechanics*, vol. 8, pp. 149–168, 1976.

[141] A. Krogh, B. Larsson, G. von Heijne, and E. L. L. Sonnhammer, "Predicting transmembrane protein topology with a hidden Markov model: Application to complete genomes," *Journal of Molecular Biology*, vol. 305, no. 3, pp. 567–580, 2001.

[142] F. R. Kschischang and B. J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," *IEEE Selected Areas in Communications*, vol. 16, no. 2, pp. 219–230, 1998.

[143] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[144] A. Kulesza and F. Pereira, "Structured learning with approximate inference," in *Advances in Neural Information Processing Systems*, pp. 785–792, Cambridge, MA: MIT Press, 2008.

[145] P. Kumar, V. Kolmogorov, and P. H. S. Torr, "An analysis of convex relaxations for MAP estimation," in *Advances in Neural Information Processing Systems*, pp. 1041–1048, Cambridge, MA: MIT Press, 2008.

[146] P. Kumar, P. H. S. Torr, and A. Zisserman, "Solving Markov random fields using second order cone programming," *IEEE Conference of Computer Vision and Pattern Recognition*, pp. 1045–1052, 2006.

[147] J. B. Lasserre, "An explicit equivalent positive semidefinite program for non-linear 0–1 programs," *SIAM Journal on Optimization*, vol. 12, pp. 756–769, 2001.

[148] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 796–817, 2001.

[149] M. Laurent, "Semidefinite relaxations for Max-Cut," in *The Sharpest Cut: Festschrift in Honor of M. Padberg's 60th Birthday*, New York: MPS-SIAM Series in Optimization, 2002.

[150] M. Laurent, "A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for 0-1 programming," *Mathematics of Operations Research*, vol. 28, pp. 470–496, 2003.

[151] S. L. Lauritzen, *Lectures on Contingency Tables.* Department of Mathematics, Aalborg University, 1989.

[152] S. L. Lauritzen, "Propagation of probabilities, means and variances in mixed graphical association models," *Journal of the American Statistical Association*, vol. 87, pp. 1098–1108, 1992.

[153] S. L. Lauritzen, *Graphical Models.* Oxford: Oxford University Press, 1996.

[154] 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, Series B*, vol. 50, pp. 155–224, 1988.

[155] M. A. R. Leisink and H. J. Kappen, "Learning in higher order Boltzmann machines using linear response," *Neural Networks*, vol. 13, pp. 329–335, 2000.

[156] M. A. R. Leisink and H. J. Kappen, "A tighter bound for graphical models," in *Advances in Neural Information Processing Systems*, pp. 266–272, Cambridge, MA: MIT Press, 2001.

[157] H. A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, pp. 28–41, 2004.

[158] L. Lovász, "Submodular functions and convexity," in *Mathematical Programming: The State of the Art*, (A. Bachem, M. Grötschel, and B. Korte, eds.), pp. 235–257, New York: Springer-Verlag, 1983.

[159] L. Lovász and A. Schrijver, "Cones of matrices, set-functions and 0-1 optimization," *SIAM Journal of Optimization*, vol. 1, pp. 166–190, 1991.

[160] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. Spielman, "Improved low-density parity check codes using irregular graphs," *IEEE Transactions on Information Theory*, vol. 47, pp. 585–598, 2001.

[161] D. M. Malioutov, J. M. Johnson, and A. S. Willsky, "Walk-sums and belief propagation in Gaussian graphical models," *Journal of Machine Learning Research*, vol. 7, pp. 2013–2064, 2006.

[162] E. Maneva, E. Mossel, and M. J. Wainwright, "A new look at survey propagation and its generalizations," *Journal of the ACM*, vol. 54, no. 4, pp. 2–41, 2007.

[163] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999.

[164] S. Maybeck, *Stochastic Models, Estimation, and Control*. New York: Academic Press, 1982.

[165] J. McAuliffe, L. Pachter, and M. I. Jordan, "Multiple-sequence functional annotation and the generalized hidden Markov phylogeny," *Bioinformatics*, vol. 20, pp. 1850–1860, 2004.

[166] R. J. McEliece, D. J. C. McKay, and J. F. Cheng, "Turbo decoding as an instance of Pearl's belief propagation algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, no. 2, pp. 140–152, 1998.

[167] R. J. McEliece and M. Yildirim, "Belief propagation on partially ordered sets," in *Mathematical Theory of Systems and Networks*, (D. Gilliam and J. Rosenthal, eds.), Minneapolis, MN: Institute for Mathematics and its Applications, 2002.

[168] T. Meltzer, C. Yanover, and Y. Weiss, "Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation," in *International Conference on Computer Vision*, pp. 428–435, Silver Springs, MD: IEEE Computer Society, 2005.

[169] M. Mézard and A. Montanari, *Information, Physics and Computation*. Oxford: Oxford University Press, 2008.

[170] M. Mézard, G. Parisi, and R. Zecchina, "Analytic and algorithmic solution of random satisfiability problems," *Science*, vol. 297, p. 812, 2002.

[171] M. Mézard and R. Zecchina, "Random K-satisfiability: From an analytic solution to an efficient algorithm," *Physical Review E*, vol. 66, p. 056126, 2002.

[172] T. Minka, "Expectation propagation and approximate Bayesian inference," in *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pp. 362–369, San Francisco, CA: Morgan Kaufmann, 2001.

[173] T. Minka, "Power EP," Technical Report MSR-TR-2004-149, Microsoft Research, October 2004.

[174] T. Minka and Y. Qi, "Tree-structured approximations by expectation propagation," in *Advances in Neural Information Processing Systems*, pp. 193–200, Cambridge, MA: MIT Press, 2004.

[175] T. P. Minka, "A family of algorithms for approximate Bayesian inference," PhD thesis, MIT, January 2001.

[176] C. Moallemi and B. van Roy, "Convergence of the min-sum message-passing algorithm for quadratic optimization," Technical Report, Stanford University, March 2006.

[177] C. Moallemi and B. van Roy, "Convergence of the min-sum algorithm for convex optimization," Technical Report, Stanford University, May 2007.

[178] J. M. Mooij and H. J. Kappen, "Sufficient conditions for convergence of loopy belief propagation," in *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pp. 396–403, Arlington, VA: AUAI Press, 2005.

[179] R. Neal and G. E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, (M. I. Jordan, ed.), Cambridge, MA: MIT Press, 1999.

[180] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. New York: Wiley-Interscience, 1999.

[181] Y. Nesterov, "Semidefinite relaxation and non-convex quadratic optimization," *Optimization methods and software*, vol. 12, pp. 1–20, 1997.

[182] M. Opper and D. Saad, "Adaptive TAP equations," in *Advanced Mean Field Methods: Theory and Practice*, (M. Opper and D. Saad, eds.), pp. 85–98, Cambridge, MA: MIT Press, 2001.

[183] M. Opper and O. Winther, "Gaussian processes for classification: Mean field algorithms," *Neural Computation*, vol. 12, no. 11, pp. 2177–2204, 2000.

[184] M. Opper and O. Winther, "Tractable approximations for probabilistic models: The adaptive Thouless-Anderson-Palmer approach," *Physical Review Letters*, vol. 64, p. 3695, 2001.

[185] M. Opper and O. Winther, "Expectation-consistent approximate inference," *Journal of Machine Learning Research*, vol. 6, pp. 2177–2204, 2005.

[186] J. G. Oxley, *Matroid Theory*. Oxford: Oxford University Press, 1992.

[187] M. Padberg, "The Boolean quadric polytope: Some characteristics, facets and relatives," *Mathematical Programming*, vol. 45, pp. 139–172, 1989.

[188] P. Pakzad and V. Anantharam, "Iterative algorithms and free energy minimization," in *Annual Conference on Information Sciences and Systems*, Princeton, NJ, 2002.

[189] P. Pakzad and V. Anantharam, "Estimation and marginalization using Kikuchi approximation methods," *Neural Computation*, vol. 17, no. 8, pp. 1836–1873, 2005.

[190] G. Parisi, *Statistical Field Theory*. New York: Addison-Wesley, 1988.

[191] P. Parrilo, "Semidefinite programming relaxations for semialgebraic problems," *Mathematical Programming, Series B*, vol. 96, pp. 293–320, 2003.

[192] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Francisco, CA: Morgan Kaufmann, 1988.

[193] J. S. Pedersen and J. Hein, "Gene finding with a hidden Markov model of genome structure and evolution," *Bioinformatics*, vol. 19, pp. 219–227, 2003.

[194] P. Pevzner, *Computational Molecular Biology: An Algorithmic Approach.* Cambridge, MA: MIT Press, 2000.

[195] T. Plefka, "Convergence condition of the TAP equation for the infinite-ranged Ising model," *Journal of Physics A*, vol. 15, no. 6, pp. 1971–1978, 1982.

[196] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition.* Englewood Cliffs, NJ: Prentice Hall, 1993.

[197] P. Ravikumar, A. Agarwal, and M. J. Wainwright, "Message-passing for graph-structured linear programs: Proximal projections, convergence and rounding schemes," in *International Conference on Machine Learning*, pp. 800–807, New York: ACM Press, 2008.

[198] P. Ravikumar and J. Lafferty, "Quadratic programming relaxations for metric labeling and Markov random field map estimation," *International Conference on Machine Learning*, pp. 737–744, 2006.

[199] T. Richardson and R. Urbanke, "The capacity of low-density parity check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, pp. 599–618, 2001.

[200] T. Richardson and R. Urbanke, *Modern Coding Theory.* Cambridge, UK: Cambridge University Press, 2008.

[201] B. D. Ripley, *Spatial Statistics.* New York: Wiley, 1981.

[202] C. P. Robert and G. Casella, *Monte Carlo statistical methods. Springer texts in statistics*, New York, NY: Springer-Verlag, 1999.

[203] G. Rockafellar, *Convex Analysis.* Princeton, NJ: Princeton University Press, 1970.

[204] T. G. Roosta, M. J. Wainwright, and S. S. Sastry, "Convergence analysis of reweighted sum-product algorithms," *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4293–4305, 2008.

[205] P. Rusmevichientong and B. Van Roy, "An analysis of turbo decoding with Gaussian densities," in *Advances in Neural Information Processing Systems*, pp. 575–581, Cambridge, MA: MIT Press, 2000.

[206] S. Sanghavi, D. Malioutov, and A. Willsky, "Linear programming analysis of loopy belief propagation for weighted matching," in *Advances in Neural Information Processing Systems*, pp. 1273–1280, Cambridge, MA: MIT Press, 2007.

[207] S. Sanghavi, D. Shah, and A. Willsky, "Message-passing for max-weight independent set," in *Advances in Neural Information Processing Systems*, pp. 1281–1288, Cambridge, MA: MIT Press, 2007.

[208] L. K. Saul and M. I. Jordan, "Boltzmann chains and hidden Markov models," in *Advances in Neural Information Processing Systems*, pp. 435–442, Cambridge, MA: MIT Press, 1995.

[209] L. K. Saul and M. I. Jordan, "Exploiting tractable substructures in intractable networks," in *Advances in Neural Information Processing Systems*, pp. 486–492, Cambridge, MA: MIT Press, 1996.

[210] M. I. Schlesinger, "Syntactic analysis of two-dimensional visual signals in noisy conditions," *Kibernetika*, vol. 4, pp. 113–130, 1976.

[211] A. Schrijver, *Theory of Linear and Integer Programming.* New York: Wiley-Interscience Series in Discrete Mathematics, 1989.

[212] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency.* New York: Springer-Verlag, 2003.

[213] M. Seeger, "Expectation propagation for exponential families," Technical Report, Max Planck Institute, Tuebingen, November 2005.

[214] M. Seeger, F. Steinke, and K. Tsuda, "Bayesian inference and optimal design in the sparse linear model," in *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, San Juan, Puerto Rico, 2007.

[215] P. D. Seymour, "Matroids and multi-commodity flows," *European Journal on Combinatorics*, vol. 2, pp. 257–290, 1981.

[216] G. R. Shafer and P. P. Shenoy, "Probability propagation," *Annals of Mathematics and Artificial Intelligence*, vol. 2, pp. 327–352, 1990.

[217] H. D. Sherali and W. P. Adams, "A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems," *SIAM Journal on Discrete Mathematics*, vol. 3, pp. 411–430, 1990.

[218] A. Siepel and D. Haussler, "Combining phylogenetic and hidden Markov models in biosequence analysis," in *Proceedings of the Seventh Annual International Conference on Computational Biology*, pp. 277–286, 2003.

[219] D. Sontag and T. Jaakkola, "New outer bounds on the marginal polytope," in *Advances in Neural Information Processing Systems*, pp. 1393–1400, Cambridge, MA: MIT Press, 2007.

[220] T. P. Speed and H. T. Kiiveri, "Gaussian Markov distributions over finite graphs," *Annals of Statistics*, vol. 14, no. 1, pp. 138–150, 1986.

[221] R. P. Stanley, *Enumerative Combinatorics.* Vol. 1, Cambridge, UK: Cambridge University Press, 1997.

[222] R. P. Stanley, *Enumerative Combinatorics.* Vol. 2, Cambridge, UK: Cambridge University Press, 1997.

[223] 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*, vol. 52, no. 11, pp. 3136–3150, 2004.

[224] E. B. Sudderth, M. J. Wainwright, and A. S. Willsky, "Loop series and Bethe variational bounds for attractive graphical models," in *Advances in Neural Information Processing Systems*, pp. 1425–1432, Cambridge, MA: MIT Press, 2008.

[225] C. Sutton and A. McCallum, "Piecewise training of undirected models," in *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pp. 568–575, San Francisco, CA: Morgan Kaufmann, 2005.

[226] C. Sutton and A. McCallum, "Improved dyamic schedules for belief propagation," in *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA: Morgan Kaufmann, 2007.

[227] R. Szeliski, R. Zabih, D. Scharstein, O. Veskler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "Comparative study of energy minimization methods for Markov random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 1068–1080, 2008.

[228] M. H. Taghavi and P. H. Siegel, "Adaptive linear programming decoding," in *International Symposium on Information Theory*, Seattle, WA, 2006.

[229] K. Tanaka and T. Morita, "Cluster variation method and image restoration problem," *Physics Letters A*, vol. 203, pp. 122–128, 1995.

[230] S. Tatikonda and M. I. Jordan, "Loopy belief propagation and Gibbs measures," in *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pp. 493–500, San Francisco, CA: Morgan Kaufmann, 2002.

[231] Y. W. Teh and M. Welling, "On improving the efficiency of the iterative proportional fitting procedure," in *Proceedings of the Ninth International Conference on Artificial Intelligence and Statistics*, Key West, FL, 2003.

[232] A. Thomas, A. Gutin, V. Abkevich, and A. Bansal, "Multilocus linkage analysis by blocked Gibbs sampling," *Statistics and Computing*, vol. 10, pp. 259–269, 2000.

[233] A. W. van der Vaart, *Asymptotic Statistics*. Cambridge, UK: Cambridge University Press, 1998.

[234] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, pp. 49–95, 1996.

[235] L. Vandenberghe, S. Boyd, and S. Wu, "Determinant maximization with linear matrix inequality constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 19, pp. 499–533, 1998.

[236] V. V. Vazirani, *Approximation Algorithms*. New York: Springer-Verlag, 2003.

[237] S. Verdú and H. V. Poor, "Abstract dynamic programming models under commutativity conditions," *SIAM Journal of Control and Optimization*, vol. 25, no. 4, pp. 990–1006, 1987.

[238] P. O. Vontobel and R. Koetter, "Lower bounds on the minimum pseudo-weight of linear codes," in *International Symposium on Information Theory*, Chicago, IL, 2004.

[239] P. O. Vontobel and R. Koetter, "Towards low-complexity linear-programming decoding," in *Proceedings of the International Conference on Turbo Codes and Related Topics*, Munich, Germany, 2006.

[240] M. J. Wainwright, "Stochastic processes on graphs with cycles: Geometric and variational approaches," PhD thesis, MIT, May 2002.

[241] M. J. Wainwright, "Estimating the "wrong" graphical model: Benefits in the computation-limited regime," *Journal of Machine Learning Research*, vol. 7, pp. 1829–1859, 2006.

[242] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "Tree-based reparameterization framework for analysis of sum-product and related algorithms," *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1120–1146, 2003.

[243] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "Tree-reweighted belief propagation algorithms and approximate ML, estimation by pseudomoment matching," in *Proceedings of the Ninth International Conference on Artificial Intelligence and Statistics*, 2003.

[244] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "Tree consistency and bounds on the max-product algorithm and its generalizations," *Statistics and Computing*, vol. 14, pp. 143–166, 2004.

[245] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "Exact MAP estimates via agreement on (hyper)trees: Linear programming and message-passing," *IEEE Transactions on Information Theory*, vol. 51, no. 11, pp. 3697–3717, 2005.

[246] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "A new class of upper bounds on the log partition function," *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2313–2335, 2005.

[247] M. J. Wainwright and M. I. Jordan, "Treewidth-based conditions for exactness of the Sherali-Adams and Lasserre relaxations," Technical Report 671, University of California, Berkeley, Department of Statistics, September 2004.

[248] M. J. Wainwright and M. I. Jordan, "Log-determinant relaxation for approximate inference in discrete Markov random fields," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2099–2109, 2006.

[249] Y. Weiss, "Correctness of local probability propagation in graphical models with loops," *Neural Computation*, vol. 12, pp. 1–41, 2000.

[250] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," in *Advances in Neural Information Processing Systems*, pp. 673–679, Cambridge, MA: MIT Press, 2000.

[251] Y. Weiss, C. Yanover, and T. Meltzer, "MAP estimation, linear programming, and belief propagation with convex free energies," in *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, Arlington, VA: AUAI Press, 2007.

[252] M. Welling, T. Minka, and Y. W. Teh, "Structured region graphs: Morphing EP into GBP," in *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pp. 609–614, Arlington, VA: AUAI Press, 2005.

[253] M. Welling and S. Parise, "Bayesian random fields: The Bethe-Laplace approximation," in *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, Arlington, VA: AUAI Press, 2006.

[254] M. Welling and Y. W. Teh, "Belief optimization: A stable alternative to loopy belief propagation," in *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pp. 554–561, San Francisco, CA: Morgan Kaufmann, 2001.

[255] M. Welling and Y. W. Teh, "Linear response for approximate inference," in *Advances in Neural Information Processing Systems*, pp. 361–368, Cambridge, MA: MIT Press, 2004.

[256] T. Werner, "A linear programming approach to max-sum problem: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 7, pp. 1165–1179, 2007.

[257] N. Wiberg, "Codes and decoding on general graphs," PhD thesis, University of Linkoping, Sweden, 1996.

[258] N. Wiberg, H. A. Loeliger, and R. Koetter, "Codes and iterative decoding on general graphs," *European Transactions on Telecommunications*, vol. 6, pp. 513–526, 1995.

[259] W. Wiegerinck, "Variational approximations between mean field theory and the junction tree algorithm," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pp. 626–633, San Francisco, CA: Morgan Kaufmann Publishers, 2000.

[260] W. Wiegerinck, "Approximations with reweighted generalized belief propaga-
tion," in *Proceedings of the Tenth International Workshop on Artificial Intel-
ligence and Statistics*, pp. 421–428, Barbardos, 2005.

[261] W. Wiegerinck and T. Heskes, "Fractional belief propagation," in *Advances in
Neural Information Processing Systems*, pp. 438–445, Cambridge, MA: MIT
Press, 2002.

[262] A. S. Willsky, "Multiresolution Markov models for signal and image process-
ing," *Proceedings of the IEEE*, vol. 90, no. 8, pp. 1396–1458, 2002.

[263] J. W. Woods, "Markov image modeling," *IEEE Transactions on Automatic
Control*, vol. 23, pp. 846–850, 1978.

[264] N. Wu, *The Maximum Entropy Method*. New York: Springer, 1997.

[265] C. Yanover, T. Meltzer, and Y. Weiss, "Linear programming relaxations
and belief propagation: An empirical study," *Journal of Machine Learning
Research*, vol. 7, pp. 1887–1907, 2006.

[266] C. Yanover, O. Schueler-Furman, and Y. Weiss, "Minimizing and learning
energy functions for side-chain prediction," in *Eleventh Annual Conference
on Research in Computational Molecular Biology*, pp. 381–395, San Francisco,
CA, 2007.

[267] J. S. Yedidia, "An idiosyncratic journey beyond mean field theory," in
Advanced Mean Field Methods: Theory and Practice, (M. Opper and D. Saad,
eds.), pp. 21–36, Cambridge, MA: MIT Press, 2001.

[268] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propaga-
tion," in *Advances in Neural Information Processing Systems*, pp. 689–695,
Cambridge, MA: MIT Press, 2001.

[269] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free energy
approximations and generalized belief propagation algorithms," *IEEE Trans-
actions on Information Theory*, vol. 51, no. 7, pp. 2282–2312, 2005.

[270] A. Yuille, "CCCP algorithms to minimize the Bethe and Kikuchi free energies:
Convergent alternatives to belief propagation," *Neural Computation*, vol. 14,
pp. 1691–1722, 2002.

[271] G. M. Ziegler, *Lectures on Polytopes*. New York: Springer-Verlag, 1995.