Graphical models, exponential families, and variational inference

Martin J. Wainwright
Department of Electrical Engineering
and Computer Science
University of California, Berkeley
wainwrig@eecs.berkeley.edu

Michael I. Jordan
Division of Computer Science
Department of Statistics
University of California, Berkeley
jordan@stat.berkeley.edu

September 21, 2003

Technical Report 649
Department of Statistics
University of California, Berkeley

Abstract

The formalism of probabilistic graphical models provides a unifying framework for the development of large-scale multivariate statistical models. Graphical models have become a focus of research in many applied statistical and computational fields, including bioinformatics, information theory, signal and image processing, information retrieval and 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 generating function and the entropy for exponential families, we develop general variational representations of the problems of computing marginal probabilities and modes. We describe how a wide variety of known computational algorithms—including mean field, sum-product, max-product and cluster variational techniques—can be understood in terms of exact or approximate forms of these variational representations. We also present novel convex relaxations based on the variational framework. 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., 11, 107]. Our focus in this paper is rather different: we present an alternative approach to 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 [45]. A similar development in the case of variational methodology would be of significant interest. In our view, the most promising avenue towards variational methodology tuned to statistics is to build on existing links between variational analysis and the exponential family of distributions [3, 5, 17, 35]. 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. Building on the foundation of conjugate duality, we develop a general variational representation for computing likelihoods and marginal probabilities in exponential families in Section 4. The bulk of the remainder of the paper—Sections 5 through 9—is devoted to the exploration of various relaxations of this exact variational principle, which in turn yield various algorithms for computing approximations to marginal probabilities. More specifically, we discuss *mean field theory* in Section 5, the *Bethe approximation* in Section 6, and general *cluster variational methods* based on hypertrees, including the *Kikuchi method*, in Section 7. All of these methods are based on non-convex optimization problems, which typically have multiple solutions. Sections 8 and 9 present convex relaxations of the variational principle that are also guaranteed to yield upper bounds on the log likelihood. Finally, in Section 10, we develop a variational formulation of the problem of computing modes of distributions, and again describe several relaxations of the exact principle.

The scope of this paper 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 paper.

2 Background

2.1 Graphical models

A graphical model consists of a collection of probability distributions[1] that factorize according to the structure of an underlying graph. A graph $G = (V, E)$ is formed by a collection of vertices V, and a collection of edges E. An edge consists of a pair of vertices, and may either be directed or undirected. Associated with each vertex $s \in V$ is a random variable x_s taking values in some set $\mathcal{X}_s$, which may either be continuous (e.g., $\mathcal{X}_s = \mathbb{R}$) or discrete (e.g., $\mathcal{X}_s = \{0, 1, \ldots, m-1\}$). For any subset A of the vertex set V, we define $x_A := \{x_s \mid s \in A\}$.

Directed graphical models: In the directed case, each edge is directed from parent to child. We let $\pi(s)$ denote the set of all parents of given node $s \in V$. (If s has no parents, then the set $\pi(s)$ should be understood to be empty.) With this notation, a *directed graphical model* consists of a collection of probability distributions that factorize in the following way:

$$p(\mathbf{x}) \quad = \quad \prod_{s \in V} p(x_s \mid x_{\pi(s)}). \tag{1}$$

It can be verified that our use of notation is consistent, in that $p(x_s \mid x_{\pi(s)})$ is, in fact, the conditional distribution for the global distribution $p(\mathbf{x})$ thus defined.

Undirected graphical models: In the undirected case, the probability distribution factorizes according to functions defined on the *cliques* of the graph (i.e., fully-connected subsets of V). In particular, associated with each clique C is a *compatibility function* $\psi_C : \mathcal{X}^n \rightarrow \mathbb{R}_+$ that depends

[1]Here we are using the terminology "distribution" loosely; our notation $p(\cdot)$ 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.

only on x_C. With this notation, an *undirected graphical model* (also known as a *Markov random field*) consists of a collection of distributions that factorize as follows:

$$p(\mathbf{x}) \quad = \quad \frac{1}{Z} \prod_C \psi_C(x_C), \tag{2}$$

where the product is taken over all cliques of the graph. The quantity Z is a constant chosen to ensure that the distribution is normalized. In contrast to the directed case (1), in general the compatibility functions ψ_C need not have any obvious or direct relation to local marginal distributions.

2.2 Conditional independence

Families of probability distributions as defined as in (1) or (2) also have a characterization in terms of conditional independencies among subsets of random variables. We only touch upon this characterization here, as it is not needed in the remainder of the paper; for a full treatment, we refer the interested reader to Lauritzen [68].

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 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) 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 [22].

An analogous result holds in the case of directed graphical models, with the only alteration being a different notion of reachability [68]. Once again, it is possible to establish an equivalence between the set of probability distributions specified by the directed factorization (1), and that defined in terms of a set of conditional independence assertions.

2.3 Inference problems and exact algorithms

Given a probability distribution $p(\cdot)$ defined by a graphical model, our focus will be solving one or more of the following *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 $\widehat{\mathbf{x}}$ in the set $\arg\max_{\mathbf{x} \in \mathcal{X}^n} p(\mathbf{x})$).

From a computational perspective, problems (a) and (b) are essentially equivalent, since they both involve summing or integrating over a subset of random variables. 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 $\mathbf{x} \in \mathcal{X}^n$, where $\mathcal{X}_s = \{0, 1, \ldots, m-1\}$ 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 $\{\mathbf{x}' \mid x'_s = x_s\}$. Since this set has m^{n-1} elements, it is clear that a brute force approach will rapidly become intractable. Even with binary variables ($m = 2$) and a graph with $n \approx 400$ nodes (a modest size for many applications), this summation involves more terms than atoms in the visible universe. 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[2] and typically harder, since they require computing a large number of integrals.

For graphs without cycles—also known as *trees*—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.

Before turning to these algorithmic issues, however, we present some examples of applications of graphical models.

2.4 Applications

This section illustrates the use of graphical models in various areas, including the general area of Bayesian hierarchical modeling, as well as specific applications in bioinformatics, speech and language processing, image processing and error-correcting coding.

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 (i.e., "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

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

distributions defines the joint probability; this factorization is simply a a particular instance of equation (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 [107].

2.4.2 Bioinformatics and language

Many classical models in the fields of bioinformatics and language processing 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 bioinformatics and language processing, both classical and recent.

Sequential data obviously play a central role in bioinformatics and language, and the workhorse underlying the modeling of sequential data is the same in both domains—namely, the *hidden Markov model* (HMM) shown in Figure 1(a). The HMM is in essence a dynamical version of a finite mixture

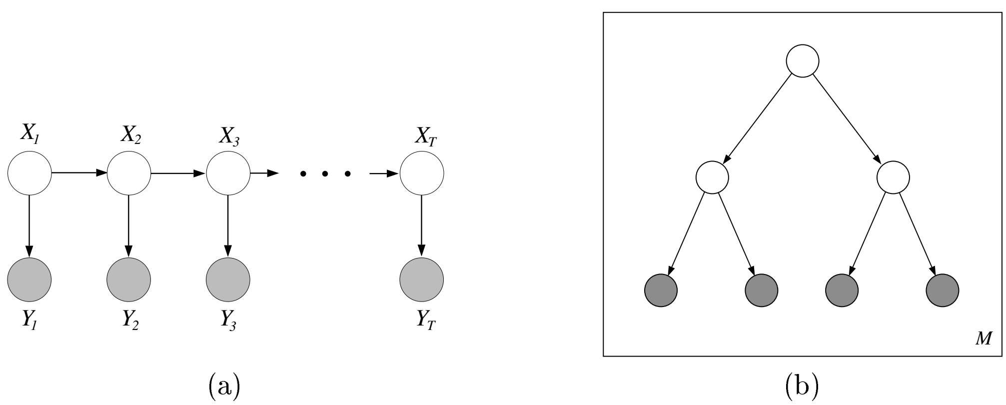


Figure 1. (a) The graphical model representation of a generic hidden Markov model. The shaded nodes are the observations and the unshaded nodes are the hidden state variables. The latter form a Markov chain, in which X_s is independent of X_u conditional on X_t, where $s < t < u$. (b) The graphical model 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.

model, in which observations are generated conditionally on a underlying latent ("hidden") state variable. The state variables, which are generally taken to be multinomial random variables,[3] form

[3]The graphical model in Figure 1(a) is also a representation of the state-space model underlying Kalman filtering and smoothing, where the state variable is a 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.

a Markov chain.

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 $p(x_t, x_{t+1} \mid y)$ and $p(x_t \mid y)$. In the HMM context, this algorithm is often referred to as the *forward-backward algorithm* [88]. 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* [40].

Gene-finding provides an important example of the application of the HMM [33]. 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. HMMs are currently the methodology of choice for attacking this problem, with designers choosing states and state transitions to reflect biological knowledge concerning gene structure [18].

HMMs 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 [62].

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 [73]. 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 [55]. 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.

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 1(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" [38]. This recursion is a special case of the junction tree algorithm.

Figure 2 gives examples of more complex graphical models that are currently being explored in bioinformatics and language processing. Figure 2(a) shows a *hidden Markov phylogeny*, an HMM in which the observations are sets of nucleotides related by a phylogenetic tree [74, 86, 99]. This model has proven useful for gene-finding in the human genome based on data for multiple primate species [74]. The graphical model shown in Figure 2(b) is a *coupled HMM*, in which two chains of state variables are coupled via links between the chains; this model is appropriate for fusing pairs of data streams such as audio and lip-reading data in speech recognition [95]. Figure 2(c) shows a

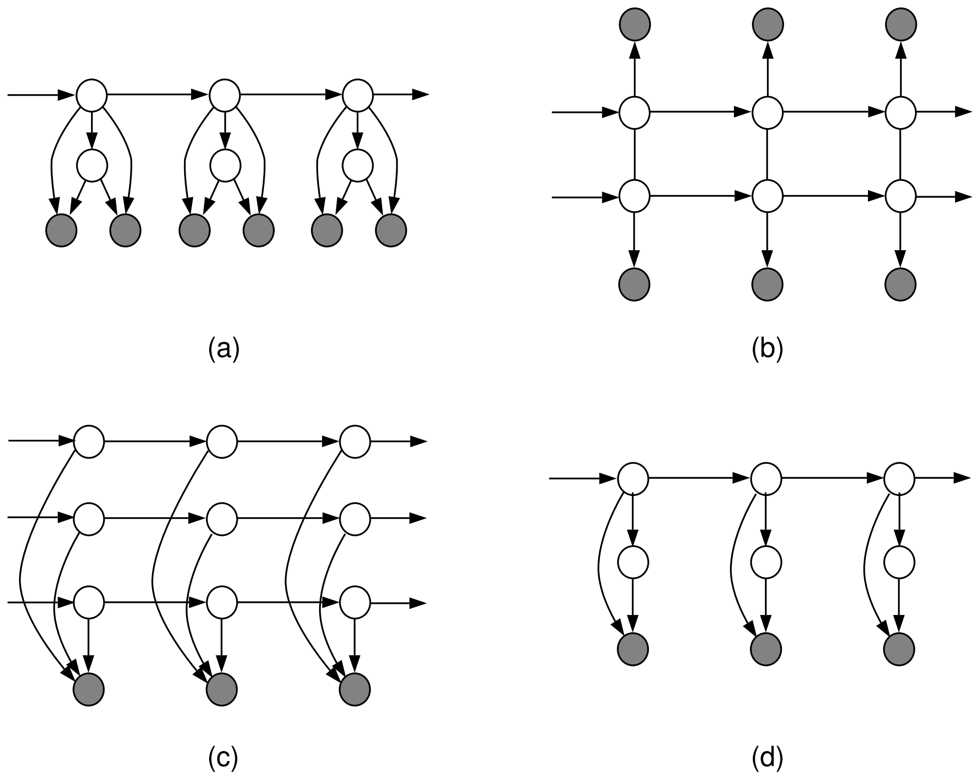


Figure 2. Variations on the HMM theme. (a) A phylogenetic HMM. (b) The coupled HHM. (c) The factorial HMM. (d) An HMM with mixture-model emissions.

factorial HMM, in which multiple chains are coupled by their links to a common set of observed variables [47]. 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 [103]. Finally, in Figure 2(d), we show a variation of the HMM in which the state-dependent observation distribution is a finite mixture model. This variant is widely used in speech recognition systems [55].

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 [12], 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 3, 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.[4]

2.4.3 Image processing

For several decades, undirected graphical models (also known as Markov random fields) have played an important role in image processing [e.g., 120, 51, 24, 10], as well as in spatial statistics more

[4]This model is discussed in more detail in Example 5 of Section 3.2.

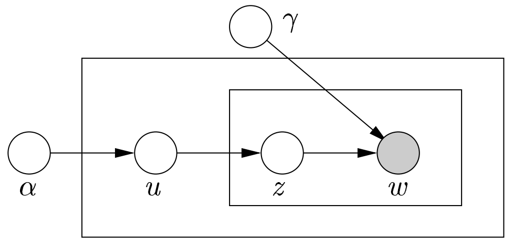
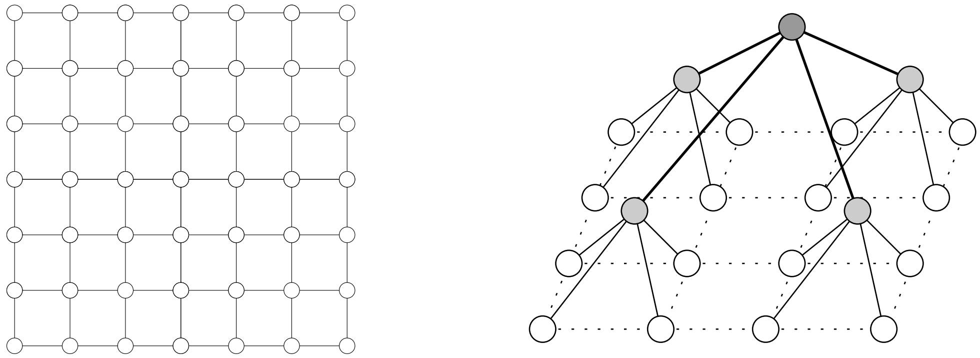


Figure 3. Graphical illustration of the latent Dirichlet allocation model. The variable u, which is distributed as Dirichlet with parameter α, specifies the parameter for the multinomial "topic" variable z. The "word" variable w is also multinomial conditioned on z, with γ specifying the word probabilities associated with each topic. The rectangles, known as *plates*, denote conditionally-independent replications of the random variables inside the rectangle.

generally [90]. The simplest use of a Markov random field model is in the pixel domain, where each pixel in the image is associated with a node 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 4(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-



(a) (b)

Figure 4. (a) The 4-nearest neighbor lattice model in 2D is often used for image modeling. (b) A multiscale quadtree approximation to a 2D lattice model. Nodes in the original lattice (drawn in white) lie at the 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.

scale lattice models is intractable, which necessitates the use of approximate methods. Markov chain Monte Carlo methods are often used [46], but they can be too slow and computationally intensive for many applications. More recently, the sum-product algorithm has become popular as an approximate inference method for image processing and computer vision problems [e.g., 41, 42].

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 4(b), can be used to approximate lattice models [119]. 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.

2.4.4 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., 44, 75], are based on graphical models. Figure 5 provides an illustration of a very

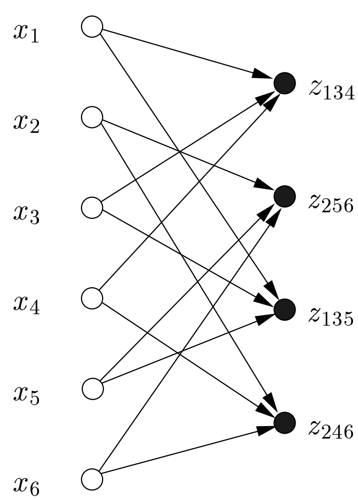


Figure 5. A directed graphical model representation of a parity check code of length $n = 6$. Open circles on the left represent bits in the code, whereas black circles on the right represent parity variables. This particular code is a $(2, 3)$ code, since each bit is connected to two parity variable, and each parity relation involves three bits.

small parity check code, represented here as a directed graphical model.[5] The six white nodes on the left represent the bits that comprise the codewords (i.e., binary sequences of length six); each of the four black nodes on the left corresponds to a binary variable z_{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, is captured in the graphical formalism by a conditional probability table of the form $p(z_{stu} \,|\, x_s, x_t, x_u)$. The directed edges incident on each parity variable z_{stu} correspond to the relevant bits; in the case shown here, the parity checks range over the set of triples $\{1, 3, 4\}$, $\{1, 3, 5\}$, $\{2, 4, 6\}$ and $\{2, 5, 6\}$. The code is defined by setting each parity variable z_{stu} to 0, which then forces the subset $\{x_s, x_t, x_u\}$ to have even parity.

The decoding problem entails estimating which codeword was transmitted on the basis of a vector $\mathbf{y}$ of noisy observations. 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 $p(x_s = 1|\mathbf{y})$ at each node, or computing the

[5]It is more standard in the coding community to represent a code via a factor graph [63], in which the black nodes are not random variables but instead represent functions or "factors" that enforce the parity check relations.

most likely codeword (i.e., the mode of the posterior). For the simple code of Figure 5, 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 n goes to infinity), where martingale arguments can be used to prove concentration results [89, 72]. 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.

To obtain the marginal probability of a single variable, $p(x_s)$, it suffices to choose a specific ordering of the remaining variables and to "eliminate" (sum or integrate) variables according to that order. Repeating this operation for each individual variable would yield the full set of marginals; this approach, however, 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 factors in (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. On the moral graph, the factors in (1) are all defined on cliques, and (1) is a special case of the undirected representation in (2). Throughout the rest of the paper, 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 [1, 29, 63, 57, 69]. 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:

$$p(\mathbf{x}) \quad = \quad \frac{1}{Z} \prod_{s \in V} \psi_s(x_s) \prod_{(s,t) \in E(T)} \psi_{st}(x_s, x_t). \tag{3}$$

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

$$\mu_s(x_s) \quad := \quad \sum_{\{\mathbf{x}' \mid x'_s = x_s\}} p(\mathbf{x}) \tag{4}$$

for every node of a tree-structured graph. We will focus on 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 non-serial dynamic programming (DP) that generalizes the usual serial form of deterministic dynamic programming [7] to arbitrary tree-structured graphs. The essential principle underlying 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 $\mathcal{N}(s) = \{u \in V | \ (s, u) \in E\}$. For each $u \in \mathcal{N}(s)$, 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 disjoint for $u \neq v$. In this way, each vertex $u \in \mathcal{N}(s)$ can be viewed as the root of a subtree T_u, as illustrated in Figure 6.

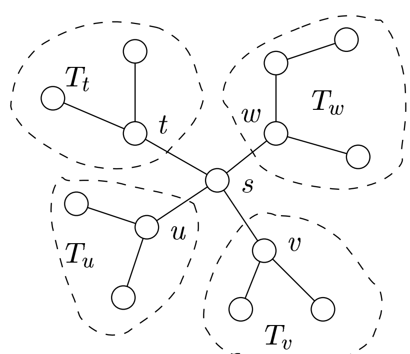


Figure 6. Decomposition of a tree, rooted at node s, into subtrees. Each neighbor (e.g., u) of node s is the root of a subtree (e.g., T_u). Subtrees T_u and T_v, for $t \neq u$, are disconnected when node s is removed from the graph.

For each subtree T_t, we define $x_{V_t} := \{x_u \mid u \in V_t\}$. Now consider the collection of terms in equation (3) associated with vertices or edges in T_t. We collect all of these terms into the following product:

$$p(x_{V_t}; T_t) \quad \propto \quad \prod_{u \, \in \, V_t} \psi_u(x_u) \prod_{(u,v) \in E_t} \psi_{uv}(x_u, x_v). \tag{5}$$

With this notation, the conditional independence properties of a tree allow the computation of the marginal at node μ_s to be broken down into a product of subproblems, one for each of the subtrees in the set $\{T_t, t \in \mathcal{N}(s)\}$, in the following way:

$$\mu_s(x_s) \quad = \quad \kappa \, \psi_s(x_s) \prod_{t \in \mathcal{N}(s)} M^*_{ts}(x_s) \tag{6a}$$

$$M^*_{ts}(x_s) \quad := \quad \sum_{\{x'_{T_t} \mid x'_s = x_s\}} \psi_{st}(x_s, x'_t) \;\; p(x'_{T_t}; T_t) \tag{6b}$$

In this 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 $u \in \mathcal{N}(t)$. 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 following recursion:

$$M_{ts}(x_s) \quad \leftarrow \quad \kappa \sum_{x'_t} \Big\{ \psi_{st}(x_s, x'_t) \, \psi_t(x'_t) \prod_{u \in \mathcal{N}(t)/s} M_{ut}(x'_t) \Big\}, \tag{7}$$

where $\kappa > 0$ is a normalization constant. It can be shown [85] that for tree-structured graphs, iterates generated by the update (7) 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 (6b), 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 (6a).

Max-product algorithm: Suppose that the summation in the update (7) is replaced by a maximization. The resulting *max-product* algorithm solves the problem of finding a mode of a tree-structured distribution $p(\mathbf{x})$. In this sense, it represents a generalization of the Viterbi algorithm [40] 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* $\nu_s(x_s) := \max_{\{\mathbf{x}' \mid x'_s = x_s\}} p(\mathbf{x}')$ at each node of the graph, via the analog of equation (5). Given these max-marginals, it is straightforward to compute a mode $\widehat{\mathbf{x}} \in \arg\max_{\mathbf{x}} p(\mathbf{x})$ of the distribution; see the papers [29, 110] for further details. More generally, updates of this form apply to arbitrary *commutative semirings* on tree-structured graphs [106, 97, 29, 1]. 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 cliques of G. Having done so, it is tempting to simply apply a standard algorithm for inference on trees. However, the clique tree must satisfy an additional restriction so as to ensure consistency 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 random variable x_s. It turns out that the following property is necessary and sufficient to enforce such consistency:

Definition 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 asserts that a graph G has a junction tree if and only if it is *triangulated.*[6] (See Lauritzen [68] for a proof.) This result underlies the *junction tree algorithm* [69] 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 1. To illustrate the junction tree construction, consider the 3×3 grid shown in Figure 7(a). The first step is to form a triangulated version $\widetilde{G}$, as shown in Figure 7(b). Note that

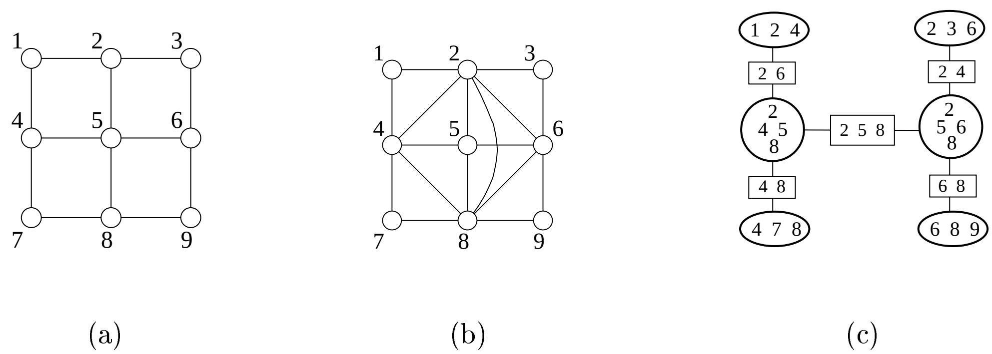


Figure 7. Illustration of junction tree construction. (a) Original graph is a 3×3 grid. (b) Triangulated version of original graph. Note the two 4-cliques in the middle. (c) Corresponding junction tree for triangulated graph in (b), with maximal cliques depicted within ellipses, and separator sets within rectangles.

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 7(c). These cliques only grow larger as the grid size is increased. $\diamondsuit$

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 [29] for an extensive discussion of the max-product version of the

[6] A graph is triangulated means that every cycle of length four or longer has a chord.

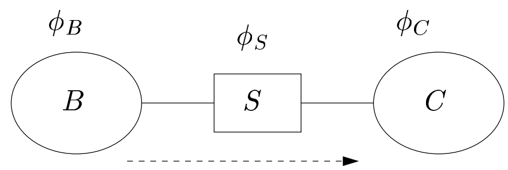


Figure 8: A message-passing operation between cliques B and C via the separator set S.

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 7). Let $\phi_C(x_C)$ denote a potential on a clique C, and let $\phi_S(x_S)$ 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 can be written in the following form:

$$\phi_S^*(x_S) = \sum_{x_{B \setminus C}} \phi_B(x_B) \tag{8a}$$

$$\phi_C^*(x_C) = \frac{\phi_S^*(x_S)}{\phi_S(x_S)} \phi_C(x_C), \tag{8b}$$

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 8). 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; that is, they agree with respect to the vertices 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 $\mu_C(x_C) \propto \phi_C(x_C)$ for all cliques C. This equivalence can be established by a suitable generalization of the proof of correctness of the sum-product algorithm presented previously (see also Lauritzen [68]). 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 $p(\cdot)$. 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). 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 $p(\cdot)$ factorizes in the form

$$p(\mathbf{x}) = \frac{\prod_{C \in \mathcal{C}} \mu_C(x_C)}{\prod_{S \in \mathcal{S}} [\mu_S(x_S)]^{d(S)-1}}, \tag{9}$$

where μ_C and μ_S are the marginal distributions over the cliques and separator sets respectively. Observe that unlike the representation of equation (2), the decomposition of equation (9) is directly in terms of marginal distributions, and does not require a normalization constant (i.e., $Z = 1$).

Example 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 equation (8), 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 $\tau_C(x_C)$ and $\tau_S(x_S)$ 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 $\{\tau_S, \tau_C\}$ are *locally consistent* in the following sense:

$$\sum_{x_S} \tau_S(x_S) = 1 \qquad \text{normalization} \tag{10a}$$

$$\sum_{\{\mathbf{x}'_C \,|\, \mathbf{x}'_S = x_S\}} \tau_C(\mathbf{x}'_C) = \tau_S(x_S) \qquad \text{marginalization} \tag{10b}$$

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 1. *A candidate set of local marginals $\{\tau_S, \tau_C\}$ on the separator sets and cliques of a junction tree is globally consistent if and only if it is locally consistent in the sense of equation* (10)*. Moreover, any such locally consistent quantities are the marginals of the probability distribution defined by equation* (9)*.*

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 (8) 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.[7] 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 [38], the forward-backward algorithms for hidden Markov models [88], and the Kalman filtering-smoothing algorithms for state-space models [58]. 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.

[7] To be more precise, the treewidth is one less than the size of this largest clique [see 13].

2.6 Message-passing algorithms for approximate inference

It is the goal of the remainder of the paper to develop a general theoretical framework for understanding and analyzing a class of techniques known as *variational inference algorithms.* 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 intuition or on the basis of analogies to exact or Monte Carlo algorithms. In this section, we give a high-level description of two variational inference algorithms, with the goal of highlighting their simple and intuitive nature.

The first variational algorithm that we consider is a so-called "loopy" form of the sum-product algorithm (also referred to as the *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 (7) 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 is used for computing approximate modes in graphical models with cycles.

A second variational algorithm is the so-called *naive mean field* algorithm. For concreteness, here we describe this algorithm in application to the Ising model of statistical physics. The Ising model is a Markov random field involving a binary random vector $\mathbf{x} \in \{0,1\}^n$, 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 of Section 3.2 for a more detailed description of this model.) Consider now the Gibbs sampler for such a 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 $\mathcal{N}(s)$ the neighbors of a node $s \in V$, and letting $x_{\mathcal{N}(s)}^{(p)}$ denote the state of the neighbors of s at iteration p, the Gibbs update for x_s takes the following form:

$$x_s^{(p+1)} = \begin{cases} 1 & \text{if } u \leq \{1 + \exp[-(\theta_s + \sum_{t \in \mathcal{N}(s)} \theta_{st} x_t^{(p)})]\}^{-1} \\ 0 & \text{otherwise} \end{cases}, \tag{11}$$

where u is a sample from a uniform distribution $\mathcal{U}(0,1)$.

In a dense graph, such that the cardinality of $\mathcal{N}(s)$ is large, we might attempt to invoke a law of large numbers or some other concentration result for $\sum_{t \in \mathcal{N}(s)} \theta_{st} x_t^{(p)}$. 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 $p(x_s = 1)$ at each vertex $s \in V$, we might consider the following averaged version of equation (11):

$$\mu_s \quad \leftarrow \quad \Big\{1 + \exp\big[-(\theta_s + \sum_{t \in \mathcal{N}(s)} \theta_{st} \mu_t)\big]\Big\}^{-1}. \tag{12}$$

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 (12) 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 (12) 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 remainder of this paper 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 are the following section, provide the appropriate framework in which to develop these variational principles in an unified manner.

3 Exponential families and convex analysis

In this section, we introduce exponential families of distributions, focusing on the links with convex analysis and specifically with the theory of conjugate duality. Further details on exponential families and their properties can be found in various sources [4, 5, 17, 35]. For further background on convex analysis, we refer the reader to [14, 53, 92].

3.1 Basics of exponential families

For the sake of readability, we begin by restating our basic notation for random vectors and sample spaces, originally given in Section 2.1. For each $s = 1, \dots, n$, let x_s be a random variable taking values in some sample space $\mathcal{X}_s$, which may be continuous (e.g., $\mathcal{X}_s = \mathbb{R}$), or a discrete alphabet (e.g., $\mathcal{X}_s = \{0, 1, \dots, m-1\}$). The random vector $\mathbf{x} = \{x_s \mid s = 1, \dots n\}$ then takes values in the Cartesian product space $\mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n$, which we denote by $\mathcal{X}^n$. Given some arbitrary function[8] $h : \mathcal{X}^n \to \mathbb{R}_+$, we endow $\mathcal{X}^n$ with the measure $\boldsymbol{\nu}$ defined via $d\boldsymbol{\nu} = h(\mathbf{x})\, d\mathbf{x}$, where component dx_s in the product $d\mathbf{x} = \prod_{s=1}^n dx_s$ is usually (a suitably restricted version of) Lebesgue measure when $\mathcal{X}_s$ is a continuous space, or the counting measure when $\mathcal{X}_s$ is discrete.

An exponential family consists of a particular class of densities taken with respect to the dominating measure $\boldsymbol{\nu}$. Let $\boldsymbol{\phi} = \{\phi_\alpha \mid \alpha \in \mathcal{I}\}$ be a collection of Borel measurable functions $\phi_\alpha : \mathcal{X}^n \to \mathbb{R}$. These functions are known either as *potentials* or *sufficient statistics*. Here $\mathcal{I}$ is an index set with $d = |\mathcal{I}|$ elements to be specified, so that $\boldsymbol{\phi}$ itself can be viewed as a vector-valued mapping from $\mathcal{X}^n$ to $\mathbb{R}^d$. Associated with $\boldsymbol{\phi}$ is a vector $\theta = \{\, \theta_\alpha \mid \alpha \in \mathcal{I} \,\}$ of *exponential* or *canonical* parameters. For each fixed $\mathbf{x} \in \mathcal{X}^n$, we use $\langle \theta, \, \boldsymbol{\phi}(\mathbf{x}) \rangle$ to denote the (Euclidean) inner product in $\mathbb{R}^d$ of the two vectors θ and $\boldsymbol{\phi}(\mathbf{x})$. With this notation, the *exponential family* associated with $\boldsymbol{\phi}$ consists of the following parameterized collection of density functions (taken with respect to $d\boldsymbol{\nu}$):

$$p(\mathbf{x}; \theta) \quad = \quad \exp\big\{ \langle \theta, \, \boldsymbol{\phi}(\mathbf{x}) \rangle \; - \; A(\theta) \big\}. \tag{13}$$

The quantity A, known as the *log partition function*, is defined by the integral:

$$A(\theta) \quad = \quad \log \int_{\mathcal{X}^n} \exp\langle \theta, \, \boldsymbol{\phi}(\mathbf{x}) \rangle \, \boldsymbol{\nu}(d\mathbf{x}). \tag{14}$$

Presuming that the integral is finite, this definition ensures that $p(\mathbf{x}; \theta)$ is properly normalized (i.e., $\int_{\mathcal{X}^n} p(\mathbf{x}; \theta) \boldsymbol{\nu}(d\mathbf{x}) = 1$).

[8] Here $\mathbb{R}_+ = \{ y \in \mathbb{R} \mid y \geq 0 \}$.

With the set of potentials $\boldsymbol{\phi}$ is fixed, each parameter vector θ indexes a particular member $p(\mathbf{x};\theta)$ of the family. The exponential parameters θ of interest belong to the set

$$\Theta \;:=\; \{\theta \in \mathbb{R}^d \mid A(\theta) < \infty\}. \tag{15}$$

We will see shortly that A is a convex function of θ, which in turn implies that Θ must be a convex set. The log partition function A plays a prominent role in this paper.
The following notions will be important in subsequent development:

Regular families: An exponential family for which the domain Θ of equation (15) is an open set is known as a *regular* family. Although there do exist exponential families for which Θ is closed [see, e.g., 17], herein we restrict our attention to regular exponential families.

Minimal: It is typical to define an exponential family with a collection of functions $\boldsymbol{\phi} = \{\phi_\alpha\}$ for which there is no linear combination $\langle a,\, \boldsymbol{\phi}(\mathbf{x})\rangle = \sum_{\alpha \in \mathcal{I}} a_\alpha \phi_\alpha(\mathbf{x})$ equal to a constant ($\boldsymbol{\nu}$-a.e). 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 an *overcomplete representation*, which is non-minimal (so that some linear combination of $\boldsymbol{\phi}$ is equal to a constant $\boldsymbol{\nu}$-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, it seems highly undesirable in a statistical setting because identifiability of the parameter vector θ is lost. However, this notion of overcompleteness will play a key role in our later analysis of the sum-product algorithm and its generalizations (Sections 6 and 7).

Table 1 provides some examples of well-known scalar exponential families. Observe that all of these families are both regular (since Θ is open), and minimal (since the collection of sufficient statistics $\boldsymbol{\phi}$ do not satisfy any linear relations).

Family	$\mathcal{X}$	$\boldsymbol{\nu}$	$\log p(\mathbf{x};\theta)$	$A(\theta)$	Θ
Bernoulli	$\{0,1\}$	Counting	$\theta x - A(\theta)$	$\log[1+\exp(\theta)]$	$\mathbb{R}$
Gaussian	$\mathbb{R}$	Lebesgue	$\theta_1 x + \theta_2 x^2 - A(\theta)$	$\frac{1}{2}[\theta_1 + \log \frac{2\pi e}{-\theta_2}]$	$\{\theta \in \mathbb{R}^2 \mid \theta_2 < 0\}$
Exponential	$(0,+\infty)$	Lebesgue	$\theta\,(-x) - A(\theta)$	$-\log\theta$	$(0,+\infty)$
Poisson	$\{0,1,2\ldots\}$	Counting $h(x)=1/x!$	$\theta x - A(\theta)$	$\exp(\theta)$	$\mathbb{R}$
Beta	$(0,1)$	Lebesgue	$\theta_1 \log x + \theta_2 \log(1-x)$ $-A(\theta)$	$\sum_{i=1}^{2} \log\Gamma(\theta_i+1)$ $-\log\Gamma(\sum_{i=1}^{2}(\theta_i+1))$	$(-1,+\infty)^2$

Table 1. Several well-known classes of scalar random variables as exponential families. In all cases, the base measure $\boldsymbol{\nu}$ is either Lebesgue or counting measure, suitably restricted to the sample space $\mathcal{X}$. All of these examples are both minimal and regular.

3.2 Graphical models and exponential families

The scalar examples in Table 1 serve as building blocks for the construction of more complex exponential families for which graphical structure does play a role. Earlier, we described graphical models in terms of products of functions, as in equations (1) and (2). In the context of exponential families, these products become additive decompositions within the exponent.

Example 3 (Ising model). We begin with the *Ising model* from statistical physics [6, 20, 83], which is a particular kind of Markov random field. Consider a graph $G = (V, E)$ and suppose that the random variable x_s associated with node $s \in V$ is Bernoulli. Components x_s and x_t of the full random vector $\mathbf{x}$ 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 of the form

$$p(\mathbf{x}; \theta) = \exp\big\{ \sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in E} \theta_{st} x_s x_t - A(\theta) \big\}, \tag{16}$$

taken with respect to counting measure restricted to $\{0, 1\}^n$. Here θ_{st} is the strength of edge (s, t), and θ_s is the node parameter for node s. (Strictly speaking, this model is more general than the classical Ising model, in which θ_{st} is constant for all edges.) The index set $\mathcal{I}$ consists of the union $V \cup E$, and the dimension of the family is $d = n + |E|$. The domain Θ is the full space $\mathbb{R}^d$, since the sum that defines the log partition function $A(\theta)$ is finite for all $\theta \in \mathbb{R}^d$. Hence, the family is regular. Moreover, it is a minimal representation, since there is no linear combination of the potentials equal to a constant $\boldsymbol{\nu}$-a.e. $\diamondsuit$

The standard Ising model can be generalized in a number of different ways. Although equation (16) 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 exponential parameter θ_{stu}, to equation (16). More generally, to incorporate coupling in k-cliques, we can add monomials up to order k. At the upper extreme, taking $k = n$ amounts to connecting all nodes in the graphical model, which allows one to represent any distribution over a binary random vector $\mathbf{x} \in \{0, 1\}^n$. It is also straightforward to extend these models to the multinomial case, in which each x_s takes values in the space $\mathcal{X}_s = \{0, 1, \ldots, m-1\}$.

We now turn to another important class of graphical models:

Example 4 (Gaussian MRF). A Gaussian Markov random field [e.g., 100] consists of a multivariate Gaussian random vector that respects the Markov properties of a graph $G = (V, E)$. It can be represented in exponential form using the potentials $\{x_s, x_s^2 \mid s \in V\} \cup \{x_s x_t \mid (s, t) \in E\}$, with associated parameters $\{\theta_s, s \in V\} \cup \{\theta_{st} \mid (s, t) \in E\}$. Note that there are a total of $d = 2n + |E|$ potential functions. It is convenient to represent the potentials and parameters compactly as $(n+1) \times (n+1)$ symmetric matrices:

$$\mathbf{X} := \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{x} \end{bmatrix}, \qquad U(\theta) := \begin{bmatrix} 0 & \theta_1 & \theta_2 & \ldots & \theta_n \\ \theta_1 & \theta_{11} & \theta_{12} & \ldots & \theta_{1n} \\ \theta_2 & \theta_{21} & \theta_{22} & \ldots & \theta_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \theta_n & \theta_{n1} & \theta_{n2} & \ldots & \theta_{nn} \end{bmatrix} = \begin{bmatrix} 0 & z^T(\theta) \\ z(\theta) & Z(\theta) \end{bmatrix} \tag{17}$$

Here $z(\theta)$ denotes the n-vector $[\theta_1 \ \ldots \ \theta_n]^T$, while $Z(\theta)$ denotes the $n \times n$ matrix of the $\{\theta_{st}\}$. It should be understood that $\theta_{st} = 0$ whenever $(s, t) \notin E$, which reflects the Markov structure of the underlying graph. For any two symmetric matrices C and D, we let $\langle\!\langle C,\ D\rangle\!\rangle$ denote the inner product defined by $\text{trace}(C\,D)$. Using this notation, a Gaussian MRF can be represented as an exponential family of the form

$$p(\mathbf{x};\theta) \quad = \quad \exp\big\{\langle\!\langle U(\theta),\ \mathbf{X}\rangle\!\rangle - A(\theta)\big\}. \tag{18}$$

The integral defining $A(\theta)$ is finite only if $Z(\theta) \prec 0$, so that $\Theta = \{\theta \in \mathbb{R}^d \mid Z(\theta) \prec 0\}$. ♢

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 in the following example.

Example 5 (LDA). The *Latent Dirichlet allocation* model [12], illustrated earlier as a graphical model in Figure 3, involves three different types of random variables: "words" w, "documents" z, and Dirichlet variables u. The vector of exponential parameters θ can be partitioned as $\theta = (\alpha, \gamma)$. The quantity α is an exponential parameter for the Dirichlet variable u, which has a density with respect to Lebesgue measure of the form $p(u;\alpha) \propto \exp\{\sum_{i=1}^n \alpha_i \log u_i\}$. The Dirichlet variable u, in turn, serves as the parameter for the multinomial variable $z \in \{1, 2, \ldots, k\}$, so that $p(z;u) = \exp\big\{\sum_{i=1}^k \mathbb{I}_i[z] \log u_i\big\}$, where $\mathbb{I}_i[z]$ is the indicator for the event $\{z = i\}$. Finally, the conditional distribution of w given z is parameterized by γ as follows:

$$p(w = j \,|\, z = i, \gamma) \quad = \quad \exp(\gamma_{ij}), \qquad \forall\ i = 1, \ldots, k, \quad j = 1, \ldots, l.$$

This set of equations can be written more compactly as $p(w \,|\, z, \gamma) = \exp\big\{\sum_{i=1}^k \sum_{j=1}^l \gamma_{ij} \mathbb{I}_i[z] \mathbb{I}_j[w]\big\}$.

Overall then, for a single triplet $\mathbf{x} := (u, z, w)$, the LDA model is an exponential family with parameter vector $\theta := (\alpha, \gamma)$, with a density of the following form:

$$p(u;\alpha)p(z;u)p(w \,|\, z, \gamma) \ \propto\ \exp\Big\{\sum_{i=1}^n \alpha_i \log u_i + \sum_{i=1}^k \mathbb{I}_i[z] \log u_i + \sum_{i=1}^k \sum_{j=1}^l \gamma_{ij} \mathbb{I}_i[z] \mathbb{I}_j[w]\Big\}. \tag{19}$$

The sufficient statistics $\boldsymbol{\phi}$ consist of the collections $\{\log u_i, i = 1, \ldots, k\}$, $\{\mathbb{I}_i[z] \log u_i,\ i = 1, \ldots k\}$, and $\{\mathbb{I}_i[z]\mathbb{I}_j[w],\ i = 1, \ldots, k, j = 1, \ldots l\}$. As illustrated in Figure 3, the full LDA model entails replicating these types of local structures many times. ♢

3.3 Properties of A

In this section, we first develop some basic properties of the log partition function, which we then build on by drawing connections to convex analysis. Of particular importance is the idea that the expectations of $\boldsymbol{\phi}(\mathbf{x})$ under $p(\mathbf{x};\theta)$ define an alternative parameterization of the exponential family, known as the *mean parameterization.* For the sake of readability, the proofs of the majority of those results given in the remainder of Section 3 as well as in Section 4 have been deferred to Appendix A.

3.3.1 Derivatives and convexity

We begin by establishing that the log partition function is both smooth and convex in terms of θ.

Proposition 2. *The log partition function is lower semi-continuous on $\mathbb{R}^d$, and C^∞ on Θ. Its derivatives are the cumulants of the random vector $\boldsymbol{\phi}(\mathbf{x})$—in particular:*

$$\frac{\partial A}{\partial \theta_\alpha}(\theta) \;=\; \mathbb{E}_\theta[\phi_\alpha(\mathbf{x})] \;:=\; \int \phi_\alpha(\mathbf{x})\; p(\mathbf{x};\theta)\boldsymbol{\nu}(d\mathbf{x}). \tag{20a}$$

$$\frac{\partial^2 A}{\partial \theta_\alpha\,\partial \theta_\beta}(\theta) \;=\; \mathbb{E}_\theta[\phi_\alpha(\mathbf{x})\phi_\beta(\mathbf{x})] - \mathbb{E}_\theta[\phi_\alpha(\mathbf{x})]\mathbb{E}_\theta[\phi_\beta(\mathbf{x})]. \tag{20b}$$

Moreover, $\|\nabla A(\theta^t)\| \to +\infty$ for any sequence $\{\theta^t\} \subset \Theta$ approaching the boundary.

The conditions in Proposition 2 ensure that A is *essentially smooth* [92], also referred to as *steep* in statistical settings [17]. This property plays an important role in subsequent development. Moreover, Proposition 2 identifies A as the *cumulant generating function* of the random vector $\boldsymbol{\phi}(\mathbf{x})$. In particular, equation (20b) shows that the Hessian $\nabla^2 A(\theta)$ can be interpreted as a particular type of Gram matrix, which leads to the following:

Corollary 1. *The log partition function A is a convex function of θ, and strictly so if the representation is minimal.*

3.3.2 Mapping to mean parameters

Given a potential function vector $\boldsymbol{\phi} : \mathcal{X}^n \to \mathbb{R}^d$, it is of interest to consider the set of vectors $\mu \in \mathbb{R}^d$ that are formed by taking expectations of $\boldsymbol{\phi}$ under an arbitrary distribution that is absolutely continuous with respect to $\boldsymbol{\nu}$. Accordingly, we define the following set:

$$\mathcal{M} \;:=\; \Big\{ \mu \in \mathbb{R}^d \;\big|\; \exists\, p(\cdot) \;\text{ s. t. }\; \int \boldsymbol{\phi}(\mathbf{x})\; p(\mathbf{x})\boldsymbol{\nu}(d\mathbf{x}) = \mu \Big\}. \tag{21}$$

Note that $\mathcal{M}$ is a convex set.

Given an arbitrary member of the exponential family defined by $\boldsymbol{\phi}$, we can define a mapping $\Lambda : \Theta \to \mathcal{M}$ as follows:

$$\Lambda(\theta) \;:=\; \mathbb{E}_\theta[\boldsymbol{\phi}(\mathbf{x})] \;=\; \int_{\mathcal{X}^n} \boldsymbol{\phi}(\mathbf{x})\; p(\mathbf{x};\theta)\;\boldsymbol{\nu}(d\mathbf{x}). \tag{22}$$

Note that Λ is a particular case of a gradient mapping, since $\Lambda(\theta) = \nabla A(\theta)$ by equation (20a). The mapping Λ associates to each $\theta \in \Theta$ a vector of *mean parameters* $\mu := \Lambda(\theta)$ belonging to the set $\mathcal{M}$. The goal of this section is to obtain a precise characterization of the nature of this correspondence between θ and μ. Of particular interest are the following two issues:

1. determining when Λ to one-to-one and hence invertible on its image, and

2. characterizing the image of Θ under the mapping Λ.

The answer to the first question turns out to be straightforward, hinging essentially on the minimality of the representation. Although the answer to the second question is also straightforward—namely, Λ is onto the (relative) interior of $\mathcal{M}$—the proof is more involved. To be clear, this question is not trivial, because the definition (21) allows the density $p(\cdot)$ to be arbitrary, whereas the mapping Λ uses only members of the exponential family. We begin with a result addressing the first question:

Proposition 3. *The mapping Λ is one-to-one if and only if the exponential representation is minimal.*

Proposition 3 asserts that the mean parameter mapping is not invertible for an overcomplete representation. More specifically, the inverse image $\Lambda^{-1}(\mu) := \{\theta \in \Theta \mid \Lambda(\theta) = \mu\}$—rather than being a singleton (as it would be for an invertible mapping)—is a (non-trivial) affine subset of Θ.

Example 6. To illustrate, consider a Bernoulli random variable $x \in \{0, 1\}$. Suppose that we use the overcomplete exponential representation $p(x; \theta) \propto \exp\{\theta_0(1 - x) + \theta_1 x\}$, so that $\Theta = \mathbb{R}^2$. In this case, the mean parameters (μ_0, μ_1) are simply marginal probabilities—viz. $\mu_i = p(x = i)$ for $i = 0, 1$. The set $\mathcal{M}$ of realizable mean parameters is the simplex $\{\mu \geq 0 \mid \mu_0 + \mu_1 = 1\}$. For a fixed mean parameter $\mu > 0$ in the simplex, it is easy to show that the inverse image consists of the affine set $\Lambda^{-1}(\mu) := \{(\theta_0, \theta_1) \in \mathbb{R}^2 \mid \theta_1 - \theta_0 = \log \frac{\mu_1}{\mu_0}\}$. $\diamondsuit$

In general, although there is no longer a bijection between Θ and $\Lambda(\Theta)$ in an overcomplete representation, there is still a bijection between each element of $\Lambda(\Theta)$ and an affine subset of Θ. For either a minimal or an overcomplete representation, we say that a pair (θ, μ) is *dually coupled* if $\mu = \Lambda(\theta)$, and hence $\theta \in \Lambda^{-1}(\mu)$. This notion of dual coupling plays an important role in the sequel.

We now turn to the second question regarding the range of Λ.

Theorem 1. *The mean parameter mapping Λ is onto the (relative) interior of $\mathcal{M}$ (i.e., $\Lambda(\Theta) = \operatorname{ri}\mathcal{M}$).*

Remarks: The relative interior of a convex set is the interior taken with respect to its affine hull. A key fact is that any non-empty convex set is guaranteed to have a non-empty relative interior. See Appendix B for more details.

Typically, the exponential family $\{p(\mathbf{x}; \theta) \mid \theta \in \Theta\}$ describes only a strict subset of all possible densities, whereas the definition (21) of $\mathcal{M}$ allows the density $p(\cdot)$ to be arbitrary. The significance of Theorem 1, then, lies in the fact that for any mean parameter $\mu \in \operatorname{ri}\mathcal{M}$, it suffices to restrict the expectations in definition (21) to members of the exponential family. Moreover, for a minimal exponential family, Proposition 3 guarantees that there is a *unique* exponential parameter $\theta(\mu)$ such that $\Lambda(\theta(\mu)) = \mu$. However, if the exponential family describes a strict subset of all densities, then there exists at least some other density $p(\cdot)$—albeit not a member of the exponential family—that also realizes μ (i.e., for which $\int \boldsymbol{\phi}(\mathbf{x}) p(\mathbf{x}) \boldsymbol{\nu}(d\mathbf{x}) = \mu$). As discussed in the following section, the distinguishing property of $p(\mathbf{x}; \theta(\mu))$ lies in the notion of maximum entropy.

3.3.3 Fenchel-Legendre conjugate

We now turn to consideration of the Fenchel-Legendre conjugate of the log partition function A. In particular, this conjugate dual function, which we denote by A^*, is defined as follows:

$$A^*(\mu) \quad := \quad \sup_{\theta \in \Theta} \{\langle \mu, \, \theta \rangle - A(\theta)\}. \tag{23}$$

Here $\mu \in \mathbb{R}^d$ is a 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.

The (Boltzmann-Shannon) entropy of the density $p(\mathbf{x}; \theta)$ with respect to $\boldsymbol{\nu}$ is defined as follows:

$$H(p(\mathbf{x}; \theta)) \quad = \quad -\int_{\mathcal{X}^n} p(\mathbf{x}; \theta) \log\left[p(\mathbf{x}; \theta)\right] \boldsymbol{\nu}(d\mathbf{x}) \quad = \quad -\mathbb{E}_\theta[\log p(\mathbf{x}; \theta)]. \tag{24}$$

The main result of Theorem 2 is that when $\mu \in \operatorname{ri}\mathcal{M}$, then the value of the dual function $A^*(\mu)$ is precisely the negative entropy of $p(\mathbf{x}; \theta(\mu))$, where $\theta(\mu)$ is an element of the inverse image $\Lambda^{-1}(\mu)$.

Of course, it is also important to consider $\mu \notin \operatorname{ri} \mathcal{M}$, in which case $\Lambda^{-1}(\mu)$ is empty. In this case, the behavior of the supremum defining $A^*(\mu)$ requires a more delicate analysis. As we show in the following theorem, it turns out that when $\mu \notin \operatorname{cl} \mathcal{M}$, then $A^*(\mu) = +\infty$.

More formally, we state the following:

Theorem 2.
(a) For any $\mu \in \operatorname{ri} \mathcal{M}$, let $\theta(\mu)$ denote a member of $\Lambda^{-1}(\mu)$. The Fenchel-Legendre dual of A has the following form:

$$A^*(\mu) \quad = \quad \begin{cases} -H(p(\mathbf{x}; \theta(\mu))) & \text{if } \mu \in \operatorname{ri} \mathcal{M} \\ +\infty & \text{if } \mu \notin \operatorname{cl} \mathcal{M}. \end{cases} \tag{25}$$

For any boundary point $\mu \in \operatorname{bd} \mathcal{M} := \operatorname{cl} \mathcal{M} \backslash \operatorname{ri} \mathcal{M}$, we have $A^*(\mu) = \lim_{n \to +\infty} [-H(p(\mathbf{x}; \theta(\mu^n)))]$, taken over a sequence $\{\mu^n\} \subset \operatorname{ri} \mathcal{M}$ converging to μ.

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

$$A(\theta) \quad = \quad \sup_{\mu \in \mathcal{M}} \big\{ \langle \theta,\, \mu \rangle - A^*(\mu) \big\}. \tag{26}$$

The fact that $A^*(\mu) = +\infty$ for $\mu \notin \operatorname{cl} \mathcal{M}$ is essential for our approach to variational inference. In particular, it implies that the variational representation of the log partition function reduces to an optimization over $\mathcal{M}$, as we see in equation (26). Consequently, $\mathcal{M}$ is the domain over which our key optimization problem takes place, and we will be interested in various approximations of this set.

Table 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 $\Theta \equiv \operatorname{dom} A$, as well as the set $\mathcal{M}$, which contains the effective domain of A^* (by Theorem 2(a)). On the basis of these examples, it

Family	Θ	$A(\theta)$	$\mathcal{M}$	$A^*(\mu)$
Bernoulli	$\mathbb{R}$	$\log[1 + \exp(\theta)]$	$[0, 1]$	$\mu \log \mu + (1 - \mu) \log(1 - \mu)$
Gaussian	$\{(\theta_1, \theta_2) \mid \theta_2 < 0\}$	$\frac{1}{2}[\theta_1 + \log \frac{2\pi e}{-\theta_2}]$	$\{(\mu_1, \mu_2) \mid \mu_2 - (\mu_1)^2 > 0\}$	$-\frac{1}{2} \log[2\pi e(\mu_2 - \mu_1^2)]$
Exponential	$(0, +\infty)$	$-\log \theta$	$(-\infty, 0)$	$-1 - \log(-\mu)$
Poisson	$\mathbb{R}$	$\exp(\theta)$	$(0, +\infty)$	$\mu \log \mu - \mu$

Table 2. Conjugate dual relations of Theorem 2 for several well-known exponential families of scalar variables.

can be seen that the specific behavior of A^* on the boundary $\operatorname{bd} \mathcal{M} := \operatorname{cl} \mathcal{M} \backslash \operatorname{ri} \mathcal{M}$ varies depending on the exponential family. For example, for the Bernoulli family, the boundary of $\mathcal{M} = [0, 1]$ consists of the points 0 and 1. As μ approaches either of these points, the dual function $A^*(\mu) = \mu \log \mu + (1 - \mu) \log(1 - \mu)$ tends to zero. This limiting behavior corresponds to the fact that the underlying distribution $p(\mathbf{x}; \theta(\mu))$ is tending to a delta function, which has a discrete entropy of zero. Therefore, we conclude that $\operatorname{dom} A^* = [0, 1] \equiv \mathcal{M}$ in the Bernoulli case. This type of reasoning can be generalized to the multinomial case.

On the other hand, in the scalar Gaussian case, the set $\mathcal{M}$ is defined by the quadratic constraint $\mu_2 - (\mu_1)^2 > 0$, corresponding to the fact that the variance of a (non-degenerate) Gaussian must be (strictly) positive. Note that $(0,0)$ is a boundary point of $\mathcal{M}$. We can compute the value of $A^*(0,0)$ by taking the limit $A^*(\mu^n)$ for a sequence $\{\mu^n\} \to (0,0)$ contained within $\operatorname{int}\mathcal{M} \equiv \mathcal{M}$. Considering, in particular, the sequence $\mu^n = (0, 1/n)$ and using the form of A^* given in Table 2, we obtain $\lim_{n\to+\infty} A^*(\mu^n) = +\infty$. This result is consistent with the limiting behavior of (differential) entropy for densities with a delta component.

3.3.4 Kullback-Leibler divergence

The conjugate duality between A and A^*, as characterized in Theorem 2, leads to several alternative forms of the Kullback-Leibler (KL) divergence for exponential family members, which we summarize here for the sake of subsequent developments. The standard definition [23] of the KL divergence between two distributions with densities q and p with respect to $\boldsymbol{\nu}$ is as follows:

$$D(q \,\|\, p) \quad := \quad \int_{\mathcal{X}^n} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})}\, \boldsymbol{\nu}(d\mathbf{x}). \tag{27}$$

The key result that underlies alternative representations for exponential families is *Fenchel's inequality* which, as applied to (A, A^*), asserts that for *any* pair $(\theta, \mu) \in \mathbb{R}^d \times \mathbb{R}^d$:

$$A(\theta) + A^*(\mu) \quad \geq \quad \langle \mu, \theta \rangle. \tag{28}$$

Moreover, equality holds in this equation if and only if θ and μ are dually coupled, meaning that $\mu = \Lambda(\theta)$ and $\theta \in \Lambda^{-1}(\mu)$.

Consider two exponential parameter vectors $\theta^1, \theta^2 \in \Theta$; with a slight abuse of notation, we use $D(\theta^1 \,\|\, \theta^2)$ to refer to the KL divergence between $p(\mathbf{x};\theta^1)$ and $p(\mathbf{x};\theta^2)$. We use μ^1 and μ^2 to denote the respective mean parameters (i.e., $\mu^i = \Lambda(\theta^i)$ for $i = 1,2$). A first alternative form of the KL divergence is obtained by substituting the exponential representations of $p(\mathbf{x};\theta^i)$ into equation (27) and then expanding and simplifying as follows:

$$D(\theta^1 \,\|\, \theta^2) \quad = \quad \mathbb{E}_{\theta^1}\left[\log \frac{p(\mathbf{x};\theta^1)}{p(\mathbf{x};\theta^2)}\right] \;=\; A(\theta^2) - A(\theta^1) - \langle \mu^1,\, \theta^2 - \theta^1 \rangle. \tag{29}$$

We refer to this representation as the *primal form* of the KL divergence. As illustrated in Figure 9,

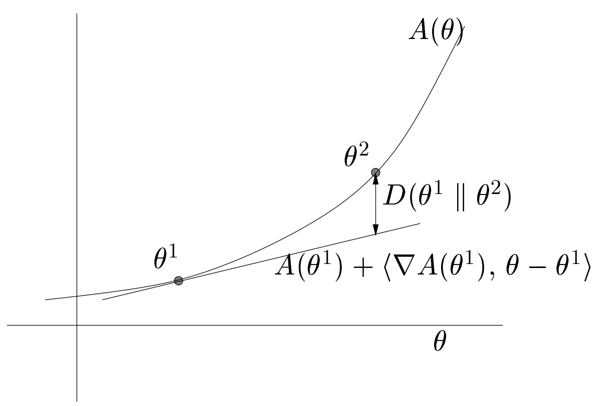


Figure 9. The hyperplane $A(\theta^1) + \langle \nabla A(\theta^1),\, \theta - \theta^1 \rangle$ supports the epigraph of A at θ^1. The Kullback-Leibler divergence $D(\theta \,\|\, \theta^2)$ is equal to the difference between $A(\theta^2)$ and this hyperplane.

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 [16, 19].

A second form of the KL divergence can be obtained by using the fact that Fenchel's inequality (28) holds with equality for the dually coupled pair (θ^1, μ^1). In this way, we can transform equation (29) into the following *mixed form* of the KL divergence:

$$D(\theta^1 \parallel \theta^2) \equiv D(\mu^1 \parallel \theta^2) = A(\theta^2) + A^*(\mu^1) - \langle \mu^1, \theta^2 \rangle. \tag{30}$$

Note that this mixed form of the divergence corresponds to the slack in Fenchel's inequality (28). It also provides an alternative view of the variational representation given in Theorem 2(b). In particular, equation (26) can be rewritten as follows:

$$\inf_{\mu \in \mathcal{M}} \big\{ A(\theta) + A^*(\mu) - \langle \theta, \mu \rangle \big\} = 0$$

Using equation (30), the variational representation in Theorem 2(b) is seen to be equivalent to the assertion that $\inf_{\mu \in \mathcal{M}} D(\mu \parallel \theta) = 0$.

Finally, by applying equation (28) as an equality once again, this time for the coupled pair (θ^2, μ^2), the mixed form (30) can be transformed into a purely *dual form* of the KL divergence:

$$D(\theta^1 \parallel \theta^2) \equiv D(\mu^1 \parallel \mu^2) = A^*(\mu^1) - A^*(\mu^2) - \langle \theta^2, \mu^1 - \mu^2 \rangle. \tag{31}$$

Note the symmetry between representations (29) and (31). In particular, to move from one to the other, we simply exchange the log partition function A for the negative entropy A^*, and we interchange the roles of θ^1 and θ^2 (as well as μ^1 and μ^2). This form of the KL divergence has an interpretation analogous to that of Figure 9, but with A replaced by the dual A^*.

4 Variational methods for computing mean parameters

For the next several sections, we focus on the first two inference problems described in Section 2.3. Restated in the language of exponential families, these problems correspond to computing the log partition function $A(\theta)$, and the mean parameters $\mu = \mathbb{E}_\theta[\boldsymbol{\phi}(\mathbf{x})]$ for a given distribution $p(\mathbf{x}; \theta)$. The current section is devoted to consideration of the ingredients in the variational approach.

Of central importance to the computation of the log partition function and the mean parameters is Theorem 2(b), which we restate here for convenient reference:

$$A(\theta) = \sup_{\mu \in \mathcal{M}} \big\{ \langle \theta, \mu \rangle - A^*(\mu) \big\}. \tag{32}$$

It should be emphasized that equation (32) is a variational representation in two senses. First, it specifies $A(\theta)$ as the solution of a particular optimization problem in which θ plays the role of a parameter. Second, equation (32) provides a variational procedure for computing mean parameters, as stated formally in the following:

Proposition 4. *For all $\theta \in \Theta$, the supremum in equation (32) is attained uniquely at the vector $\mu \in \operatorname{ri} \mathcal{M}$ specified by:*

$$\mu = \mathbb{E}_\theta[\boldsymbol{\phi}(\mathbf{x})] = \int_{\mathcal{X}^n} \boldsymbol{\phi}(\mathbf{x}) p(\mathbf{x}; \theta) \boldsymbol{\nu}(d\mathbf{x}).$$

The essence of Proposition 4 is the following: an additional by-product of solving problem (32), apart from computing the log partition function, is the set of the mean parameters $\mu = \mathbb{E}_\theta[\boldsymbol{\phi}(\mathbf{x})]$ associated with $p(\mathbf{x};\theta)$. It is tempting, then, to assert that the problem of computing mean parameters is now solved, since we have "reduced" it to a convex optimization problem. In this context, the simple scalar examples of Table 2, for which problem (32) 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.

These difficulties motivate the use of approximations to $\mathcal{M}$ and A^*. Indeed, as shown in later sections, a broad class of methods for approximate inference are based on this strategy. The remainder of this section is devoted a more in-depth consideration of the nature of the set $\mathcal{M}$ of realizable mean parameters, as well as the dual function A^*. We also investigate particular large-scale exponential families for which the variational principle (32) is tractable; such cases provide building blocks for our later development of approximate variational principles.

4.1 Sets of realizable mean parameters

Recall that for a given set of sufficient statistics $\boldsymbol{\phi}$, the set $\mathcal{M}$ consists of all mean parameters μ that are realizable—viz:

$$\mathcal{M} \;:=\; \Big\{ \mu \in \mathbb{R}^d \;\Big|\; \exists\; p(\cdot)\,\text{s.t.} \;\int \boldsymbol{\phi}(\mathbf{x})p(\mathbf{x})\boldsymbol{\nu}(d\mathbf{x}) = \mu \Big\}. \tag{33}$$

Despite the apparent simplicity of this representation of $\mathcal{M}$, even assessing whether a single μ belongs to $\mathcal{M}$ poses a serious challenge. The difficulty stems from the fact that there can exist global—and often rather subtle—dependencies among the mean parameters associated with the vector of sufficient statistics $\boldsymbol{\phi}$.

We begin by discussing some general properties of the sets $\mathcal{M}$. We then discuss two important classes of exponential families for which $\mathcal{M}$ is straightforward to characterize—namely, arbitrary Gaussian distributions, and multinomial distributions on junction trees. Before proceeding, a remark on notation: since much of our discussion involves graphs, it is convenient to introduce the notation $\mathcal{M}(G)$, which indicates explicitly that $\mathcal{M}$ arises from a vector of sufficient statistics $\boldsymbol{\phi}$ associated with a graph G.

4.1.1 General 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. 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 5. *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 $\mathbb{R}^d$.

Remark: The necessity of the condition $\Theta = \mathbb{R}^d$ for $\mathcal{M}$ to be bounded (part (b) is clear from the boundary behavior of ∇A given in Proposition 2. However, the additional global Lipschitz condition is also necessary, as demonstrated by the Poisson family (see Table 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.

We now turn to some specific cases for which we can give explicit characterizations of $\mathcal{M}$.

4.1.2 Gaussian distributions

The exponential parameterization of a Gaussian Markov random field was described in Example 4. In this example, we consider the structure of $\mathcal{M}$ for such a model, focusing for simplicity on the case where $G = K_n$, the complete graph on n nodes. The case with arbitrary G can be dealt with by considering suitable projections of the set $\mathcal{M}_{Gauss} \equiv \mathcal{M}_{Gauss}(K_n)$ characterized here.

Associated with the exponential parameterization (17) of a multivariate Gaussian is a mean parameter vector $\mu \in \mathbb{R}^d$. It is convenient to represent μ in terms of $(n+1) \times (n+1)$ matrix, denoted by $W(\mu)$, defined in the following way:

$$W(\mu) \quad := \quad \mathbb{E}_\theta \Big\{ \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{x} \end{bmatrix} \Big\} \quad = \quad \begin{bmatrix} 1 & z^T(\mu) \\ z(\mu) & Z(\mu) \end{bmatrix}. \tag{34}$$

In this definition, $z(\mu) := \mathbb{E}_\theta[\mathbf{x}]$ is a column vector of means, whereas $Z(\mu) := \mathbb{E}_\theta[\mathbf{x}\mathbf{x}^T]$ is the $n \times n$ matrix of second order moments.

An attractive feature of the Gaussian case is that the validity of the mean parameter vector $\mu = \{\mu_s, \, \mu_{st} \mid s, t = 1, \ldots, n\}$ can be assessed very easily:

Proposition 6. *In the Gaussian case, the set $\mathcal{M}$ has the form*

$$\mathcal{M}_{Gauss} \quad = \quad \{\mu \in \mathbb{R}^d \mid W(\mu) \succ 0\}. \tag{35}$$

The geometry of the set $\mathcal{M}_{Gauss}$ can be understood as follows. Let $\mathcal{S}_+^{n+1}$ denote the cone of symmetric positive definite matrices. Then $\mathcal{M}_{Gauss}$ is the intersection of $\mathcal{S}_+^{n+1}$ with a single hyperplane, corresponding to the constraint $[W(\mu)]_{11} = 1$. As a consequence, $\mathcal{M}_{Gauss}$ is not itself a cone. For instance, in the scalar case $n = 1$, it is a parabolic set of the form $\{\mu \in \mathbb{R}^2 \mid \mu_{11} - \mu_1^2 > 0\}$.

4.1.3 Multinomial distributions

Now suppose that $\mathcal{X}^n = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \mathcal{X}_n$ is a Cartesian product of finite discrete sets (i.e., $\mathcal{X}_s = \{0, 1, \ldots, m_s - 1\}$ for each $s = 1, \ldots, n$), so that $\mathbf{x} \in \mathcal{X}^n$ is a multinomial random vector. In this case, the boundary of $\mathcal{M}$ is no longer curved, but rather formed of straight lines.

Proposition 7. *In the multinomial case, the set $\mathcal{M}$ is a polytope, meaning that it has a representation of the form*

$$\mathcal{M} \quad = \quad \{\, \mu \in \mathbb{R}^d \mid \langle a_j, \, \mu \rangle \leq b_j \quad \forall\, j \in \mathcal{J}\}, \tag{36}$$

where the index set $\mathcal{J}$ is finite. *Moreover, any extreme point is of the form $\mu_{\mathbf{e}} := \boldsymbol{\phi}(\mathbf{e})$.*

Remark: Since any convex set can be represented as the intersection of half-spaces containing it [92], the crucial part of Proposition 7, then, is that the index set $\mathcal{J}$ in equation (36) has *finite cardinality.*

Motivated by Proposition 7, we use $\mathrm{MARG}(G)$ to denote the *marginal polytope* associated with the graph G. Figure 10 provides a geometrical illustration of such an object. The extreme points (i.e., those that cannot be expressed as convex combinations of other points) are all of the form $\mu_{\mathbf{e}} := \boldsymbol{\phi}(\mathbf{e})$. Such a point is realized by the distribution $\delta_{\mathbf{e}}(\mathbf{x})$, which is equal to one if $\mathbf{x} = \mathbf{e}$ and 0 otherwise. The faces of the marginal polytope are specified by hyperplane constraints of the form $\langle a_j,\, \mu \rangle \leq b_j$. The maximal faces (i.e., those that are not contained in any other face) are known as *facets*.

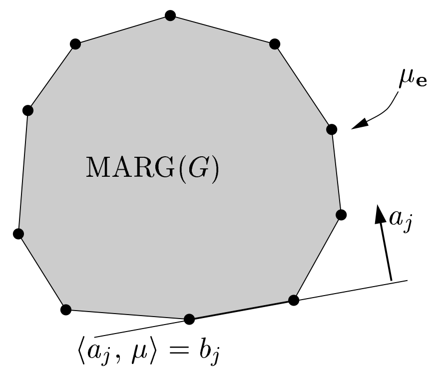


Figure 10. Geometrical illustration of a marginal polytope. Each vertex corresponds to the mean parameter $\mu_{\mathbf{e}} := \boldsymbol{\phi}(\mathbf{e})$ realized by the distribution $\delta_{\mathbf{e}}(\mathbf{x})$ that puts all of its mass on the configuration $\mathbf{e} \in \mathcal{X}^n$. The faces of the marginal polytope are specified by hyperplane constraints $\langle a_j,\, \mu \rangle \leq b_j$.

As one might expect, it turns out that the nature of a marginal polytope depends crucially on the underlying graph structure. Although the number of constraints $|\mathcal{J}|$ defining $\mathrm{MARG}(G)$ is always finite, the number can grow very quickly with increasing graph size. In this sense, Figure 10 is not at all faithful, since marginal polytopes may have an exceedingly large number of facets. The book by Deza and Laurent [32] provides a wealth of information on the binary case; as a concrete example, for a binary random vector on the complete graph with $n = 7$ nodes, the associated marginal polytope is known to have in excess of 2×10^8 facets. In contrast, tree-structured graphs are dramatically different: the number of facets grows only linearly in the number of nodes n, as shown via the following example.

Example 7 (Minimal representation of tree marginal polytope). The case of a binary random vector $\mathbf{x} \in \{0,1\}^n$ suffices to illustrate the nature of the marginal polytope $\mathrm{MARG}(T)$ for a tree-structured graph $T = (V, E(T))$. We use the minimal (Ising) representation of Example 3, with the singleton x_s for each $s \in V$, and the pairwise product $x_s x_t$ for each edge $(s,t) \in E(T)$. The relevant mean parameters in this representation, then, are as follows:

$$\mu_s \;=\; \mathbb{E}_\theta[x_s] \;=\; p(x_s = 1; \theta), \qquad \mu_{st} \;=\; \mathbb{E}_\theta[x_s\, x_t] \;=\; p(x_s = 1, x_t = 1; \theta).$$

(The fact that these mean parameters are equal to particular values of marginal probabilities justifies our terminology.) For each edge (s,t), the triplet $\{\mu_s, \mu_t, \mu_{st}\}$ uniquely determines a joint marginal $p(x_s, x_t; \mu)$ as follows:

$$p(x_s, x_t; \mu) \;=\; \begin{bmatrix} (1 + \mu_{st} - \mu_s - \mu_t) & (\mu_t - \mu_{st}) \\ (\mu_s - \mu_{st}) & \mu_{st} \end{bmatrix}.$$

Note that for any choice of $\{\mu_s, \mu_t, \mu_{st}\}$, this joint marginal satisfies the normalization constraint $\sum_{x_s, x_t} p(x_s, x_t; \mu) = 1$. Therefore, to ensure that it is a joint marginal, it is necessary and sufficient to impose non-negativity constraints on all four entries, as follows:

$$\begin{aligned} 1 + \mu_{st} - \mu_s - \mu_t &\geq 0 && (37a)\\ \mu_{st} &\geq 0 && (37b)\\ \mu_v - \mu_{st} &\geq 0 \qquad \text{for } v = s, t && (37c) \end{aligned}$$

We are now set up to apply the junction tree theorem; in particular, by Proposition 1, the full collection $\mu = \{\mu_s, s \in V\} \cup \{\mu_{st}, (s,t) \in E(T)\}$ determines a globally-consistent distribution if and only if the four inequalities (37) are satisfied for every edge. Since any tree on n nodes has $n-1$ edges, the marginal polytope for a tree-structured graph in the binary case can be characterized by $4\,(n-1)$ constraints. $\diamondsuit$

Remarks: (a) Example 7 can be extended to minimal exponential families for multinomials (i.e., $\mathcal{X}_s = \{0, 1, \ldots, m-1\}$) as well. In particular, given a minimal representation, we simply determine the inequality constraints that guarantee the existence of a pairwise marginal, and then invoke Proposition 1.
(b) Similarly, this development can be extended to junction tree models, of which ordinary trees are a particular case. See Section 2.5.2 for a description of junction trees.

Canonical overcomplete representation: One unpleasant feature of describing marginal polytopes in minimal representations is that the interpretation of constraints can be far from transparent. Consider, for instance, equation (37a): with a bit of thought and an application of the inclusion-exclusion principle, one can see that $(1 + \mu_{st} - \mu_s - \mu_t)$ is equal to the marginal probability $p(x_s = 0, x_t = 0)$, from which the non-negativity constraint follows. This interpretation, however, may not be obvious at a glance. This lack of transparency only becomes worse in the general case, where individual mean parameters need not correspond to individual marginal values.

In contrast, a judicious choice of an overcomplete exponential representation leads to easily interpretable constraints. Here we describe a particular overcomplete representation, applicable to the multinomial space $\mathcal{X}_s = \{0, 1, \ldots, m_s - 1\}$, that plays an important role in the sequel. For each $j \in \mathcal{X}_s$, let $\mathbb{I}_j(x_s)$ be an indicator function for the event $\{x_s = j\}$. Similarly, for each pair $(j,k) \in \mathcal{X}_s \times \mathcal{X}_t$, let $\mathbb{I}_{jk}(x_s, x_t)$ be an indicator for the event $\{(x_s, x_t) = (j,k)\}$. These indicator functions define the statistics for the following overcomplete representation:

$$\begin{aligned} \mathbb{I}_j(x_s) \quad &\text{for} \quad s = 1, \ldots n, \quad j \in \mathcal{X}_s && (38a)\\ \mathbb{I}_{jk}(x_s, x_t) \quad &\text{for} \quad (s,t) \in E, \quad (j,k) \in \mathcal{X}_s \times \mathcal{X}_t. && (38b) \end{aligned}$$

The overcompleteness is clear in various linear relations satisfied by the indicator functions (e.g., $\sum_{j \in \mathcal{X}_s} \mathbb{I}_j(x_s) = 1$). More generally, we can define indicators on higher order cliques; for instance, to treat a graph with a 3-clique $\{s, t, u\}$, we incorporate a term of the form $\mathbb{I}_{stu}(x_s, x_t, x_u)$. We refer to the representation defined by (38a) and (38b) as the *canonical overcomplete representation* for multinomial distributions.

An attractive feature of this representation is that mean parameters are simply local marginal probabilities—viz.:

$$\mu_{s;j} := p(x_s = j; \theta) \quad \forall\, s \in V, \qquad \mu_{st;jk} := p((x_s, x_t) = (j,k); \theta) \quad \forall\, (s,t) \in E \qquad (39)$$

For calculations in the sequel, it is convenient to use these marginals to define functional forms of the single node and joint marginal parameters as follows:

$$\mu_s(x_s) \;:=\; \sum_{j \in \mathcal{X}_s} \mu_{s;j} \mathbb{I}_j(x_s), \qquad\qquad \mu_{st}(x_s, x_t) \;:=\; \sum_{(j,k) \in \mathcal{X}_s \times \mathcal{X}_t} \mu_{st;jk} \mathbb{I}_{jk}(x_s, x_t) \tag{40}$$

More generally, marginal functions over higher-order cliques are defined in an analogous manner.

Example 8 (Tree marginals in overcomplete form). To illustrate the use of the canonical overcomplete representation, we show that the tree-structured MARG(T) has a simple and easily interpretable characterization. Consider the single node μ_s and joint pairwise marginal functions μ_{st}. As marginal distributions, they must of course be non-negative. In addition, they must satisfy normalization conditions (i.e., $\sum_{x_s} \mu_s(x_s) = 1$), and the pairwise marginalization conditions (i.e., $\sum_{x_t} \mu_{st}(x_s, x_t) = \mu_s(x_s)$). Accordingly, we define for an *arbitrary* graph G the following constraint set:

$$\text{LOCAL}(G) \;\;:=\;\; \{\, \mu \geq 0 \mid \sum_{x_s} \mu_s(x_s) = 1, \;\; \sum_{x_t} \mu_{st}(x_s, x_t) = \mu_s(x_s) \,\}, \tag{41}$$

for $(s,t) \in E$. Note that the normalization of the single node marginal, in conjunction with the marginalization constraint, imply that each joint marginal μ_{st} is also properly normalized. Since any set of local marginals (regardless of the underlying graph structure) must satisfy these local consistency constraints, we are guaranteed that MARG$(G) \subseteq$ LOCAL(G) for *any* graph G. When the graph is actually tree-structured, then the junction tree theorem, in the form of Proposition 1, guarantees that the local consistency constraints in equation (41) imply global consistency, so that in fact MARG$(T) =$ LOCAL(T). $\diamondsuit$

Remark: It is worthwhile understanding the link between the tree marginal polytope MARG(T) in the canonical overcomplete representation, and its analogue in a minimal representation. In the overcomplete representation,[9] there are a total of $d' = mn + m^2\,|E|$ mean parameters; therefore, the marginal polytope lies in $\mathbb{R}^{d'}$. However, the presence of equality constraints in equation (41) indicates the polytope actually lies strictly within an affine subset of $\mathbb{R}^{d'}$. Therefore, consistent with Proposition 5(a), it is not a full-dimensional set. Eliminating the equality constraints leads a reduced but equivalent description in a lower-dimensional space $\mathbb{R}^d$, wherein all of the constraints are one-sided inequalities. It is not difficult to show that the dimension of the reduced representation is $d = (m-1)\,n + (m-1)^2\,|E|$. In the binary case ($m = 2$), Example 7 provides an explicit representation of this reduced representation.

4.2 Nature of the dual function

We now turn to a more in-depth consideration of the nature of the dual function A^*. Its variational definition in equation (23) is both a blessing and a curse. On one hand, it guarantees that A^* is a convex and well-behaved function; however, the absence of a closed form expression for A^* presents substantial computational challenges. As with our earlier discussion of $\mathcal{M}$, important exceptions include the Gaussian and tree-structured cases.

[9] For simplicity, we are assuming that $m_s = m$ for all nodes.

4.2.1 General properties of A^*

As noted earlier, the examples given in Table 2, in which the dual A^* had a closed form, are the exception rather than the rule. In general, the dual function is defined implicitly via the composition of two functions: (i) first compute an exponential parameter $\theta(\mu)$ in the inverse image $\Lambda^{-1}(\mu)$; and then (ii) compute the negative entropy of the distribution $p(\mathbf{x};\theta(\mu))$. The block diagram in Figure 11 illustrates this decomposition of the function A^*.

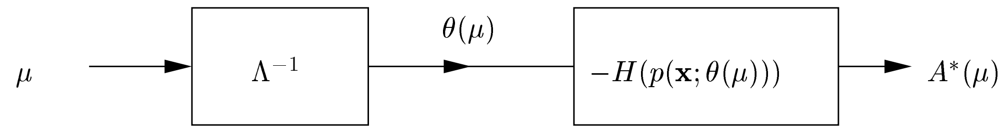


Figure 11. A block diagram decomposition of A^* as the composition of two functions. Given a marginal vector μ, first compute an exponential parameter $\theta(\mu)$ in the inverse image image Λ^{-1}, then compute the negative entropy of $p(\mathbf{x};\theta(\mu))$.

Despite the fact that A^* is not given in closed form, a number of properties can be inferred from its variational definition (23). For instance, an immediate consequence is that A^* is always convex. More specific properties of A^* depend on the nature of the exponential family, as summarized in the following:

Proposition 8. *The dual function A^* is always convex and lower semi-continuous. Moreover, in a minimal representation:*

(a) A^* is differentiable on $\operatorname{int}\mathcal{M}$, *and* $\nabla A^*(\mu) = \Lambda^{-1}(\mu)$.

(b) A^* is strictly convex.

(c) For any sequence $\{\mu^n\}$ contained in $\operatorname{int}\mathcal{M}$ *and approaching the boundary* $\operatorname{bd}\mathcal{M}$, *we have* $\lim_{n\to+\infty}\|\nabla A^*(\mu^n)\| = +\infty$.

Remarks: This result is analogous to the earlier Proposition 2, 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 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.

Despite the desirable properties guaranteed by Proposition 8, the function A^* presents substantial computational challenges. Indeed, both operations in the decomposition of A^* given in Figure 11 are troublesome. First of all, the inverse image $\Lambda^{-1}(\mu)$ of the mean parameter mapping, while well-defined mathematically, does not usually have a closed form expression. It is typically necessary to resort to iterative methods, such as iterative proportional fitting or generalized iterative scaling [e.g., 28, 27], in order to compute this mapping. In any case, these algorithms presuppose that it is possible to perform exact inference, which is the problem that we are trying to solve in the first place. Second, even if we were able to compute a parameter $\theta(\mu) \in \Lambda^{-1}(\mu)$, there remains the task of computing the entropy $H(p(\mathbf{x};\theta(\mu)))$, which is not possible in general for a large problem.

To parallel our earlier discussion of $\mathcal{M}$, we now turn to two important cases where A^* can be characterized in closed form, even for large problems.

4.2.2 Gaussian distributions

Consider the case of a multivariate Gaussian random vector $\mathbf{x}$, discussed previously in Section 4.1.2. It is well-known [23] that the Gaussian entropy is $\frac{1}{2}\log\det\mathrm{cov}(\mathbf{x}) + \frac{n}{2}\log 2\pi e$, where $\mathrm{cov}(\mathbf{x})$ is the $n \times n$ covariance matrix of $\mathbf{x}$. As originally defined in equation (34), let $W(\mu)$ be the matrix of mean parameters associated with $\mathbf{x}$:

$$W(\mu) \;=\; \begin{bmatrix} 1 & z^T(\mu) \\ z(\mu) & Z(\mu) \end{bmatrix}. \tag{42}$$

Applying the Schur complement formula [54] yields $\det W(\mu) = \det[Z(\mu) - z(\mu)z^T(\mu)] = \det(\mathrm{cov}(\mathbf{x}))$, from which we conclude that

$$A^*_{Gauss}(\mu) \;=\; -\frac{1}{2}\log\det W(\mu) - \frac{n}{2}\log 2\pi e, \tag{43}$$

valid for all $\mu \in \mathcal{M}_{Gauss}$. (To understand the negative signs, recall from Theorem 2 that A^* is equal to negative entropy for $\mu \in \mathcal{M}_{Gauss}$.) Combining this exact expression for A^*_{Gauss} with our characterization of $\mathcal{M}_{Gauss}$ from Proposition 6 leads to

$$A_{Gauss}(\theta) \;=\; \sup_{W(\mu)\succ 0,\; [W(\mu)]_{11}=1} \Big\{ \langle\langle U(\theta),\; W(\mu)\rangle\rangle + \frac{1}{2}\log\det W(\mu) + \frac{1}{2}\log 2\pi e \Big\}, \tag{44}$$

which corresponds to the variational principle (32) specialized to the Gaussian case.

If $W(\mu) \succ 0$ were the only constraint, then, using the fact that $\nabla \log\det W = W^{-1}$ for any symmetric positive matrix W, the optimal solution to problem (44) would simply be $W(\mu) = -2[U(\theta)]^{-1}$. Accordingly, if we enforce the constraint $[W(\mu)]_{11} = 1$ using a Lagrange multiplier λ, then it follows from the Karush-Kuhn-Tucker conditions [8] that the optimal solution will assume the form $W(\mu) = -2[U(\theta) + \lambda^* E_{11}]^{-1}$, where λ^* is the optimal setting of the Lagrange multiplier and E_{11} is an $(n+1) \times (n+1)$ matrix with a one in the upper left hand corner, and zero in all other entries. Finally, using the standard formula for the inverse of a block-partitioned matrix [54], it is straightforward to verify that the blocks in the optimal $W(\mu)$ are related to the blocks of $U(\theta)$ by the relations:

$$Z(\mu) - z(\mu)z^T(\mu) \;=\; -2[Z(\theta)]^{-1} \tag{45a}$$

$$z(\mu) \;=\; -[Z(\theta)]^{-1}\, z(\theta) \tag{45b}$$

(The multiplier λ^* turns out not to be involved in these particular blocks.) In order to interpret these relations, it is helpful to return to the definition of $U(\theta)$ given in equation (17), and the Gaussian density of equation (18). In this way, we see that equation (45a) corresponds to the fact[10] that the covariance matrix is the inverse of the precision matrix, whereas equation (45b) corresponds to the normal equations for the mean $z(\mu)$ of a Gaussian. Thus, as a special case of the general variational principle (32), we have re-derived the familiar equations for Gaussian inference.

4.2.3 Tree-structured problems

We now return to the case of tree-structured multinomial distributions, discussed previously in Section 4.1.3. Another consequence of the junction tree representation is that A^* has a closed-form expression for any distribution defined by a junction tree. The case of an ordinary tree

[10] The factor of negative two in equation (45a) arises due to the exponential parameterization of the multivariate Gaussian used in equation (18).

$T = (V, E(T))$ suffices to illustrate. In the canonical overcomplete representation of equation (38), the mean parameters $\mu = \{\mu_s, \mu_{st}\}$ correspond to local marginals associated with single nodes and edges. In particular, we make use of the local marginal functions $\mu_s(x_s)$ and $\mu_{st}(x_s, x_t)$ defined in equation (40).

By a special case of the junction tree decomposition (9), any tree-structured distribution factorizes in terms of the local marginal distributions as follows:

$$p(\mathbf{x}) \;\; = \;\; \prod_{s \in V} \mu_s(x_s) \prod_{(s,t) \in E(T)} \frac{\mu_{st}(x_s, x_t)}{\mu_s(x_s)\mu_t(x_t)}. \tag{46}$$

Using the definition of Boltzmann-Shannon entropy (24) and Theorem 2(a), this factorization leads immediately to an explicit form for the dual function, valid for all $\mu \in \mathrm{MARG}(T)$:

$$A^*_{tree}(\mu) \;\; = \;\; -\sum_{s \in V} H_s(\mu_s) + \sum_{(s,t) \in E(T)} I_{st}(\mu_{st}). \tag{47}$$

Here H_s and I_{st} are, respectively, single node entropy and mutual information terms:

$$H_s(\mu_s) \;:=\; -\sum_{x_s} \mu_s(x_s) \log \mu_s(x_s), \qquad I_{st}(\mu_{st}) \;:=\; \sum_{x_s, x_t} \mu_{st}(x_s, x_t) \log \frac{\mu_{st}(x_s, x_t)}{\mu_s(x_s)\mu_t(x_t)}. \tag{48}$$

Again, it is worth combining this expression with our characterization of the marginal polytope $\mathrm{MARG}(T)$ from Example 8. In this way, we obtain the following tree-structured form of the general variational principle (32):

$$\max_{\mu \in \mathrm{MARG}(T)} \big\{ \langle \theta, \, \mu \rangle + \sum_{s \in V} H_s(\mu_s) - \sum_{(s,t) \in E(T)} I_{st}(\mu_{st}) \big\}. \tag{49}$$

Note that this problem has a simple structure: the cost function is concave and differentiable, and the constraint set $\mathrm{MARG}(T)$ is a polytope specified by a small ($\mathcal{O}(n)$) number of constraints. In fact, we will establish, as a corollary of our analysis of the Bethe approximation in Section 6, that the sum-product updates (7) are a Lagrangian-based method for solving problem (49).

In overview, we have seen how the general variational principle (32) takes explicit and simple forms for multivariate Gaussians on arbitrary graphs, and discrete random vectors on tree-structured graphs. The perspective given here clarifies that inference in such models is relatively easy because the underlying variational problem has simple structure. In the following sections, we demonstrate how the Gaussian and tree-structured characterizations of $\mathcal{M}$ and A^*, though presented as exact representations in the preceding sections, also play an important role in approximate inference.

5 Mean field theory

This section is devoted to a discussion of mean field methods, which are classical techniques in statistical physics [e.g., 83, 6, 20]. From the perspective of this paper, mean field theory is based on the variational principle of equation (32), but entails imposing limitations on the optimization. More specifically, as discussed in Section 4, there are two fundamental difficulties associated with the variational principle (32): the nature of the constraint set $\mathcal{M}$, and the lack of an explicit form for the dual function A^*. Mean field theory entails limiting the optimization to a subset of distributions for which A^* is relatively easy to characterize. Throughout this section, we will refer to a distribution with this property as a *tractable* distribution.

5.1 Tractable families

Let H represent a subgraph of G over which it feasible to perform exact calculations (e.g., a graph with small treewidth). We refer to H as a *tractable subgraph*. In an exponential formulation, the set of all distributions that respect the structure of H can be represented by a linear subspace of exponential parameters. More specifically, letting $\mathcal{I}(H)$ be the subset of indices associated with cliques in H, the set of exponential parameters corresponding to distributions structured according to H is given by:

$$\mathcal{E}(H) \quad := \quad \{\theta \in \Theta \mid \theta_\alpha = 0 \quad \forall \ \alpha \in \mathcal{I} \backslash \mathcal{I}(H)\}. \tag{50}$$

Note that $\mathcal{E}(H)$ is an affine subset, or an e-flat manifold in the sense of information geometry [4]. We consider some examples to illustrate:

Example 9 (Tractable subgraphs). The simplest (non-trivial) instance of a tractable subgraph is the completely disconnected graph $H_0 = (V, \emptyset)$. Permissible parameters belong to the subspace $\mathcal{E}(H_0) := \{\theta \in \Theta \mid \theta_{st} = 0 \ \forall \ (s,t) \in E\}$, where θ_{st} refers to the collection of exponential parameters associated with edge (s,t). The associated distributions are of the product form $p(\mathbf{x};\theta) = \prod_{s \in V} p(x_s; \theta_s)$, where θ_s refers to the collection of exponential 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 exponential parameters corresponding to vertices and edges in T, but we must set to zero any exponential parameters corresponding to edges not in the tree. Accordingly, the subspace of tree-structured distributions is given by $\mathcal{E}(T) = \{\theta \mid \theta_{st} = 0 \ \ \forall \ (s,t) \notin E(T)\}$. $\diamondsuit$

For a given subgraph H, consider the set of all possible mean parameters that are realizable by tractable distributions:

$$\mathcal{M}_{tract}(G; H) \quad := \quad \{\mu \in \mathbb{R}^d \mid \mu = \mathbb{E}_\theta[\boldsymbol{\phi}(\mathbf{x})] \text{ for some } \theta \in \mathcal{E}(H)\}. \tag{51}$$

The notation $\mathcal{M}_{tract}(G; H)$ indicates that mean parameters in this set correspond to potentials on the graph G, but that they must be realizable by a tractable distribution—i.e., one that respects the structure of H. Since any μ that arises from a tractable distribution is certainly a valid mean parameter, the inclusion $\mathcal{M}_{tract}(G; H) \subseteq \mathcal{M}(G)$ always holds. In this sense, $\mathcal{M}_{tract}$ is an *inner approximation* to the set $\mathcal{M}$ of realizable mean parameters.

5.2 Optimization and lower bounds

We now have the necessary ingredients to develop the mean field approach to approximate inference. Let $p(\mathbf{x}; \theta)$ denote the *target distribution* that we are interested in approximating. The basis of the mean field method is the following fact: any valid mean parameter specifies a lower bound on the log partition function.

Proposition 9 (Mean field lower bound). *For any $\mu \in \operatorname{ri} \mathcal{M}$, we have the following lower bound:*

$$A(\theta) \quad \geq \quad \langle \theta, \mu \rangle - A^*(\mu). \tag{52}$$

Proof. In the context of our exposition, the validity of this lower bound is an immediate consequence of the variational principle (32). Alternatively, it can be established via Jensen's inequality. For any mean parameter $\mu \in \operatorname{ri}\mathcal{M}$, Theorem 1 guarantees the existence of some $\theta(\mu) \in \Lambda^{-1}(\mu)$. Using the distribution $p(\mathbf{x};\theta(\mu))$, we write:

$$\begin{aligned} A(\theta) &= \log \int_{\mathcal{X}^n} p(\mathbf{x};\theta(\mu)) \frac{\exp\{\langle \theta,\, \boldsymbol{\phi}(\mathbf{x})\rangle\}}{p(\mathbf{x};\theta(\mu))}\, \boldsymbol{\nu}(d\mathbf{x}) \\ &\overset{(a)}{\geq} \int_{\mathcal{X}^n} p(\mathbf{x};\theta(\mu)) \Big[\langle \theta,\, \boldsymbol{\phi}(\mathbf{x})\rangle - \log p(\mathbf{x};\theta(\mu))\Big] \boldsymbol{\nu}(d\mathbf{x}) \\ &\overset{(b)}{=} \langle \theta,\, \mu\rangle - A^*(\mu). \end{aligned}$$

In this argument, step (a) follows from Jensen's inequality [e.g., 53], whereas step (b) follows from the relations $\mathbb{E}_{\theta(\mu)}[\boldsymbol{\phi}(\mathbf{x})] = \mu$, and $A^*(\mu) = -H(p(\mathbf{x};\theta(\mu)))$ from Theorem 2(a). □

Since the dual function A^* typically lacks an explicit form, it is not possible, at least in general, to compute the lower bound (52). The mean field approach circumvents this difficulty by restricting the choice of μ to a tractable subset $\mathcal{M}_{tract}(G;H)$, for which the dual function has an explicit form A^*_H. As long as μ belongs to $\mathcal{M}_{tract}(G;H)$, then the lower bound (52) will be computable.

Of course, for a non-trivial class of tractable distributions, there are many such bounds. The goal of the mean field method is the natural one: find the best approximation μ^{MF}, as measured in terms of the tightness of the bound. This optimal approximation is specified as the solution of the optimization problem

$$\sup_{\mu \in \mathcal{M}_{tract}(G;H)} \big\{\langle \mu,\, \theta\rangle - A^*_H(\mu)\big\}. \tag{53}$$

The optimal value specifies a lower bound on $A(\theta)$, and it is (by definition) the best one that can be obtained by using a distribution from the tractable class.

An important alternative interpretation of the mean field solution (53) is as minimizing the Kullback-Leibler divergence between the approximating (tractable) distribution and the target distribution. In particular, for a given mean parameter $\mu \in \mathcal{M}_{tract}(G;H)$, the difference between the log partition function $A(\theta)$ and the quantity $\langle \mu,\, \theta\rangle - A^*_H(\mu)$ to be maximized is equivalent to

$$D(\mu \,\|\, \theta) \;=\; A(\theta) + A^*_H(\mu) - \langle \mu,\, \theta\rangle,$$

corresponding to the mixed form of the Kullback-Leibler divergence defined in equation (30). On the basis of this relation, it can be seen that solving the variational problem (53) is equivalent to minimizing the KL divergence $D(\mu \,\|\, \theta)$ subject to the constraint that $\mu \in \mathcal{M}_{tract}(G;H)$. Note that this problem entails a minimization over mean parameters with respect to the *first* argument of the Kullback-Leibler divergence. As a consequence, the mean field procedure is an operation that differs in fundamental ways from the I-projection with KL divergences [3, 26].

5.2.1 Naive mean field updates

The *naive mean field* approach corresponds to choosing a fully factorized or product distribution in order to approximate the original distribution. The naive mean field updates are a particular set of recursions for finding a stationary point of the resulting optimization problem.

Example 10. As an illustration, we derive the naive mean field updates for the Ising model introduced in Example 3. Letting H_0 denote the fully disconnected graph (i.e., no edges), the tractable set $\mathcal{M}_{tract}(G;H_0)$ consists of all mean parameters $\{\mu_s, \mu_{st}\}$ that arise from a product distribution. Explicitly, in this binary case, we have

$$\mathcal{M}_{tract}(G;H_0) \;:=\; \{(\mu_s, \mu_{st}) \mid 0 \;\leq\; \mu_s \;\leq 1,\;\; \mu_{st} = \mu_s\, \mu_t\;\}.$$

Moreover, the negative entropy of a product distribution over binary random variables decomposes into the sum $A^*_{H_0}(\mu) = \sum_{s \in V} \big[\mu_s \log \mu_s + (1-\mu_s)\log(1-\mu_s)\big]$. Accordingly, the associated naive mean field problem takes the form $\max_{\mu \in \mathcal{M}_{tract}(G;H_0)} \big\{\langle \mu,\, \theta\rangle - A^*_{H_0}(\mu)\big\}$. In this particular case, it is straightforward to eliminate μ_{st} by replacing it by the product $\mu_s\mu_t$. Doing so leads to a reduced form of the problem:

$$\max_{\{\mu_s\} \in [0,1]^n} \Big\{ \sum_{s \in V} \theta_s \mu_s + \sum_{(s,t) \in E} \theta_{st}\mu_s\mu_t - \sum_{s \in V} \big[\mu_s \log \mu_s + (1-\mu_s)\log(1-\mu_s)\big] \Big\} \tag{54}$$

Let F denote the function of μ within curly braces in equation (54). It can be seen that for any $s \in V$, it is strictly concave in μ_s when all the other coordinates are held fixed. Moreover, it is straightforward to show that the maximum over μ_s with $\mu_t, t \neq s$ fixed is attained in the interior $(0,1)$, and can be found by taking the gradient and setting it equal to zero. Doing so yields the following update for μ_s:

$$\mu_s \quad \leftarrow \quad \sigma\big(\theta_s + \sum_{t \in \mathcal{N}(s)} \theta_{st}\mu_t\big), \tag{55}$$

where $\sigma(z) := [1+\exp(-z)]^{-1}$ is the logistic function. Applying equation (55) iteratively to each node in succession amounts to performing coordinate ascent of the mean field variational problem (54). Thus, we have derived the update equation presented earlier in equation (12). $\diamondsuit$

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 11 (Gaussian mean field). The mean parameters for a multivariate Gaussian are of the form $\mu_s = \mathbb{E}[x_s]$, $\mu_{ss} = \mathbb{E}[x_s^2]$ and $\mu_{st} = \mathbb{E}[x_s x_t]$ for $s \neq t$. Using only Gaussians in product form, the set of tractable mean parameters takes the form

$$\mathcal{M}_{tract}(G;H_0) \quad = \quad \{\mu \in \mathbb{R}^d \mid \mu_{st} = \mu_s\mu_t \; \forall s \neq t, \;\; \mu_{ss} - \mu_s^2 > 0\,\}.$$

As with naive mean field on the Ising model, the constraints $\mu_{st} = \mu_s\mu_t$ for $s \neq t$ can be imposed directly, thereby leaving only the inequality $\mu_{ss} - \mu_s^2 > 0$ for each node. The negative entropy of a Gaussian in product form can be written as $A^*_{Gauss}(\mu) = -\sum_{s=1}^n \frac{1}{2}\log(\mu_{ss} - \mu_s^2) - \frac{n}{2}\log 2\pi e$. Combining A^*_{Gauss} with the constraints leads to the naive MF problem for a multivariate Gaussian:

$$\sup_{\{(\mu_s, \mu_{ss}) \mid \mu_{ss} - \mu_s^2 > 0\}} \big\{\langle\!\langle U(\theta),\, W(\mu)\rangle\!\rangle + \sum_{s=1}^n \frac{1}{2}\log(\mu_{ss} - \mu_s^2) + \frac{n}{2}\log 2\pi e\big\}.$$

Here it should be understood that any terms $\mu_{st}, s \neq t$ contained in $W(\mu)$ are replaced with the product $\mu_s\mu_t$.

Taking derivatives with respect to μ_{ss} and μ_s and re-arranging yields the stationary conditions $\frac{1}{2(\mu_{ss} - \mu_s^2)} = -\theta_{ss}$ and $\frac{\mu_s}{2(\mu_{ss} - \mu_s^2)} = \theta_s + \sum_{t \in \mathcal{N}(s)} \theta_{st}\mu_t$. Since $\theta_{ss} < 0$, we can combine both equations into the update $\mu_s \leftarrow -\frac{1}{\theta_{ss}}\big\{\theta_s + \sum_{t \in \mathcal{N}(s)} \theta_{st}\mu_t\big\}$. The resulting algorithm is equivalent, in fact, to the Gauss-Jacobi method for solving the quadratic system associated with the Gaussian problem. Therefore, under suitable conditions the algorithm will converge [30], in which case the algorithm computes the correct mean vector $[\mu_1 \ldots \mu_n]$. $\diamondsuit$

5.2.2 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 [94], and further developed by various researchers [e.g., 117].

Here we discuss a general form of the updates for an approximation based on an arbitrary subgraph H of the original graph G. We make no claims as to the practical advantages of these updates; rather, the main goal here is the conceptual one of understanding the structure of the solution. Depending on the particular context, other types of updates [117, 56] or techniques from nonlinear programming may be more suitable for solving the mean field problem (53).

Let $\mathcal{I}(H)$ be the subset of indices corresponding to sufficient statistics associated with H, and let $\mu(H) := \{\mu_\alpha \mid \alpha \in \mathcal{I}(H)\}$ be the associated set of mean parameters. The mean field problem has the following key properties:

(a) the subvector $\mu(H)$ can be an arbitrary member of $\mathcal{M}(H)$, the set of realizable mean parameters defined by the subgraph H.

(b) the dual function A^*_H actually depends only on $\mu(H)$, and *not* on mean parameters μ_β for indices β in the complement $\mathcal{I}^c(H) := \mathcal{I}(G)\backslash\mathcal{I}(H)$.

Of course, mean parameters μ_β with $\beta \in \mathcal{I}^c(H)$ 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 nonlinear way by the choice of $\mu(H)$. Accordingly, for each $\beta \in \mathcal{I}^c(H)$, we write $\mu_\beta = g_\beta(\mu(H))$ for some nonlinear function g_β, of which particular examples are given below. Based on these observations, the optimization problem (53) can be rewritten in the form

$$\sup_{\mu \in \mathcal{M}_{tract}(G;H)} \big\{\langle \theta,\, \mu\rangle - A^*_H(\mu)\big\} = \sup_{\mu(H) \in \mathcal{M}(H)} \big\{ \sum_{\alpha \in \mathcal{I}(H)} \theta_\alpha \mu_\alpha + \sum_{\alpha \in \mathcal{I}^c(H)} \theta_\alpha g_\alpha(\mu(H)) - A^*_H(\mu(H))\big\}. \quad (56)$$

On the LHS, the optimization takes place over vector $\mu \in \mathcal{M}_{tract}(G;H)$, which is of the same dimension as $\theta \in \Theta \subseteq \mathbb{R}^d$. The optimization on the RHS, in contrast, takes place over a lower-dimensional vector $\mu(H) \in \mathcal{M}(H)$.

To illustrate this transformation, consider the case of naive mean field for the Ising model, where $H \equiv H_0$ is the completely disconnected graph. In this case, each edge $(s,t) \in E$ corresponds to an index in the set $\mathcal{I}^c(H_0)$; moreover, for any such edge, we have $g_{st}(\mu(H_0)) = \mu_s\mu_t$. Since H_0 is the completely disconnected graph, $\mathcal{M}(H_0)$ is simply the hypercube $[0,1]^n$. Therefore, for this particular example, the RHS of equation (56) is equivalent to equation (54).

Returning to the general case, let $F(\mu(H))$ denote the cost function on the RHS of equation (56). Taking derivatives with respect to some μ_β with $\beta \in \mathcal{I}(H)$ yields:

$$\frac{\partial F}{\partial \mu_\beta}(\mu(H)) \quad = \quad \theta_\beta + \sum_{\alpha \in \mathcal{I}(G)\backslash\mathcal{I}(H)} \theta_\alpha \frac{\partial g_\alpha}{\partial \mu_\beta}(\mu(H)) - \frac{\partial A^*_H}{\partial \mu_\beta}(\mu(H)). \quad (57)$$

From Proposition 8, the derivative $\frac{\partial A^*_H}{\partial \mu_\beta}$ defines the inverse moment mapping (i.e., from mean parameters to exponential parameters). Consequently, this derivative term is equal to the exponential parameter associated with $\mu_\beta(H)$, which we denote by $\gamma_\beta(H)$. We then set $\frac{\partial F}{\partial \mu_\beta}$ to zero and re-arrange to obtain a generalized MF update:

$$\gamma_\beta(H) \quad \leftarrow \quad \theta_\beta + \sum_{\alpha \in \mathcal{I}(G) \backslash \mathcal{I}(H)} \theta_\alpha \frac{\partial g_\alpha}{\partial \mu_\beta}(\mu(H)). \tag{58}$$

After any such update, it is then necessary to adjust all of the mean parameters $\mu_\delta(H)$ that depend on $\gamma_\beta(H)$ (e.g., via junction tree updates), so that global consistency is maintained.

Let us check that equation (58) reduces appropriately to the naive mean field updates (55), when $H = H_0$ is the completely disconnected graph. In particular, for product distributions on the Ising model, we have $g_{st}(\mu(H_0)) = \mu_s \mu_t$ 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, equation (58) is equivalent to $\gamma_s(H_0) \leftarrow \theta_s + \sum_{t \in \mathcal{N}(s)} \theta_{st} \mu_t$. In the product distribution defined on H_0, only the mean parameter μ_s depends on $\gamma_s(H_0)$ (and vice versa); more concretely, $\gamma_s(H_0)$ and μ_s are linked by the logistic transform. Consequently, μ_s is updated by applying the logistic function $\sigma(\cdot)$, which recovers equation (55).

Example 12 (Structured MF for factorial HMMs). To provide a more interesting example of the updates (58), consider a factorial hidden Markov model, as described in Ghahramani and Jordan [47]. Figure 12(a) shows the original model, which consists of a set of M Markov chains ($M = 3$ in this diagram), which share at each time a common observation (shaded nodes). Although the separate chains are a priori independent, the common observation induces an effective coupling between all nodes at each time. 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

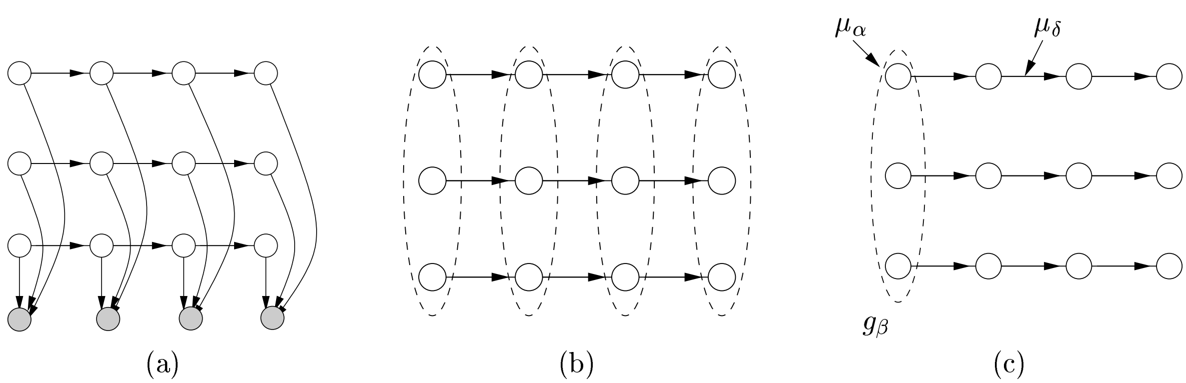


Figure 12. Structured mean field approximation for a factorial HMM. (a) Original model consists of a set of hidden Markov models (defined on chains), coupled at each time by a common observation. (b) An equivalent model, where the ellipses represent interactions among all nodes at a fixed time, induced 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.

distribution in this case is based on the subgraph H consisting of the decoupled set of M chains, as illustrated in panel (c).

Now consider the nature of the quantities $g_\beta(\mu(H))$, which arise in the cost function (56). In this case, any function g_β will be defined 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 $g_\beta(\mu(H))$ will decouple into a product of terms of the form $f_i(\{\mu_i(H)\})$, where each f_i is some function of the mean parameters $\{\mu_i\} \equiv \{\mu_i(H)\}$ 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, all intermediate quantities necessary for the updates can be calculated 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 $g_\beta(\mu(H))$ will ever depend on mean parameters associated with edges in 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 $\beta \in \mathcal{I}(G) \backslash \mathcal{I}(H)$. As an immediate consequence of these derivatives vanishing, the mean field exponential parameter $\gamma_\delta(H)$ remains equal to to θ_δ for all iterations of the updates (58). Any intermediate junction tree steps to maintain consistency will not affect $\gamma_\delta(H)$ 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. $\diamondsuit$

In addition to structured mean field, there are various other extensions to naive mean field, which we mention only in passing here. A large class of techniques, including linear response theory and the TAP method [e.g., 87, 59, 81], seek to improve the mean field approximation by introducing higher-order correction terms. Typically, the lower bound on the log partition function is not usually preserved by these higher-order methods. Leisinck and Kappen [70] demonstrated how to generate tighter lower bounds based on higher-order expansions.

5.3 Non-convexity of mean field

An important fact about the mean field approach is that the variational problem (53) may be non-convex, so that there may be local minima, and the mean field updates can have multiple solutions. Here we explore the source of the non-convexity, which can be understood in several different ways.

Consider first the representation of the mean field problem on the RHS of equation (56). The constraint set in this formulation—namely, $\mathcal{M}(H)$—is certainly convex. The cost function consists of a (concave) entropy term $-A^*_H(\mu)$ and a set of terms $\sum_{\alpha \in \mathcal{I}(H)} \theta_\alpha \mu_\alpha$ that are linear in μ. In contrast, the terms $\sum_{\alpha \notin \mathcal{I}(H)} \theta_\alpha g_\alpha(\mu)$ involve the *nonlinear* functions g_α, so that they may introduce non-convexity. For example, in the case of naive mean field on the Ising model, these functions are monomials of the form $\mu_s \mu_t$. Consequently, the overall cost function in equation (54) includes a quadratic form in $\{\mu_v\}$, so that it need not be convex in general.

In these simple cases, it can be seen explicitly how the nonlinear functions $g_\alpha(\mu)$ lead to non-convexity in the MF problem. In order to gain a more general and geometric understanding of this non-convexity, let us return to the form of mean field variational problem given in equation (53). In this formulation, observe that the function to be maximized—namely, $\langle \mu, \theta \rangle - A^*_H(\mu)$—is always a concave function of μ. Consequently, the source of any non-convexity (in the formulation (53)) must lie in the nature of the constraint set $\mathcal{M}_{tract}(G; H)$. To provide some geometric intuition for this set, let us return again to naive mean field on the Ising model.

Example 13 (Non-convexity of $\mathcal{M}_{tract}$). Consider a pair of binary variables on the (trivial) graph G consisting of a single edge. In the standard minimal representation, there are three mean parameters—μ_1, μ_2 and μ_{12}. From the development of Example 7, the marginal polytope $\mathcal{M}(G) \equiv \mathrm{MARG}(G)$ is fully characterized by the four inequalities $1 + \mu_{12} - \mu_1 - \mu_2 \geq 0$, $\mu_{12} \geq 0$, and $\mu_s - \mu_{12} \geq 0$ for $s = 1, 2$. 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:

$$\mu_{12} \geq 2\mu_1 - 1, \qquad \mu_{12} \geq 0, \qquad \mu_1 \geq \mu_{12}. \tag{59}$$

Figure 13 shows the resulting two-dimensional polytope, shaded in gray. Let us now consider the

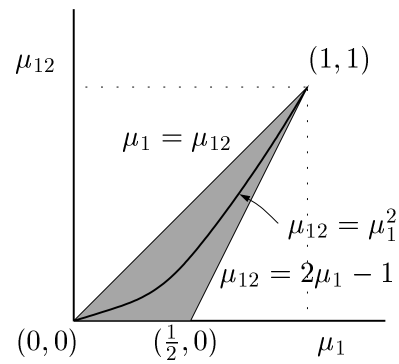


Figure 13. Non-convexity of the set of fully factorized marginals for a pair of binary variables. The gray area shows the polytope defined by equation (59), corresponding to the intersection of $\mathcal{M}(G)$ with the hyperplane $\mu_1 = \mu_2$. The (non-convex) quadratic curve $\mu_{12} = \mu_1^2$ corresponds to the intersection of $\mu_1 = \mu_2$ with the set $\mathcal{M}_{tract}(G; H_0)$ of fully factorized marginals.

intersection between $\mu_1 = \mu_2$ and the set of factorized marginals $\mathcal{M}_{tract}(G; H_0)$. It is easy to see that this intersection is given by the equation $\mu_{12} = \mu_1^2$. This quadratic curve lies within the two-dimensional polytope described by the equations (59), as illustrated in Figure 13. Since this quadratic set is not convex, this establishes that $\mathcal{M}_{tract}$ is not convex either. (If $\mathcal{M}_{tract}$ were convex, then its intersection with any hyperplane would also be convex.) $\diamondsuit$

Equipped with intuition from this example, we can formulate a result that characterizes the non-convexity of mean field approximations more generally:

Proposition 10 (Non-convexity). *Suppose that* $\mathrm{cl}\, \mathcal{M}(G)$ *contains no full lines. Consider a set of tractable mean parameters* $\mathrm{cl}\, \mathcal{M}_{tract}(G; H) \subsetneq \mathrm{cl}\, \mathcal{M}(G)$ *that contains all extreme points and directions of* $\mathrm{cl}\, \mathcal{M}(G)$. *Then* $\mathcal{M}_{tract}(G; H)$ *is a non-convex set.*

Proof. The assumption that $\mathrm{cl}\, \mathcal{M}(G)$ contains no full lines guarantees that it can be represented as the convex hull of its extreme points and directions (Thm. 18.5, [92]). Since $\mathrm{cl}\, \mathcal{M}_{tract}(G; H)$ contains these extreme points and directions by assumption, its convex hull must be equal to $\mathrm{cl}\, \mathcal{M}(G)$. Therefore, $\mathrm{cl}\, \mathcal{M}_{tract}(G; H)$ cannot be convex, since it is properly contained within $\mathrm{cl}\, \mathcal{M}(G)$. (If it were convex, then $\mathrm{cl}\, \mathcal{M}_{tract}(G; H) = \mathrm{conv}\, \mathrm{cl}\, \mathcal{M}_{tract}(G; H) = \mathrm{cl}\, \mathcal{M}(G)$, which is a contradiction.) $\square$

The assumptions underlying Proposition 10 hold in most applications of mean field. Certainly, the set of tractable mean parameters is invariably a subset of $\mathrm{cl}\, \mathcal{M}(G)$, since we are trying to approximate a model assumed to be intractable. Moreover, the set $\mathrm{cl}\, \mathcal{M}_{tract}(G; H)$ typically contains the extreme points and directions of $\mathrm{cl}\, \mathcal{M}(G)$. Let us consider some examples to illustrate:

Multinomial: In this case, Proposition 7 ensures that $\operatorname{cl}\mathcal{M}(G) = \operatorname{cl}\mathrm{MARG}(G)$ is bounded, and so contains no full lines or extreme directions. Its extreme points are simply the vertices $\mu_{\mathbf{e}}$, as illustrated below in Figure 14. Whether $\operatorname{cl}\mathcal{M}(G; H)$ is realized by product distributions (as in

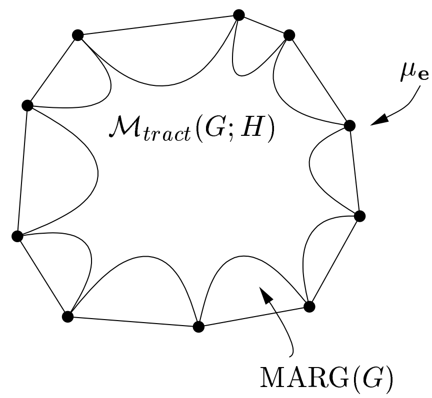


Figure 14. The set $\mathcal{M}_{tract}(G; H)$ of mean parameters that arise from tractable distributions is a non-convex inner bound on $\mathcal{M}(G)$. Illustrated here is the multinomial case where $\mathcal{M}(G) \equiv \mathrm{MARG}(G)$ is a polytope. The circles correspond to mean parameters that arise from delta distributions, and belong to both $\mathcal{M}(G)$ and $\mathcal{M}_{tract}(G; H)$.

naive mean field) or by more structured distributions, it will contain these vertices (since delta distributions are certainly product distributions). Therefore, the set $\mathcal{M}_{tract}(G; H)$ is a *non-convex inner approximation* to the polytope $\mathrm{MARG}(G)$.

Gaussian case: In this case, it is clear that $\operatorname{cl}\mathcal{M}_{Gauss}$ contains no full lines (since, e.g., $\mu_{ss} = \mathbb{E}[x_s^2]$ is always non-negative). It can also be verified that although $\operatorname{cl}\mathcal{M}_{Gauss}$ contains half-lines, none of them are extreme directions. Lastly, it can be shown that the extreme points of $\operatorname{cl}\mathcal{M}_{Gauss}$ correspond to the mean parameters $\mu_{\mathbf{e}}$ such that the matrix $W(\mu_{\mathbf{e}})$ is rank one. Such mean parameters are realized by delta distributions (i.e., the limit of a Gaussian as the covariance shrinks to zero), so that they will also be realized by typical sets of tractable distributions.

It should be noted that the non-convexity of the mean field approximation has important consequences. First, there are often multiple local minima, so that in practical terms, the result of applying mean field updates can be sensitive to the initial conditions. Second, the mean field method can exhibit "spontaneous symmetry breaking", wherein the mean field approximation is asymmetric even though the original problem is perfectly symmetric; see Jaakkola [56] for an illustration of this phenomenon. Despite this non-convexity, the mean field approximation becomes exact for certain types of models as the number of nodes n grows to infinity (i.e., in the "thermodynamic" limit) [6, 121]. Such exact cases include the ferromagnetic Ising model (i.e., $\theta_{st} > 0$ for all $(s, t) \in E$), defined either on the complete graph K_n, or on the infinite-dimensional lattice (i.e., Z^d as $d \to +\infty$).

5.4 Parameter estimation and variational EM

In this section, we consider the problem of parameter estimation, focusing specifically on the case in which a subset of variables are observed whereas others are unobserved (i.e., "latent" or "hidden"). It is this setting in which the expectation-maximization (EM) algorithm provides a general approach to maximum likelihood parameter estimation [31]. 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 [25, 78]. 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 clarifies how the mean field approach is particularly appropriate for this task.

A brief outline of our presentation in this section is as follows. In the exponential family setting, 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 that are associated with the EM algorithm. In the specific case of mean field algorithms, however, a convergence guarantee is retained: in particular, the algorithm will converge to a stationary point of a lower bound for the likelihood function.

More precisely, suppose that the set of random variables is partitioned into *observed* variables $\mathbf{y}$ and *unobserved* variables $\mathbf{x}$, and that the probability model is a joint exponential family distribution for $(\mathbf{y}, \mathbf{x})$:

$$p(\mathbf{y}, \mathbf{x}; \theta) \;\; = \;\; \exp\big\{\langle \theta,\, \boldsymbol{\phi}(\mathbf{y}, \mathbf{x})\rangle \; - \; A(\theta)\big\}. \tag{60}$$

Given an observation $\mathbf{y}$, we can also form the conditional distribution

$$p(\mathbf{x} \mid \mathbf{y}; \theta) \;\; = \;\; \frac{\exp\{\langle \theta,\, \boldsymbol{\phi}(\mathbf{y}, \mathbf{x})\rangle\}}{\int_{\mathcal{X}^n} \exp\{\langle \theta,\, \boldsymbol{\phi}(\mathbf{y}, \mathbf{x})\rangle\}\, \boldsymbol{\nu}(d\mathbf{x})} \;\; := \;\; \exp\{\langle \theta,\, \boldsymbol{\phi}(\mathbf{y}, \mathbf{x})\rangle \; - \; A_y(\theta)\}, \tag{61}$$

where the second equality defines the log partition function A_y associated with the conditional. Thus, we see the conditional is also an exponential family distribution, so that Theorem 2 provides the variational representation

$$A_y(\theta) \;\; = \;\; \sup_{\mu \in \mathcal{M}_y} \big\{\langle \theta,\, \mu\rangle - A_y^*(\mu)\big\}, \tag{62}$$

where the conjugate dual is also defined variationally:

$$A_y^*(\mu) \;\; := \;\; \sup_{\theta \in \Theta} \{\langle \mu,\, \theta\rangle - A_y(\theta)\}. \tag{63}$$

In equation (62), $\mathcal{M}_y$ is defined as the set of possible expectations of $\boldsymbol{\phi}(\mathbf{y}, \mathbf{x})$, where $\mathbf{x}$ is random and $\mathbf{y}$ is held fixed.

The *incomplete log likelihood* is the log probability of the observed data $\mathbf{y}$. From equation (60) and the definition of A_y, it is easy to verify that this log likelihood can be written as a difference of log partition functions:

$$\log p(\mathbf{y}; \theta) \;\; = \;\; A_y(\theta) - A(\theta). \tag{64}$$

From the variational representation (62) (i.e., from Fenchel's inequality), we obtain the lower bound $A_y(\theta) \geq \langle \mu,\, \theta\rangle - A_y^*(\mu)$, valid for any $\mu \in \mathcal{M}_y$, and hence a lower bound for the incomplete log likelihood:

$$\log p(\mathbf{y}; \theta) \;\; \geq \;\; \langle \mu,\, \theta\rangle - A_y^*(\mu) - A(\theta) \tag{65a}$$

$$:= \;\; \mathcal{L}(\mu, \theta), \tag{65b}$$

where the final line defines $\mathcal{L}(\mu, \theta)$.

With this set-up, the EM algorithm is coordinate ascent in $\mathcal{L}$:

$$\textbf{(E step)} \qquad \mu^{(t+1)} = \operatorname{argmax}_{\mu \in \mathcal{M}_y} \ \mathcal{L}(\mu, \theta^{(t)}) \tag{66a}$$

$$\textbf{(M step)} \qquad \theta^{(t+1)} = \operatorname{argmax}_{\theta \in \Theta} \ \mathcal{L}(\mu^{(t+1)}, \theta). \tag{66b}$$

To see the correspondence with the traditional presentation of the EM algorithm, note first that the maximization underlying the E step reduces to

$$\max_{\mu \in \mathcal{M}_y} \ \{\langle \mu, \, \theta^{(t)} \rangle\rangle - A_y^*(\mu)\}, \tag{67}$$

which by (62) is equal to $A_y(\theta^{(t)})$, with the maximizing argument equal to the mean parameter that is dually coupled with $\theta^{(t)}$. Thus the vector $\mu^{(t+1)}$ that is computed by maximization in the first argument of $\mathcal{L}(\mu, \theta)$ is exactly the expectation of the sufficient statistics given the current parameter value $\theta^{(t)}$, a computation that is traditionally referred to as the E step. Moreover, the maximization underlying the M step reduces to

$$\max_{\theta \in \Theta} \ \{\langle \mu^{(t+1)}, \, \theta \rangle - A(\theta)\}, \tag{68}$$

which is simply a maximum likelihood problem based on the expected sufficient statistics $\mu^{(t+1)}$—traditionally referred to as the M step.

Moreover, given that the value achieved by the E step on the right-hand-side of (67) is equal to $A_y(\theta^{(t)})$, the inequality in (65a) becomes an equality by (64). Thus, after the E step, the lower bound $\mathcal{L}(\mu, \theta^{(t)})$ is actually equal to the incomplete log likelihood $\log p(\mathbf{y}; \theta^{(t)})$, and the subsequent maximization of $\mathcal{L}$ with respect to θ in the M step is guaranteed to increase the log likelihood as well.

What if it is infeasible to compute the expected sufficient statistics? One possible response to this problem is to make use of a variational relaxation for the E step. In particular, we compute

$$\textbf{(Variational E step)} \qquad \mu^{(t+1)} = \operatorname{argmax}_{\mu \in \mathcal{M}_{tract}(G;H)} \ \mathcal{L}(\mu, \theta^{(t)}), \tag{69}$$

where $\mathcal{M}_{tract}(G; H)$ is the set of dual parameters associated with a tractable subgraph. The variational E step thus reduces to

$$\max_{\mu \in \mathcal{M}_{tract}(G;H)} \big\{\langle \mu, \, \theta^{(t)} \rangle - A_{x,H}^*(\mu)\big\}, \tag{70}$$

which is exactly the mean field approximation. The variational E step thus involves replacing expected sufficient statistics with the approximate expected sufficient statistics obtained by a mean field algorithm. 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 $\mathcal{L}$ and the incomplete log likelihood, it is no longer the case that the algorithm necessarily goes uphill in the latter quantity.

In the following sections of the paper, we present a number of variational relaxations that can be viewed as extensions of mean field methods. It is tempting, and common in practice, to substitute the approximate expected sufficient statistics 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 generally involve lower bounds on the log partition function. Consequently, the connection to EM is thus less clear than in the mean field case, and the algorithm is not guaranteed to maximize a lower bound.

6 Bethe entropy approximation and sum-product algorithm

In this section, we turn to another important message-passing algorithm for approximate inference, known either as *belief propagation*, or the *sum-product algorithm.* In Section 2.5.1, we described the use of the sum-product algorithm for trees, in which context it is guaranteed to converge and perform exact inference. When applied to graphs with cycles, there are no such guarantees, but it is nonetheless widely used to compute approximate marginals. In this section, we will describe the variational interpretation of the sum-product updates, first elucidated by Yedidia, Freeman and Weiss [122]. While mean field and sum-product are similar as message-passing algorithms, their respective variational interpretations are fundamentally different. In particular, whereas the essence of mean field is to *restrict* optimization to a limited class of distributions for which the negative entropy and mean parameters can be characterized *exactly*, the the sum-product algorithm, in contrast, is based on *enlarging* the constraint set and *approximating* the entropy function.

6.1 Basic ingredients

The standard Bethe approximation applies to an undirected graphical model with potential functions involving at most pairs of variables; we refer to any such model as a *pairwise Markov random field.* 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 C 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 7.

For the current section, let us assume that the given model is a pairwise Markov random field defined by a graph $G = (V, E)$. Although the Bethe approximation can be developed more generally, we also limit our discussion to multinomial random vectors, for which each x_s takes values in the space $\mathcal{X}_s = \{0, 1, \ldots, m_s - 1\}$. In this section, we will use the canonical overcomplete representation which, as previously described in equation (38), is based on the indicator functions $\{\mathbb{I}_j(x_s),\ j \in \mathcal{X}_s\}$ for $s \in V$ and $\{\mathbb{I}_{jk}(x_s, x_t),\ (j,k) \in \mathcal{X}_s \times \mathcal{X}_t\}$ for $(s,t) \in E$. In this representation, any pairwise Markov random field has the form:

$$p(\mathbf{x}; \theta) \quad \propto \quad \exp\big\{\sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t)\big\}, \tag{71}$$

where we have used the convenient shorthand

$$\theta_s(x_s) \;:=\; \sum_{j \in \mathcal{X}_s} \theta_{s;j} \mathbb{I}_j(x_s), \qquad \theta_{st}(x_s, x_t) \;:=\; \sum_{(j,k) \in \mathcal{X}_s \times \mathcal{X}_t} \theta_{st;jk} \mathbb{I}_{jk}(x_s, x_t). \tag{72}$$

The associated marginal functions $\mu_s(x_s)$ and $\mu_{st}(x_s, x_t)$ are defined analogously to $\theta_s(x_s)$ and $\theta_{st}(x_s, x_t)$, as in equation (40). Finally, we denote the marginal polytope associated with this exponential representation by $\mathrm{MARG}(G)$.

6.1.1 Bethe entropy approximation

As discussed at length in Section (40), the negative entropy A^*, as a function of only the mean parameters μ, typically lacks a closed form expression. We observed, moreover, that junction tree theorem provides an important class of exceptions to this general rule. As a special case, for tree-structured distributions, the function A^* has a closed-form expression that is straightforward to compute (see Section 4.2.3).

In the tree case, the negative entropy A^* decomposes into a sum of single node entropy and edgewise mutual information terms, defined as follows:

$$H_s(\mu_s) \;:=\; -\sum_{x_s} \mu_s(x_s) \log \mu_s(x_s), \tag{73a}$$

$$I_{st}(\mu_{st}) \;:=\; \sum_{x_s, x_t} \mu_{st}(x_s, x_t) \log \frac{\mu_{st}(x_s, x_t)}{\mu_s(x_s)\mu_t(x_t)} \;=\; H_s(\mu_s) + H_t(\mu_t) - H_{st}(\mu_{st}). \tag{73b}$$

Of course, the entropy of a distribution defined by a graph with cycles will not, in general, decompose additively like a tree. Nonetheless, one can imagine using the sum of local terms as an approximation to the entropy. Doing so yields the following *Bethe approximation* to the entropy on a graph with cycles:

$$H_{Bethe}(\mu) \;:=\; \sum_{s \in V} H_s(\mu_s) - \sum_{(s,t) \in E} I_{st}(\mu_{st}). \tag{74}$$

To be clear, $H_{Bethe}(\mu)$ is an approximation to $-A^*(\mu)$. From our development in Section 4.2.3, we know if that if the graph is tree-structured, then $H_{Bethe}(\mu) = -A^*(\mu)$, so that the approximation is exact.

Remark: An alternative form of the Bethe entropy approximation can be derived by using the relation between mutual information and entropy given in equation (73b). In particular, expanding the mutual information terms in this way, and then collecting all the single node entropy terms yields $H_{Bethe}(\mu) = \sum_{s \in V}(1 - d_s) H_s(\mu_s) + \sum_{(s,t) \in E} H_{st}(\mu_{st})$, where d_s denotes the number of neighbors of node s. This representation is the form of the Bethe entropy introduced by Yedidia et al. [122]; however, the form given in equation (74) turns out to be more convenient for our purposes.

6.1.2 Tree-based outer bound

Note that the Bethe entropy approximation H_{Bethe} is certainly well-defined for any $\mu \in \mathrm{MARG}(G)$. However, as discussed in Section 4.1.3, characterizing this polytope of realizable marginals is a very challenging problem. Accordingly, a natural approach is to specify a subset of necessary constraints, which leads to an outer bound on $\mathrm{MARG}(G)$.

Let $\tau_s(x_s)$ and $\tau_{st}(x_s, x_t)$ be a set of candidate marginal distributions. In Example 8, we considered the following constraint set:

$$\mathrm{LOCAL}(G) \;=\; \{\, \tau \geq 0 \mid \sum_{x_s} \tau_s(x_s) = 1, \; \sum_{x_s} \tau_{st}(x_s, x_t) = \tau_t(x_t) \,\}.$$

Although $\mathrm{LOCAL}(G)$ is an exact description of the marginal polytope for a tree-structured graph, it is only an outer bound for graphs with cycles. (See Example 14 for a vector $\tau \in \mathrm{LOCAL}(G)$ that does *not* belong to $\mathrm{MARG}(G)$). For this reason, our change in notation—i.e., from μ to τ—is quite deliberate, with the goal of emphasizing that members τ of $\mathrm{LOCAL}(G)$ need not be realizable. We refer to members of $\mathrm{LOCAL}(G)$ as *pseudomarginals.* Note that the Bethe entropy is also well-defined for any pseudomarginal in $\mathrm{LOCAL}(G)$.

Figure 15 provides an idealized illustration of the constraint set $\mathrm{LOCAL}(G)$, and its relation to the exact marginal polytope $\mathrm{MARG}(G)$. By construction, $\mathrm{LOCAL}(G)$ is another polytope that is a *convex outer approximation* to $\mathrm{MARG}(G)$. It is worthwhile contrasting with the *non-convex inner approximation* used by a mean field approximation, as illustrated in Figure 14.

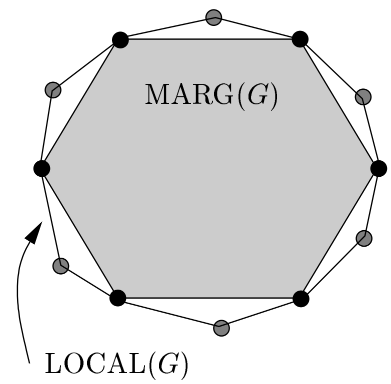


Figure 15. An idealized illustration of the tree-based constraint set LOCAL(G) as an outer bound on the marginal polytope MARG(G).

6.2 Bethe variational problem and sum-product

Combining the entropy approximation H_{Bethe} with the tree-based constraint set LOCAL(G) leads to the *Bethe variational problem*:

$$\max_{\tau \in \mathrm{LOCAL}(G)} \big\{ \langle \theta, \, \tau \rangle + \sum_{s \in V} H_s(\tau_s) - \sum_{(s,t) \in E} I_{st}(\tau_{st}) \big\}. \tag{75}$$

Although ostensibly similar to a (structured) mean field approach, the Bethe variational problem (BVP) is fundamentally different in a number of ways. First, as discussed in Section 5, a mean field method is based on an exact representation of the entropy, albeit over a limited class of distributions. In contrast, with the exception of tree-structured graphs, the Bethe entropy is a bona fide *approximation* to the entropy. For instance, it is not difficult to see that it can be negative, which of course can never happen for an exact entropy. Second, the mean field approach entails optimizing over an *inner bound* on the marginal polytope, which ensures that any mean field solution is always globally consistent with respect to at least one distribution, and that it yields a lower bound on the log partition function. In contrast, since LOCAL(G) is a strict outer bound on the set of realizable marginals MARG(G), the optimizing pseudomarginals τ^* of the BVP may not be globally consistent with any distribution.

6.2.1 Solving the Bethe variational problem

We now consider methods for solving the BVP. Observe that the set LOCAL(G) is a polytope defined by $\mathcal{O}(n + |E|)$ constraints. A natural approach to solving the BVP, then, is to attach Lagrange multipliers to these constraints, and find stationary points of the Lagrangian. The key insight of Yedidia et al. [122] is that the sum-product updates (7) are a particular technique for trying to find such Lagrangian stationary points.

Proposition 11 (Message-passing). *For each $x_s \in \mathcal{X}_s$, let $\lambda_{st}(x_s)$ be a Lagrange multiplier associated with the constraint $C_{ts}(x_s) = 0$, where $C_{ts}(x_s) := \tau_s(x_s) - \sum_{x_t} \tau_{st}(x_s, x_t)$. Consider the partial Lagrangian corresponding to the Bethe variational problem* (75)*:*

$$\mathcal{L}(\tau; \lambda) \;\; := \;\; \langle \theta, \, \tau \rangle + \sum_{s \in V} H_s(\tau_s) - \sum_{(s,t) \in E} I_{st}(\tau_{st}) + \sum_{(s,t) \in E} \big[\sum_{x_s} \lambda_{ts}(x_s) C_{ts}(x_s) + \sum_{x_t} \lambda_{st}(x_t) C_{st}(x_t) \big].$$

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

$$\nabla_\tau \mathcal{L}(\tau^*; \lambda^*) \;=\; 0, \qquad\qquad \nabla_\lambda \mathcal{L}(\tau^*; \lambda^*) \;=\; 0 \tag{76}$$

Proof. Note that this Lagrangian formulation is a partial one, because it assigns Lagrange multipliers to the constraints $C_{ts}(x_s) = 0$, and deals with the normalization and non-negativity constraints explicitly. Computing $\nabla_\tau \mathcal{L}(\tau; \lambda)$ and setting it to zero yields the relations:

$$\begin{aligned}
\log \tau_s(x_s) \;&=\; c + \theta_s(x_s) + \sum_{t \in \mathcal{N}(s)} \lambda_{ts}(x_s) && \text{(77a)}\\
\log \frac{\tau_{st}(x_s, x_t)}{\left[\sum_{x_s} \tau_{st}(x_s, x_t)\right]\left[\sum_{x_t} \tau_{st}(x_s, x_t)\right]} \;&=\; c' + \theta_{st}(x_s, x_t) - \lambda_{ts}(x_s) - \lambda_{st}(x_t). && \text{(77b)}
\end{aligned}$$

Here c, c' are constants that we are free to adjust in order to satisfy normalization conditions.[11]

The condition $\nabla_\lambda \mathcal{L}(\tau; \lambda)$ is equivalent to $C_{ts}(x_s) = 0$. Using this consistency condition and equation (77a), we can re-arrange equation (77b) to obtain:

$$\log \tau_{st}(x_s, x_t) \;=\; c + \theta_{st}(x_s, x_t) + \theta_s(x_s) + \theta_t(x_t) + \sum_{u \in \mathcal{N}(s)\backslash t} \lambda_{us}(x_s) + \sum_{u \in \mathcal{N}(t)\backslash s} \lambda_{ut}(x_t). \tag{78}$$

So as to make explicit the connection to the sum-product algorithm, we define messages in terms of the Lagrange multipliers via $M_{ts}(x_s) = \exp(\lambda_{ts}(x_s))$. With this notation, we can then write equivalent forms of equations (77a) and (78):

$$\begin{aligned}
\tau_s(x_s) \;&=\; \kappa \;\exp(\theta_s(x_s)) \prod_{t \in \mathcal{N}(s)} M_{ts}(x_s) && \text{(79a)}\\
\tau_{st}(x_s, x_t) \;&=\; \kappa' \;\exp\big(\theta_{st}(x_s, x_t) + \theta_s(x_s) + \theta_t(x_t)\big) \prod_{u \in \mathcal{N}(s)\backslash t} M_{us}(x_s) \prod_{u \in \mathcal{N}(t)\backslash s} M_{ut}(x_t) && \text{(79b)}
\end{aligned}$$

Here κ, κ' are positive constants chosen so that the pseudomarginals satisfy normalization conditions. Note that τ_s and τ_{st} so defined are clearly non-negative.

To conclude, we need to adjust the Lagrange multipliers or messages so that the constraint $\sum_{x_s} \tau_{st}(x_s, x_t) = \tau_s(x_s)$ is satisfied for every edge. Using equations (79a) and (79b) and performing some algebra, the end result is

$$M_{ts}(x_s) \;=\; \kappa \sum_{x_t} \exp\left\{\theta_{st}(x_s, x_t) + \theta_t(x_t)\right\} \prod_{u \in \mathcal{N}(t)\backslash s} M_{ut}(x_t), \tag{80}$$

which is equivalent to the familiar sum-product update (7). By construction, any fixed point M^* of these updates specifies a pair (τ^*, λ^*) that satisfies the stationary conditions (76). $\square$

Remarks: (a) An important consequence of Proposition 11 is to guarantee the existence of sum-product fixed points. Observe that the cost function in the Bethe variational problem (75) is continuous and bounded above, and the constraint set $\mathrm{LOCAL}(G)$ is non-empty and compact; therefore, at least some (possibly local) maximum is attained. Moreover, since the constraints are linear, there will always be a set of Lagrange multipliers associated with any local maximum [8]. For any optimum in the relative interior of $\mathrm{LOCAL}(G)$, these Lagrange multipliers can be used to construct a fixed point of the sum-product updates, as in the proof of Proposition 11.

[11] The value of arbitrary constants like c can change from line to line.

(b) For graphs with cycles, Proposition 11 provides no guarantees on the convergence of the sum-product updates; 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 (80) in parallel. Other more global schemes for message-passing are possible, and commonly used in certain applications like turbo-decoding [e.g., 75]. Tatikonda and Jordan [102] have shown that the convergence of parallel updates is related to the structure of Gibbs measures on the computation tree. Other researchers [e.g., 124, 115, 52] have proposed alternatives to sum-product that are guaranteed to converge, albeit at the price of increased computational cost.
(c) With the exception of trees and other special cases [82, 76], the BVP is usually a non-convex problem, in that H_{Bethe} fails to be concave. As a consequence, there may be multiple local optima to the BVP, and there are no guarantees that sum-product (or other iterative algorithms) will find a global optimum.

6.2.2 Nature of fixed points

This section explores an alternative characterization of sum-product fixed points [112], one which makes connections to the junction tree algorithm for exact inference described in Section 2.5.2. One view of the junction tree algorithm 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 (46). Indeed, the sum-product algorithm for trees can be understood as an efficient method for computing this alternative parameterization.

The following result [112] shows that the sum-product algorithm, when applied to a graph with cycles, can still be interpreted as performing a type of reparameterization:

Proposition 12. *Consider the Bethe variational problem and sum-product algorithm applied to the distribution $p(\mathbf{x};\theta)$. Then any fixed point $\tau^* = \{\tau_s^*, \tau_{st}^*\}$ of the sum-product algorithm, and more generally any local optimum of the Bethe variational problem in the relative interior of* $\text{LOCAL}(G)$, *specifies a reparameterization $p(\mathbf{x};\tau^*) \equiv p(\mathbf{x};\theta)$ of the original distribution of the following form:*

$$p(\mathbf{x};\tau^*) \quad := \quad \frac{1}{Z(\tau^*)}\prod_{s\in V}\tau_s^*(x_s) \prod_{(s,t)\in E}\frac{\tau_{st}^*(x_s,x_t)}{\tau_s^*(x_s)\tau_t^*(x_t)}. \tag{81}$$

Remark: Equation (81) is the analog of the tree-structured factorization (46), but as applied to a graph with cycles. By definition of the sum-product algorithm, the pseudomarginals $\{\tau_s^*, \tau_{st}^*\}$ are elements of $\text{LOCAL}(G)$, and hence locally consistent. However, they need not belong to the marginal polytope $\text{MARG}(G)$, and hence may fail global consistency (as illustrated in Example 14 to follow). Moreover, in contrast to the tree factorization (46), the normalization constant $Z(\tau^*)$ in equation (81) will not be unity in general.

Proof: By remark (a) following Proposition 11, any local optimum of the BVP can be associated with a sum-product fixed point. By definition of the pseudomarginals in equations (79a) and (79b), we have the equivalence $\tau_{st}^*(x_s,x_t)/[\tau_s^*(x_s)\tau_t^*(x_t)] \propto \exp\{\theta_{st}(x_s,x_t)\}\ [M_{st}^*(x_t)M_{ts}^*(x_s)]$. Using this relation, we rewrite the product on the RHS of equation (81) as follows:

$$\prod_{s\in V}\tau_s^*(x_s)\prod_{(s,t)\in E}\frac{\tau_{st}^*(x_s,x_t)}{\tau_s^*(x_s)\tau_t^*(x_t)} \quad = \quad \kappa\prod_{s\in V}\big[\exp\{\theta_s(x_s)\}\prod_{u\in\mathcal{N}(s)}M_{us}^*(x_s)\big]\prod_{(s,t)\in E}\frac{\exp(\theta_{st}(x_s,x_t))}{M_{st}^*(x_t)M_{ts}^*(x_s)}.$$

Consider a particular message M_{vw}^* associated with the edge in the direction v to w. Observe that it appears once in the numerator (in the term for node w in the product over vertices) and once in the denominator (in the term for edge (v, w) in the product over edges). Consequently, all the messages cancel out in the full product, thereby establishing that

$$\prod_{s \in V} \tau_s^*(x_s) \prod_{(s,t) \in E} \frac{\tau_{st}^*(x_s, x_t)}{\tau_s^*(x_s)\tau_t^*(x_t)} \quad \propto \quad \exp\big\{\sum_{s \in V} \theta_s(x_s) \; + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t)\big\},$$

which is the reparameterization claim. □

Proposition 12 provides some insight into the geometry of the Bethe variational problem (BVP) and the nature of its local optima [112]. It should be noted that the reparameterization statement (81) is possible only because the BVP is formulated in an overcomplete representation. A key consequence of this overcompleteness is that given distribution $p(\mathbf{x}; \theta)$ can be associated with an affine subset $\mathcal{C}(\theta)$ of exponential parameters, known as an e-flat manifold in information geometry [4], such that $p(\mathbf{x}; \gamma) = p(\mathbf{x}; \theta)$ for all $\gamma \in \mathcal{C}(\theta)$. Using this notion, equation (81) can be re-stated in the following way: for any local optimum τ^* of the BVP lying in the relative interior of $\mathrm{LOCAL}(G)$, the parameter γ^* with components defined as follows

$$\gamma_s^*(x_s) \quad := \quad \begin{cases} \log \tau_s^*(x_s) & \text{for } s \in V, \;\; x_s \in \mathcal{X}_s \\ \log[\tau_{st}^*(x_s, x_t)/\tau_s^*(x_s)\tau_t^*(x_t)] & \text{for } (s,t) \in E, \;\; (x_s, x_t) \in \mathcal{X}_s \times \mathcal{X}_t \end{cases}$$

is a member of the e-flat manifold $\mathcal{C}(\theta)$. The proof of Proposition 12 shows that the sum-product algorithm has a stronger property—namely, that all its iterates are confined to $\mathcal{C}(\theta)$. Although alternative algorithms [e.g., 124, 115, 52] for solving the BVP may evolve outside of this affine set, Proposition 12 shows that they must eventually converge to it.

This result can also be exploited to gain insight into the nature of the *approximation error*: that is, the difference between the exact marginals μ_s of $p(\mathbf{x}; \theta)$ and the approximations τ_s^* computed by the sum-product algorithm. Given any spanning tree $T = (V, E(T))$ contained within G, let $\tau^*(T)$ denote the pseudomarginals associated with nodes and edges in T. This reduced set of pseudomarginals defines a tree-structured distribution as follows:

$$p(\mathbf{x}; \tau^*(T)) \quad := \quad \prod_{s \in V} \tau_s^*(x_s) \prod_{(s,t) \in E(T)} \frac{\tau_{st}^*(x_s, x_t)}{\tau_s^*(x_s) \; \tau_t^*(x_t)}. \tag{82}$$

By the local consistency guaranteed by membership in $\mathrm{LOCAL}(G)$ and the junction tree theorem (see Proposition 1), we are guaranteed that $\tau^*(T)$ are the *exact marginals* for the tree-structured distribution $p(\mathbf{x}; \tau^*(T))$. Consequently, each tree of the graph can be used to assess the error in the BVP approximation. In particular, the difference between τ_s^* and μ_s stems from the perturbation of removing from the original distribution $p(\mathbf{x}; \theta) \equiv p(\mathbf{x}; \tau^*)$ a set of *reparameterized* compatibility functions so as to obtain the tree-structured distribution $p(\mathbf{x}; \tau^*(T))$. On this basis, 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. [112].

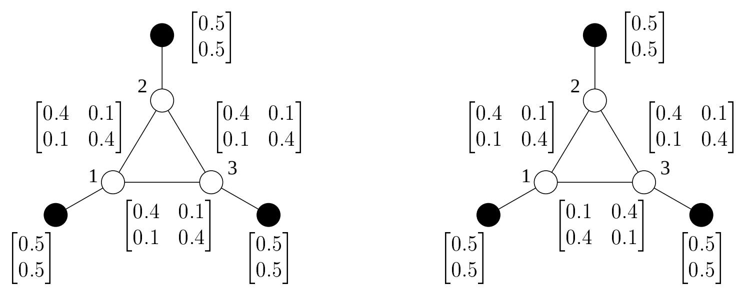
As illustrated in Figure 15, the constraint set $\mathrm{LOCAL}(G)$ of the Bethe variational problem is a strict outer bound on $\mathrm{MARG}(G)$, in which the exact marginals of $p(\mathbf{x}; \theta)$ must lie. A natural question, then, is whether solutions to the Bethe variational problem ever fall into the region $\mathrm{LOCAL}(G) \backslash \mathrm{MARG}(G)$. Proposition 12 provides a straightforward answer to this question: it enables us to specify, for *any* pseudomarginal τ in the relative interior of $\mathrm{LOCAL}(G)$, a distribution

$p(\mathbf{x};\theta)$ for which τ is a fixed point of the sum-product algorithm. The following example illustrates this construction.

Example 14 (Globally inconsistent fixed point). We illustrate using a binary random vector on the simplest possible graph for which sum-product is not exact—namely, a single cycle with three nodes. Consider candidate marginal distributions $\{\tau_s, \tau_{st}\}$ of the form illustrated in Figure 16(a), where $\beta_{st} \in [0, 0.5]$ is a parameter to be specified independently for each edge (s,t). It

$$\tau_s \quad := \quad \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}$$

$$\tau_{st} \quad := \quad \begin{bmatrix} \beta_{st} & 0.5 - \beta_{st} \\ 0.5 - \beta_{st} & \beta_{st} \end{bmatrix}$$



(a) (b) (c)

Figure 16. Illustration of the marginal polytope for a single cycle graph on three nodes. (a) Form of single node and joint pairwise marginals. The parameter β_{st} takes values in $[0, 0.5]$. (b) Setting $\beta_{st} = 0.4$ for all three edges gives a globally consistent set of marginals. (c) With β_{13} perturbed to 0.1, the marginals (though locally consistent) are no longer globally so.

is straightforward to verify that $\{\tau_s, \tau_{st}\}$ belong to $\mathrm{LOCAL}(G)$ for any choice of $\beta_{st} \in [0, 0.5]$.

First, consider the setting $\beta_{st} = 0.4$ for all edges (s,t), as illustrated in panel (b). It is not difficult to show that the resulting marginals thus defined are realizable; in fact, they can be obtained from the distribution that places probability 0.35 on each of the configurations $[0\ 0\ 0]$ and $[1\ 1\ 1]$, and probability 0.05 on each of the remaining six configurations. Now suppose that we perturb one of the pairwise marginals—say τ_{13}—by setting $\beta_{13} = 0.1$. The resulting problem is illustrated in panel (c). Observe that there are now strong (positive) dependencies between the pairs of variables (x_1, x_2) and (x_2, x_3): both pairs are quite likely to agree (with probability 0.8). In contrast, the pair (x_1, x_3) can only share the same value relatively infrequently (with probability 0.2). This arrangement should provoke some doubt. Indeed, it can be shown that $\tau \notin \mathrm{MARG}(G)$ by attempting but *failing* to construct a distribution that realizes τ. (See Example 24 of Section 9 for a quick proof using semidefinite constraints.)

We now wish to construct a problem $p(\mathbf{x};\theta)$ for which the pseudomarginals τ are a fixed point of the sum-product algorithm. Proposition 12 enables us to do so easily. In particular, suppose that we define $\theta_s(x_s) = \log \tau_s(x_s)$ and $\theta_{st}(x_s, x_t) = \log \tau_{st}(x_s, x_t)/[\tau_s(x_s)\tau_t(x_t)]$. Now consider the sum-product algorithm updates of equation (80) with the messages M_{st} initialized to all ones. With these uniform messages and $\theta = \{\theta_s, \theta_{st}\}$ defined as above, we have:

$$\kappa \sum_{x_t} \exp\left\{\theta_{st}(x_s, x_t) + \theta_t(x_t)\right\} \prod_{u \in \mathcal{N}(t)\setminus s} M_{ut}(x_t) \quad = \quad \kappa \sum_{x_t} \frac{\tau_{st}(x_s, x_t)}{\tau_s(x_s)} \;=\; 1$$

Thus, the vector $\tau \in \mathrm{LOCAL}(G) \setminus \mathrm{MARG}(G)$ is a fixed point of sum-product as applied to the constructed $p(\mathbf{x};\theta)$. $\diamondsuit$

More generally, this construction applies to an arbitrary member of $\mathrm{ri}\,\mathrm{LOCAL}(G)$. Accordingly, we conclude that the sum-product algorithm induces a mapping from the space of exponential parameters $\Theta = \mathbb{R}^d$ that is *onto* the relative interior of LOCAL(G). In contrast to the mean parameter mapping of Theorem 1, this mapping is, in general, multi-valued since the sum-product algorithm may have multiple fixed points.

7 Hypertree-based approximations and generalized sum-product

From our development in the previous section, it is clear that there are two *distinct* components to the Bethe variational principle: (a) the entropy approximation H_{Bethe}, and (b) the approximation LOCAL(G) to the set of realizable marginal parameters. In principle, the BVP could be strengthened by improving either one, or both, of these components.

In this section, we discuss a natural generalization of the BVP, first proposed by Yedidia et al. [123] and further explored by various researchers [e.g., 82, 76, 52, 108, 124], that improves both components simultaneously. 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.

7.1 Hypergraphs

A hypergraph $G = (V, E)$ is a natural generalization of a graph; in particular, it consists of a vertex set $V = \{1, \ldots, n\}$, and a set of hyperedges E, where each *hyperedge* h is a particular subset of V (i.e., an element of the power set of V). The set of hyperedges can be viewed naturally as a partially-ordered set [101], where the partial ordering is specified by inclusion. Given two hyperedges g and h, one of three possibilities can hold: (a) the hyperedge g is contained within h, in which case we write $g < h$; (b) if h is contained within g, then we write $h < g$; and (c) finally, if neither containment relation holds, then g and h are incomparable. 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 (i.e., an ordinary edge of the graph). Note the minor inconsistency in our definition of the hypertree edge set E; for hypergraphs (unlike graphs), the set of hyperedges can include (a subset of the) individual vertices.

7.1.1 Poset diagrams

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* [101]. Such poset representations have been used in previous work on generalized sum-product [82, 76], whereas Yedidia et al. [122] make use of closely related structures known as region graphs. Figure 17 provides some simple graphical illustrations of hypergraphs. Any ordinary graph, as a special case of a hypergraph, can be drawn as a hypergraph; in particular, panel (a) shows the hypergraph representation of a single cycle on four nodes. Panel (b) shows a hypergraph that is not equivalent to an ordinary graph, consisting of two hyperedges of size three joined by their intersection of size two. Shown in panel (c) is a more complex hypergraph, to which we will return in the sequel.

Given any hyperedge h, we define the sets of its *descendants* and *ancestors* in the following way:

$$\mathcal{D}(h) := \{g \in E \mid g < h\}, \qquad \mathcal{A}(h) := \{g \in E \mid g > h\}. \tag{83}$$

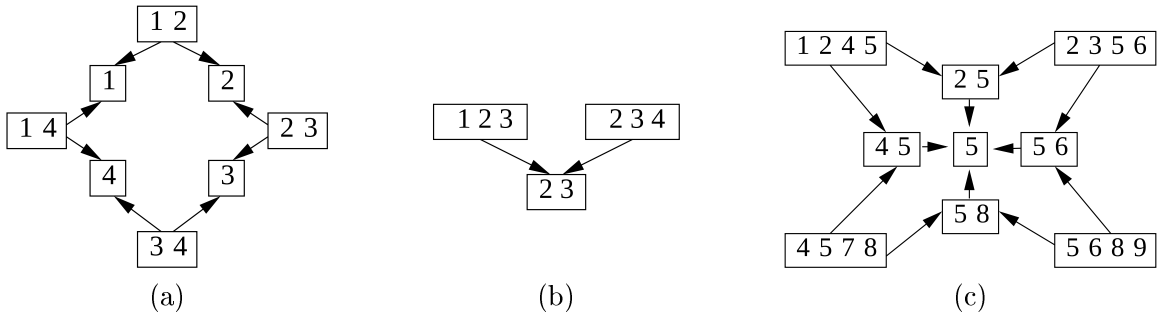


Figure 17. Graphical representations of hypergraphs. Subsets of nodes corresponding to hyperedges are shown in rectangles, whereas the arrows represent inclusion relations among hyperedges. (a) An ordinary single cycle graph represented as a hypergraph. (b) A simple hypergraph. (c) A more complex hypergraph.

For example, given the hyperedge $h = (1245)$ in the hypergraph in Figure 17(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.

7.1.2 Hypertrees

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 a singly-connected acyclic hypergraph of width k. Thus, for example, a spanning tree of an ordinary graph is a 1-hypertree, because its maximal hyperedges (i.e., ordinary edges) all have size two. As a second example, consider the hypergraph shown in Figure 18(a). It is clear that this hypergraph is equivalent to the junction tree with maximal cliques $\{(1245), (4578), (2356)\}$ and separator sets $\{(25), (45)\}$. Therefore, it is a hypertree with width three, since the maximal hyperedges have size four.

It should be noted that there is *not* a one-to-one correspondence between poset diagrams without cycles, and acyclic hypergraphs. In particular, a poset diagram may have cycles, but nonetheless correspond to a hypertree. This possibility is exemplified by Figure 18(b), which shows a set of hyperedges for which the poset diagram involves cycles. Nonetheless, it can be seen that this

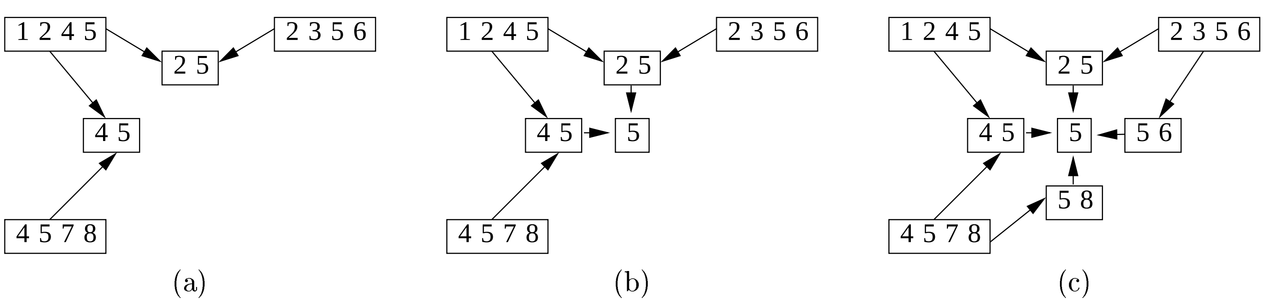


Figure 18. Three different graphical representations of the same underlying hypertree. (a) This diagram clearly corresponds to an acyclic hypergraph. (b) This representation seems different, but in fact corresponds to the same hypertree. (c) Another representation of the same hypertree. Hence hypertrees cannot be identified simply by the absence (or presence) of cycles in poset diagrams.

hypergraph is acyclic. A similar statement holds for the hypergraph in (c), which has even more cycles in its poset diagram. In fact, the junction tree corresponding to the hypertree in Figure 18(a) also constitutes a junction tree for the hypergraphs in Figures 18(b) and (c).

7.2 Hypertree factorization and entropy

In this section, we develop an alternative form of the junction tree factorization (9), and show how it leads to a local decomposition of the entropy. Associated with any poset is a Möbius function $\omega : E \times E \to \mathbb{Z}$; see Stanley [101] and Appendix D for more details. We use the Möbius function to define a bijection between the collection of marginals $\mu := \{\mu_h\}$ associated with the hyperedges of a hypergraph, and a new set of functions $\varphi := \{\varphi_h\}$, as follows:

$$\log \mu_h(x_h) = \sum_{g \in \mathcal{D}^+(h)} \log \varphi_g(x_g), \qquad \log \varphi_h(x_h) = \sum_{g \in \mathcal{D}^+(h)} \omega(g,h) \log \mu_g(x_g). \tag{84}$$

For a hypertree with an edge set containing all intersections between maximal hyperedges, the underlying distribution is guaranteed to factorize as follows:

$$p(\mathbf{x}) = \prod_{h \in E} \varphi_h(x_h). \tag{85}$$

Equation (85) is an alternative formulation of the well-known junction tree decomposition (9), for which some examples provide intuition.

Example 15. (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. For any vertex s, we have $\varphi_s(x_s) = \mu_s(x_s)$, whereas for any edge (s,t) we have $\varphi_{st}(x_s, x_t) = \mu_{st}(x_s, x_t)/[\mu_s(x_s)\,\mu_t(x_t)]$. In this special case, equation (85) reduces to the tree factorization in equation (46).

(b) Now consider the acyclic hypergraph specified by $\{(1245), (2356), (4578), (25), (45), (56), (58), (5)\}$, as illustrated in Figure 18(c). Omitting explicit dependence on $\mathbf{x}$ for notational simplicity, we first calculate $\varphi_{1245} = \frac{\mu_{1245}}{\varphi_{25}\varphi_{45}\varphi_5} = \frac{\mu_{1245}}{[\mu_{25}/\mu_5][\mu_{45}/\mu_5]\mu_5}$, with analogous expressions for φ_{2356} and φ_{4578}. We also have $\varphi_{25} = \mu_{25}/\mu_5$, with analogous expressions for the other pairwise terms. Putting the pieces together yields

$$p = \frac{\mu_{1245}}{\frac{\mu_{25}}{\mu_5}\frac{\mu_{45}}{\mu_5}\mu_5} \; \frac{\mu_{2356}}{\frac{\mu_{25}}{\mu_5}\frac{\mu_{56}}{\mu_5}\mu_5} \; \frac{\mu_{4578}}{\frac{\mu_{45}}{\mu_5}\frac{\mu_{58}}{\mu_5}\mu_5} \; \frac{\mu_{25}}{\mu_5}\,\frac{\mu_{45}}{\mu_5}\,\frac{\mu_{56}}{\mu_5}\,\frac{\mu_{58}}{\mu_5}\;\mu_5 = \frac{\mu_{1245}\,\mu_{2356}\,\mu_{4578}}{\mu_{25}\,\mu_{45}},$$

which agrees with the expression from the junction tree formula (9). $\diamondsuit$

An immediate but important consequence of the factorization (85) is a local decomposition of the entropy.

Proposition 13 (Hypertree entropy). *The entropy of a hypertree-structured distribution decomposes as*

$$H_{hyper}(\mu) \overset{(a)}{=} -\sum_{h \in E} I_h(\mu_h) \overset{(b)}{=} \sum_{h \in E} c(h) H_h(\mu_h), \tag{86}$$

where $I_h(\mu_h) := \sum_{\mathbf{x}} \mu_h(x_h) \log \varphi_h(x_h)$ is a multi-information, $H_h(\mu_h) := -\sum_{\mathbf{x}} \mu_h(x_h) \log \mu_h(x_h)$ is the entropy associated with hyperedge $h \in E$, and $c(f) := \sum_{e \in \mathcal{A}^+(f)} \omega(f,e)$ are overcounting numbers.

Proof. Equality (a) follows immediately from the hypertree factorization (85) and the definition of I_h. Equality (b) follows by applying the Möbius inversion relation (84) between $\log \varphi_h(\mathbf{x})$ and $\log \mu_h(x_h)$, expanding, and simplifying. $\square$

We illustrate by continuing with Example 15:

Example 16. (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 (86) is equivalent to the tree entropy given in equation (47). The overcounting numbers for a tree are $c((s,t)) = 1$ for any edge (s,t), and $c(s) = 1 - \deg(s)$ for any vertex s, where $\deg(s)$ denotes its degree.

(b) Consider again the hypertree in Figure 18(c). On the basis of our previous calculations in Example 15(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 $H_{hyper} = H_{1245} + H_{2356} + H_{4578} - H_{25} - H_{45}$. $\diamondsuit$

7.3 Augmented hypergraphs

Recall that the core of the Bethe approach of Section 6 consists of a particular tree-based (Bethe) approximation to entropy, and a tree-based outer bound on the marginal polytope. Our ultimate goal is to extend these tree-based approximations to ones based on (more general) hypertrees.[12] In this section, we take a step towards this goal by describing how to construct, on the basis of the original graph, an *augmented hypergraph* that serves as the basis for defining these approximations.

Our starting point is a Markov random field (MRF) defined by some (non-acyclic) hypergraph $G' = (V, E')$, meaning that $p(\cdot)$ has a factorization of the form:

$$p(\mathbf{x}) \quad \propto \quad \exp\big\{\sum_{h \in E'} \theta_h(x_h)\big\}. \tag{87}$$

Note that this equation reduces to our earlier representation (71) of a pairwise MRF when the hypergraph is an ordinary graph.

One strategy is to develop techniques for approximate inference based directly on the structure of G'. The Bethe approximation is of this form, corresponding to the case when G' is an ordinary graph. Rather than basing approximations on the structure of G', it can be beneficial to build them based on an *augmented hypergraph* $G = (V, E)$. A natural way in which to construct such augmented hypergraphs is by clustering nodes so as to define new hyperedges; different techniques of this nature are discussed in the papers [e.g., 123, 82, 76].

For the purposes of this discussion, we focus on a subclass of augmented hypergraphs. In particular, we require that the original hypergraph G' is *covered* by the augmented hypergraph, meaning that the hyperedge set E of the augmented hypergraph includes all hyperedges in E' (as well as the vertices of G'). A desirable feature of this requirement is that any Markov random field 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 17 (Covering hypergraph). To illustrate, suppose that the original hypergraph G' is simply an ordinary graph—namely, the 3×3 grid shown in Figure 19(a). As illustrated in panel (b),

[12] An ordinary tree is simply a hypertree of width one.

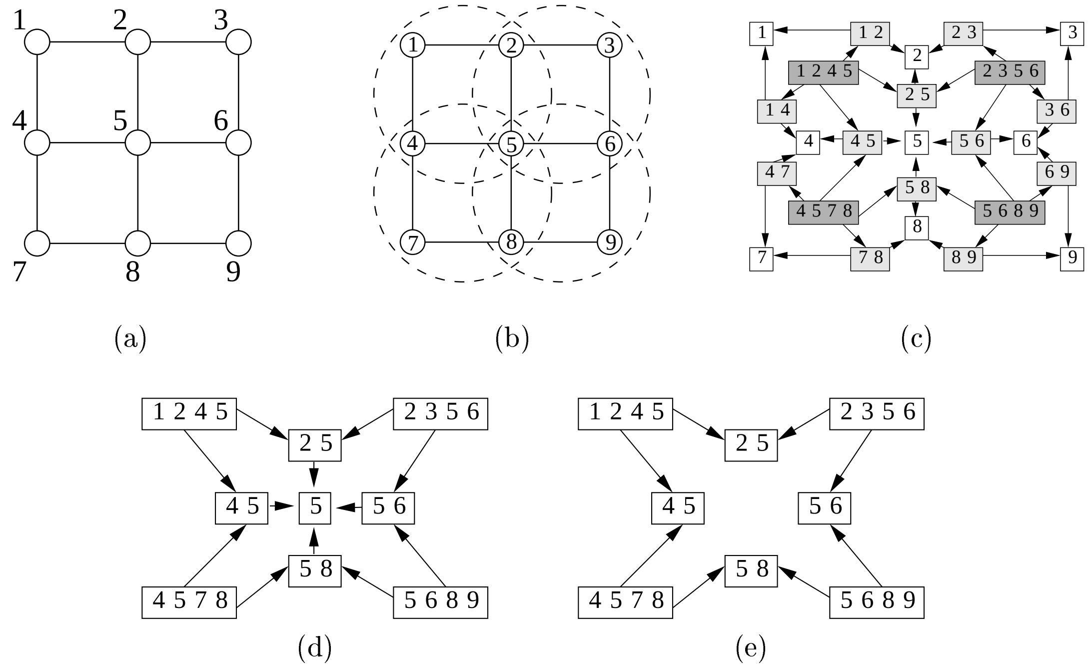


Figure 19. Constructing new hypergraphs via clustering and the single counting criterion. (a) Original (hyper)graph G' is a 3×3 grid. Its hyperedge set E' consists of the union of the vertex set with the (ordinary) edge set. (b) Nodes are clustered into groups of four. (c) A covering hypergraph G formed 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).

we cluster the nodes into groups of four, which is known as Kikuchi 4-plaque clustering in statistical physics [123]. We then form the augmented hypergraph G shown in panel (c), with hyperedge set $E := E' \cup \{(1245), (2356), (4578), (5689)\}$. 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. $\diamondsuit$

As emphasized by Yedidia et al. [123], 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 $\mathcal{C}(h') := \{f \in E \mid f \geq h'\}$ of hyperedges in E that contain h'. For ease of reference, we restate the definition of the overcounting numbers $c(\cdot)$ associated with the hypergraph G, originally defined in Proposition 13. In particular, these overcounting numbers are defined in terms of the Möbius function associated with G, viewed as a poset, in the following way:

$$c(f) \;\; := \;\; \sum_{e \in \mathcal{A}^+(f)} \omega(f, e). \tag{88}$$

The *single counting criterion* requires that for all $h' \in E'$ (including all single vertices), there holds

$$\sum_{f \in \mathcal{C}(h')} c(f) \;\; = \;\; 1. \tag{89}$$

Example 18 (Single counting). To illustrate the single counting criterion, we consider two additional hypergraphs that can be constructed from the 3×3 grid of Figure 19(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. [123]. 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 $\mathcal{C}(5) = E$ and $\sum_{h \in \mathcal{C}(5)} c(h) = 0$. 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 (14) which appears in (c) but not in (d). Since it has only one parent (which is a maximal hyperedge), we have $c(14) = 0$. In a similar fashion, we see that $c(12) = 0$. These two equalities together imply that $c(1) = 0$, so that we can eliminate hyperedges (12), (14) and (1) from hypergraph (c). By applying a similar argument to the remaining hyperedges, we can fully reduce hypergraph (c) to hypergraph (d). $\diamondsuit$

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 (89) should hold for the vertices. The proof is quite straightforward: we begin by observing that under the covering condition, the set $\mathcal{C}(h)$ is equal to $\mathcal{A}^+(h)$ in the augmented hypergraph G. We then invoke the following result:

Lemma 1 (Single counting). *For any $h \in E$, the associated overcounting 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:

$$\sum_{h \in \mathcal{A}^+(g)} c(h) \quad = \quad \sum_{h \in \mathcal{A}^+(g)} \; \sum_{f \in \mathcal{A}^+(h)} \omega(h, f). \tag{90}$$

Considering the double sum on the RHS, we see that for a fixed $d \in \mathcal{A}^+(g)$, there is a term $\omega(d, e)$ for each e such that $g \leq e \leq d$. Using this observation, we can write

$$\sum_{h \in \mathcal{A}^+(g)} \; \sum_{f \in \mathcal{A}^+(h)} \omega(h, f) \;=\; \sum_{d \in \mathcal{A}^+(g)} \; \sum_{\{e \,|\, g \leq e \leq d\}} \omega(e, d) \;\stackrel{(a)}{=}\; \sum_{d \in \mathcal{A}^+(g)} \delta(d, g) \;\stackrel{(b)}{=}\; 1.$$

Here equality (a) follows from the definition of the Möbius function (see Appendix D), and $\delta(d, g)$ is the Kronecker delta function, from which equality (b) follows. $\square$

Thus, the construction that we have described, in which the hyperedges (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 exists a variety of other valid constructions [e.g., 123, 82, 76]. All subsequent analysis will assume that the augmented hypergraph satisfies the single counting criterion.

7.4 Hypertree-based approximations

We now have the necessary ingredients to specify hypertree-based approximations to the exact variational principle of Theorem 2.

7.4.1 Approximate variational principle

Let G be a given augmented hypergraph, and let $t+1$ be the size of the largest hyperedge in G. Moreover, let $\tau = \{\tau_h\}$ be a collection of local marginals associated with the hyperedges $h \in E$. Of course, any such quantity must satisfy the obvious normalization condition $\sum_{x'_h} \tau_h(x'_h) = 1$. Similarly, these local marginals must be consistent with one another whereover they overlap; more precisely, for any pair of hyperedges $g < h$, the following *marginalization* condition must hold:

$$\sum_{\{x'_h \,\mid\, x'_g = x_g\}} \tau_h(x'_h) \quad = \quad \tau_g(x_g).$$

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

$$\mathrm{LOCAL}_t(G) \quad = \quad \Big\{ \tau \geq 0 \mid \sum_{x'_h} \tau_h(x'_h) = 1 \;\; \forall\, h \in E, \qquad \sum_{\{x'_h \,\mid\, x'_g = x_g\}} \tau_h(x'_h) = \tau_g(x_g) \;\; \forall g < h \Big\}. \tag{91}$$

Note that this constraint set is a natural generalization of the tree-based constraint set defined in equation (41). In particular, definition (91) coincides with definition (41) when the hypergraph G is simply an ordinary graph. As before, we refer to members $\mathrm{LOCAL}_t(G)$ as *pseudomarginals*. By the junction tree conditions in Proposition 1, the local constraints defining $\mathrm{LOCAL}_t(G)$ are sufficient to guarantee global consistency whenever G is a hypertree.

In analogy to the Bethe entropy approximation, Proposition 13 motivates the following hypertree-based approximation to the entropy:

$$H_{app}(\tau) \quad = \quad \sum_{g \in E} c(g) H_g(\tau_g). \tag{92}$$

Here $c(g) = \sum_{f \in \mathcal{A}^+(g)} \omega(g, f)$ are the overcounting numbers defined in equation (88). This entropy approximation and the outer bound $\mathrm{LOCAL}_t(G)$ on the marginal polytope, in conjunction, lead to the following hypertree-based approximation to the exact variational principle:

$$\max_{\tau \in \mathrm{LOCAL}_t(G)} \Big\{ \langle \theta, \tau \rangle + H_{app}(\tau) \Big\}. \tag{93}$$

This problem is the hypertree-based generalization of the Bethe variational problem (75).

Example 19 (Kikuchi method). To illustrate the approximate variational principle (93), consider the augmented hypergraph in Figure 19(d). To determine the form of the entropy approximation H_{app}, we first calculate the overcounting numbers $c(\cdot)$. 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, Lemma 1 yields that $c(g) = -1$ in this case. Applying Lemma 1 once more yields that $c(5) = 1$. Overall, the entropy approximation takes the form

$$H_{app} \quad = \quad \big[H_{1245} + H_{2356} + H_{4578} + H_{5689}\big] - \big[H_{25} + H_{45} + H_{56} + H_{58}\big] + H_5. \tag{94}$$

$\diamondsuit$

7.4.2 Generalized sum-product

In principle, the variational problem (93) could be solved by a number of methods. Here we describe an algorithm, referred to as *parent-to-child message-passing* by Yedidia et al. [123], 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.

Our first 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 (87). 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:

$$\psi_h(x_h) \quad := \quad \psi'_h(x_h) \prod_{g \in \mathcal{S}(h)} \psi'_g(x_g), \tag{95}$$

where $\mathcal{S}(h) := \{g \in E' \backslash E \ \mid g < h\}$ is the set of hyperedges in $E' \backslash E$ that are subsets of h. To illustrate this definition, consider the Kikuchi construction of Figure 19(d), which is an augmented hypergraph for the 3×3 grid in Figure 19(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 (95) yields $\psi_{1245} = \psi'_1\psi'_{12}\psi'_{14}$. More generally, using the definition (95), 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.

In the hypertree-based variational problem (93), 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 re-formulations of the original problem [e.g., 122, 76], which lead to different message-passing algorithms. In the case of parent-to-child form of message-passing derived by Yedidia et al. [123], the pseudomarginal τ_h takes the following form:

$$\tau_h(x_h) \quad = \quad \kappa \left[\prod_{g \in \mathcal{D}^+(h)} \psi_g(x_g) \right] \left[\prod_{g \in \mathcal{D}^+(h)} \; \prod_{f \in \mathrm{Par}(g) \backslash \mathcal{D}^+(h)} M_{f \to g}(x_g) \right]. \tag{96}$$

In this equation, the pseudomarginal τ_h includes a compatibility function ψ_g for each hyperedge g in $\mathcal{D}^+(h) := \mathcal{D}(h) \cup h$. 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 with an example:

Example 20 (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 19(d). Focusing first on the hyperedge (1245), the first term in equation (96) 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 use the definition (95) to determine the equivalent expression $\psi'_{12}\psi'_{14}\psi'_{25}\psi'_{45}\psi'_1\psi'_2\psi'_4\psi'_5$, now in terms of compatibility functions from the original hypergraph.

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 20(a) provides an illustration; the set $\mathcal{D}^+\{(1245)\}$ is given by the hyperedges within the dotted ellipses. In this case, the set

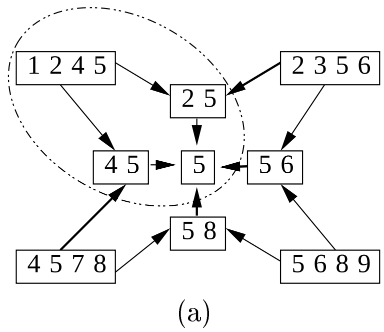


Figure 20. Illustration of relevant regions for parent-to-child message-passing in a Kikuchi approximation. (a) Message-passing for hyperedge (1245). Set of descendants $\mathcal{D}^+\{(1245)\}$ is shown within a dotted ellipse. Relevant parents for τ_{1245} consists of the set $\{(2356), (4578), (56), (58)\}$. (b) Message-passing for hyperedge (45). Dotted ellipse shows descendant set $\mathcal{D}^+\{(45)\}$. In this case, relevant parent hyperedges are $\{(1245), (4578), (25), (56), (58)\}$.

$\cup_g \operatorname{Par}(g) \backslash \mathcal{D}^+(h)$ 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:

$$\tau_{1245} \;=\; \kappa\, \psi'_{12}\, \psi'_{14}\, \psi'_{25}\, \psi'_{45}\;\; \psi'_1\, \psi'_2\, \psi'_4\, \psi'_5\;\; M_{(2356)\to(25)} M_{(4578)\to(45)} M_{(56)\to 5} M_{(58)\to 5}.$$

By symmetry, the expressions for the pseudomarginals on the other 4-hyperedges are analogous. By similar arguments, it is straightforward to compute the following expression for τ_{45} and τ_5:

$$\begin{aligned} \tau_{45} &= \kappa\, \psi'_{45}\, \psi'_4\, \psi'_5\;\; M_{(1245)\to(45)} M_{(4578)\to(45)}\; M_{(25)\to 5}\; M_{(56)\to 5}\; M_{(58)\to 5} \\ \tau_5 &= \kappa\, \psi'_5\, M_{(45)\to 5}\; M_{(25)\to 5}\; M_{(56)\to 5}\; M_{(58)\to 5}. \end{aligned}$$

$\diamondsuit$

Generalized forms of the sum-product updates follow by updating the messages so as to enforce the marginalization constraints defining membership in LOCAL(G); as in the proof of Proposition 11, 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 [123, 82, 76]. Moreover, in analogy to our earlier analysis of the ordinary sum-product algorithm, Proposition 12 can be suitably generalized: any fixed point of such generalized sum-product message-passing updates defines a hypertree-consistent reparameterization of the original distribution [112]. Furthermore, as with the Bethe approximation and any sum-product solution, the error (i.e., difference between the true and approximate marginals) stems from the *reparameterized set* of potentials that must be removed from the full hypergraph G so as to obtain a hypertree-structured distribution [112].

We conclude this section by noting that other approximations to the entropy, in addition to those based on hypertrees given here, are possible. For instance, the general region graph method of Yedidia et al. [123] includes entropy approximations that need not follow from hypertrees on the original vertex set. Moreover, Minka [77] proposed the expectation-propagation updates, which can be understood as a sequential technique for solving approximations to the variational principle based on other structured choices of entropy approximation.

8 Upper bounds via convex relaxations

Up to this point, we have considered two broad classes of variational methods: mean field methods (Section 5) and Bethe/Kikuchi approaches (Sections 6 and 7). Mean field methods provide not only approximate mean parameters but also lower bounds on the log partition function. In contrast, the Bethe/Kikuchi approaches lead to *neither* upper or lower bounds on this important quantity. Bounds on the log partition function are useful in a variety of contexts, including parameter estimation and large deviations bounds. A feature common to both mean field and Bethe/Kikuchi methods is that the underlying variational methods are usually not convex. As we have discussed, this lack of convexity can lead to local minima, and may cause substantial algorithmic difficulties.

The motivation of this section, then, is to describe variational principles that are both convex, and also provide upper bounds on the log partition function. The basic strategy is straightforward: so as to obtain upper bounds, we *relax* the variational representation of A in equation (32) by modifying it in the following two ways:

(a) by using a convex *outer bound* on the marginal polytope $\mathrm{MARG}(G)$.

(b) by replacing the negative dual function $-A^*$ with a concave *upper bound*.

From the variational principle of Theorem 2, it is clear that the solution of the modified variational problem so obtained will yield an upper bound on the log partition function. In addition, requiring a concave upper bound ensures that the modified variational problem has a unique (global) optimum.

The convex relaxation procedure summarized by steps (a) and (b) can be applied quite generally to obtain upper bounds on A in arbitrary exponential families. In this section, we illustrate this procedure by developing a class of convex relaxations for multinomial problems that are closely related to the Bethe/Kikuchi approximations discussed in Sections 6 and 7. Further details on the results described in this section can be found in the papers [108, 109, 113].

8.1 Combinations of trees

As in Section 6, let us consider again the case of a pairwise Markov random field, and use the standard overcomplete representation based on indicator functions at single nodes and edges. Suppose that $\mu = \{\mu_s, \mu_{st}\} \in \mathrm{MARG}(G)$ is a valid set of single node and joint pairwise marginals. We begin by describing how to upper bound the entropy $-A^*(\mu) = H(p(\mathbf{x}; \theta(\mu))$ using tree-structured distributions.

Given an arbitrary spanning tree $T = (V, E(T))$ of the graph, we let $\mu(T)$ represent the set of all mean parameters associated with vertices $s \in V$, and edges $(s,t) \in E(T)$. The vector $\mu(T)$ defines a tree-structured distribution of the following form:

$$p(\mathbf{x}; \mu(T)) \quad = \quad \prod_{s \in V} \mu_s(x_s) \prod_{(s,t) \in E(T)} \frac{\mu_{st}(x_s, x_t)}{\mu_s(x_s)\mu_t(x_t)}.$$

By construction, the value of the dual function $A^*(\mu(T))$ is simply the negative entropy of $p(\mathbf{x}; \mu(T))$.

We now claim that the tree-structured entropy $-A^*(\mu(T))$ provides an upper bound on the original entropy $-A^*(\mu)$. The intuition is based on the maximum entropy characterization of graphical models. In particular, the presence of an edge in a graph-structured distribution corresponds to some constraint that is active in the associated maximum entropy problem. Removing edges, then, corresponds to dropping constraints from the maximum entropy problem, so that the entropy of the tree-structured distribution with matched mean parameters must be larger than the entropy of the original distribution. More formally, we state and prove the following result:

Lemma 2 (Trees as maximum entropy). *For any $\mu \in \mathrm{ri}\,\mathrm{MARG}(G)$, and for any spanning tree $T = (V, E(T))$, we have $-A^*(\mu) \leq -A^*(\mu(T))$.*

Proof. Since T is a subgraph of G, we have $\mathrm{MARG}(G) \subset \mathbb{R}^{d+}$ and $\mathrm{MARG}(T) \subset \mathbb{R}^d$, where $d < d^+$. We use $\mu(T) \in \mathbb{R}^d$ to construct an exponential parameter $\theta^+(T) \in \mathbb{R}^{d^+}$ as follows:

$$\theta^+(T)_\alpha \quad := \quad \begin{cases} \log \mu_{s;j} & \text{if } \alpha = (s;j) \text{ for } s \in V \\ \log \left[\mu_{st;jk} / (\mu_{s;j}\mu_{t;k}) \right] & \text{if } \alpha = (st;jk) \text{ for } (s,t) \in E(T) \\ 0 & \text{otherwise.} \end{cases}$$

The constraint $\mu \in \mathrm{ri}\,\mathrm{MARG}(G)$ ensures that $\mu_\alpha > 0$ for all indices α, so that the logarithms are well-defined. Let $\theta(T)$ denote the lower-dimensional vector in $\mathbb{R}^d$, obtained by removing the zeroed-out coordinates in $\theta^+(T)$. Observe that by construction, $\mu(T)$ and $\theta(T)$ are dually coupled with respect to T (i.e., $\mu(T) = \Lambda(\theta(T))$). We can then write:

$$\begin{aligned} A^*(\mu) & \stackrel{(a)}{=} \sup_{\theta \in \Theta} \left\{ \langle \theta, \mu \rangle - A(\theta) \right\} \\ & \geq \langle \theta^+(T), \mu \rangle - A(\theta^+(T)) \\ & \stackrel{(b)}{=} A^*(\mu(T)), \end{aligned}$$

where equality (a) follows from the variational representation of Theorem 2(a), and equality (b) follows because $\mu(T)$ and $\theta(T)$ are dually coupled by construction. □

Since the upper bound of Lemma 2 holds for each spanning tree of the graph, it will also hold for any convex combination of spanning trees. In particular, let us consider a probability distribution $\boldsymbol{\rho}$ over spanning trees:

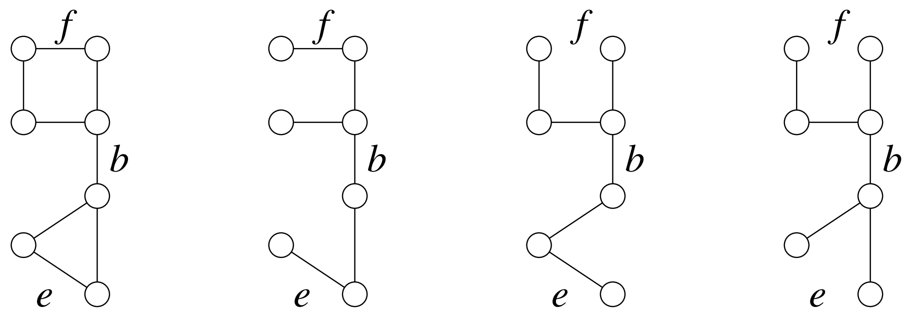
$$\boldsymbol{\rho} \quad = \quad \left\{ \; \rho(T) \mid \sum_T \rho(T) = 1, \;\; \rho(T) \geq 0 \; \right\} \tag{97}$$

In the sequel, it will be necessary to study the probability $\rho_{st} := \mathrm{Pr}_{\boldsymbol{\rho}}\{(s,t) \in T\}$ that a given edge (s,t) belongs to a tree chosen randomly under $\boldsymbol{\rho}$. By definition, the vector $\boldsymbol{\rho_e} = \{\rho_e \mid e \in E\}$ of edge appearance probabilities must belong to the so-called *spanning tree polytope* associated with G, which we denote by $\mathbb{S}(G)$. Let $\mathbb{I}\,[e \in T]$ denote an indicator function for the event that edge e belongs to spanning tree T. The spanning tree polytope is defined as the convex hull of these indicator functions:

$$\mathbb{S}(G) = \{ \boldsymbol{\rho_e} \in \mathbb{R}^{|E|} \mid \forall \; e \; \in \; E \quad \rho_e = \mathbb{E}_{\boldsymbol{\rho}}\{\mathbb{I}\,[e \in T]\} \quad \text{for some } \boldsymbol{\rho} \}. \tag{98}$$

A simple example should help to provide intuition.

Example 21 (Edge appearance probabilities). Figure 21(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 $\boldsymbol{\rho}$ 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 $\boldsymbol{\rho}$, 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 $\rho_b = 1$. ♢



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

Figure 21. 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 $\rho_f = 1/3$.

As defined in equation (98), the spanning tree polytope is the convex hull of a finite—albeit large—number of vectors. Therefore, by the Minkowski-Weyl theorem [92], it has an equivalent characterization in terms of linear inequalities. The linear inequality description of $\mathbb{S}(G)$ is well-known from combinatorial optimization and matroid theory [e.g., 34, 21].

We begin with some definitions: given a subset $F \subseteq E$, let $G(F)$ denote the induced subgraph. Let $v(F)$ be the number of vertices in $G(F)$, and let $c(F)$ be the number of connected components in $G(F)$. The *rank* of F is defined as $r(F) = v(F) - c(F)$. When $G(F)$ is connected so that $c(F) = 1$, then the rank function $r(F)$ corresponds simply to the number of edges in the largest acyclic subgraph of $G(F)$. For example, when $F = E$, then the largest acyclic subgraph is a spanning tree. Since any spanning tree has $n - 1$ edges, we have $r(E) = n - 1$.
The following lemma, based on a result of Edmonds [34], provides a characterization of $\mathbb{S}(G)$:

Lemma 3 (Spanning tree polytope). *The spanning tree polytope $\mathbb{S}(G)$ is characterized completely by the non-negativity condition $\boldsymbol{\rho_e} \geq 0$ combined with the constraints:*

$$\sum_{e \in F} \rho_e \leq r(F) \qquad \forall \quad F \subseteq E, \tag{99a}$$

$$\sum_{e \in E} \rho_e = n - 1. \tag{99b}$$

In order to gain some intuition for the constraints in equation (99), we consider some particular cases. The necessity of the non-negativity constraints $\rho_e \geq 0$ is clear, since each ρ_e corresponds to an edge appearance probability. The corresponding upper bounds $\rho_e \leq 1$ are obtained by applying equation (99a) to the singleton edge set $F = \{e\}$. In this case, we have $v(F) = 2$ and $c(F) = 1$, so that $r(F) = 1$ and equation (99a) reduces to $\rho_e \leq 1$. Finally, equation (99b) can be established via the following argument. Letting $\boldsymbol{\rho} = \{\rho(T)\}$ be the distribution giving rise to the edge appearance probabilities $\boldsymbol{\rho}$, we have the sequence of equalities:

$$\sum_{e \in E} \rho_e = \sum_{e \in E} \sum_{T \in \mathfrak{T}} \rho(T) \mathbb{I}\left[e \in T\right] = \sum_{T \in \mathfrak{T}} \rho(T) \sum_{e \in E} \mathbb{I}\left[e \in T\right] \stackrel{(a)}{=} n - 1.$$

The final equality (a) follows from the fact that any spanning tree T on n nodes has $n - 1$ edges, and hence $\sum_{e \in E} \mathbb{I}\left[e \in T\right] = n - 1$.

8.2 Tree-based upper bound

We now have the necessary ingredients to state and prove an upper bound on A based on a convex combination of trees. Not surprisingly, the resulting variational problem [108, 109] turns out to be closely related to the Bethe variational problem.

Proposition 14 (Tree-based upper bounds). *For any choice of edge appearance vector $\boldsymbol{\rho_e}$ in the spanning tree polytope $\mathbb{S}(G)$, the log partition function is upper bounded by the solution of the following variational problem:*

$$A(\theta) \quad \leq \quad \max_{\tau \in \mathrm{LOCAL}(G)} \big\{ \langle \tau,\, \theta \rangle + \sum_{s \in V} H_s(\tau_s) - \sum_{(s,t) \in E} \rho_{st} I_{st}(\tau_{st}) \big\}. \tag{100}$$

For any $\boldsymbol{\rho_e} \in \mathbb{S}(G)$, this problem is convex, and the optimum is unique if $\rho_e > 0$ for all edges e.

Remark: Observe that equation (100) is closely related to the Bethe variational problem of equation (75). In particular, if we set $\rho_{st} = 1$ for all edges $(s,t) \in E$, then the two formulations are equivalent. However note that $\rho_{st} = 1$ 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. (See, in particular, constraint (99b) in the definition of the spanning tree polytope.) In the context of Proposition 14, then, the ordinary Bethe choice $\boldsymbol{\rho_e} = \mathbf{1}$ is valid only for tree-structured graphs.

Proof: By definition, for any $\boldsymbol{\rho_e} \in \mathbb{S}(G)$, there is an underlying distribution $\boldsymbol{\rho} = \{\rho(T)\}$ such that $\mathbb{E}_{\boldsymbol{\rho}}[\mathbb{I}\,[e \in T] = \rho_e$ for all $e \in E$. By Lemma 2, for any tree T, we have the upper bound $-A^*(\mu) \leq -A^*(\mu(T))$. Taking averages with respect to $\boldsymbol{\rho}$ yields

$$-A^*(\mu) \quad \leq \quad -\mathbb{E}_{\boldsymbol{\rho}}[A^*(\mu(T))] \;=\; -\mathbb{E}_{\boldsymbol{\rho}}\Big[\sum_{s \in V} H_s(\mu_s) - \sum_{(s,t) \in E} I_{st}(\mu_{st})\Big], \tag{101}$$

where we have used the standard decomposition of tree entropy from equation (47). We now expand the expectation over $\boldsymbol{\rho}$ by linearity. Since the trees are all spanning, each entropy term H_s for node $s \in V$ receives a weight of one. On the other hand, the edge (s,t) receives exactly weight $\rho_{st} = \mathbb{E}_{\boldsymbol{\rho}}(\mathbb{I}\,[e \in T])$. Overall, we obtain the following upper bound on the exact entropy:

$$-A^*(\mu) \quad \leq \quad \sum_{s \in V} H_s(\mu_s) - \sum_{(s,t) \in E} \rho_{st} I_{st}(\mu_{st}).$$

Applying this upper bound to the variational formulation of equation (32) yields

$$A(\theta) \quad \leq \quad \max_{\mu \in \mathrm{MARG}(G)} \big\{ \langle \mu,\, \theta \rangle + \sum_{s \in V} H_s(\tau_s) - \sum_{(s,t) \in E} \rho_{st} I_{st}(\tau_{st}) \big\}. \tag{102}$$

Finally, using the fact that $\mathrm{LOCAL}(G)$ is an outer bound on the marginal polytope leads to the upper bound (100).

The cost function consists of a linear term $\langle \theta,\, \mu \rangle$ and a convex combination$-\mathbb{E}_{\boldsymbol{\rho}}[A^*(\mu(T))]$ of tree entropies, and hence is concave. Moreover, the constraint set $\mathrm{LOCAL}(G)$ is a polytope, and hence convex. Therefore, the variational problem (100) is always convex. We now establish uniqueness 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 overcomplete representation, it can be reduced to an equivalent minimal formulation.) To establish uniqueness, it suffices to establish that the function $\mathbb{E}_{\boldsymbol{\rho}}[A^*(\mu(T))]$ is strictly convex when $\rho_e > 0$.

This function is a convex combination of functions of the form $A^*(\mu(T))$, each of which is strictly convex in the (non-zero) components of $\mu(T)$, but independent of the other components in the full vector μ. 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

$$\langle \lambda, \, \nabla^2 A^*(\mu(T)) \lambda \rangle \quad = \quad \langle \Pi^T(\lambda), \, \nabla^2 A^*(\mu(T)) \Pi^T(\lambda) \rangle \;\geq\; 0,$$

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

$$\langle \lambda, \, \mathbb{E}_{\boldsymbol{\rho}}[\nabla^2 A^*(\mu(T))] \, \lambda \rangle \quad \geq \quad \langle \Pi^{T'}(\lambda), \, \nabla^2 A^*(\mu(T')) \Pi^{T'}(\lambda) \rangle \;>\; 0,$$

which establishes the assertion of strict convexity. □

8.3 Tree-reweighted sum-product

Recall that Proposition 11 established that the sum-product algorithm can be understood as an iterative algorithm for attempting to solve the Bethe variational problem (75). Given the close link between the variational formulation of Proposition 14 and the Bethe problem, it is natural to suspect that the sum-product algorithm could be appropriately modified so as to apply to the tree-reweighted case. Indeed this intuition is correct. We first state the form of the tree-reweighted updates [113], and then establish their link to the unique optimal solution of the variational problem (100).

Like the ordinary sum-product updates, the algorithm involves passing messages $M_{ts}(x_s)$ from node at node. These messages are initialized with arbitrary positive numbers, and then updated according to the following recursion:

$$M_{ts}^{n+1}(x_s) \quad = \quad \kappa \sum_{x'_t \in \mathcal{X}_t} \exp\left(\frac{1}{\rho_{st}} \theta_{st}(x_s, x'_t) + \theta_t(x'_t) \right) \left\{ \frac{\prod_{v \in \mathcal{N}(t) \backslash s} \left[M_{vt}^n(x'_t) \right]^{\rho_{vt}}}{\left[M_{st}^n(x'_t) \right]^{(1-\rho_{ts})}} \right\}. \tag{103}$$

Note that the update equation (103) reduces to the ordinary sum-product update under the choice $\boldsymbol{\rho_e} = \mathbf{1}$. In general, however, equation (103) differs from the usual updates in three ways. First, the messages passed along edge (s,t) are reweighted by ρ_{st}. Second, the potential function on edge (s,t) is reweighted by $1/\rho_{st}$. Third, in sharp contrast to ordinary sum-product, the update for message M_{ts} from node t to node s depends on the message M_{st} running in the *reverse direction* on the same edge. The properties of these updates are summarized in the following:

Proposition 15 (Tree-reweighted sum-product). *For any $\boldsymbol{\rho_e} \in \mathbb{S}(G)$ with $\rho_e > 0$, any fixed point M^* of the updates* (103) *specifies the optimal solution of the variational problem* (100) *as follows:*

$$\tau_s^*(x_s) \quad = \quad \kappa \, \exp\left\{ \theta_s(x_s) \right\} \prod_{v \in \mathcal{N}(s)} \left[M_{vs}^*(x_s) \right]^{\rho_{vs}} \tag{104a}$$

$$\tau_{st}^*(x_s, x_t) \quad = \quad \kappa \, \varphi_{st}(x_s, x_t; \theta) \, \frac{\prod_{v \in \mathcal{N}(s) \backslash t} \left[M_{vs}^*(x_s) \right]^{\rho_{vs}}}{\left[M_{ts}^*(x_s) \right]^{(1-\rho_{st})}} \, \frac{\prod_{v \in \mathcal{N}(t) \backslash s} \left[M_{vt}^*(x_t) \right]^{\rho_{vt}}}{\left[M_{st}^*(x_t) \right]^{(1-\rho_{ts})}}, \tag{104b}$$

where $\varphi_{st}(x_s, x_t; \theta) := \exp\{ \frac{1}{\rho_{st}} \theta_{st}(x_s, x_t) + \theta_s(x_s) + \theta_t(x_t) \}$.

Proof. As in the proof of Proposition 11, we enforce the non-negativity constraints (i.e., $\tau_s(x_s) \geq 0$ and $\tau_{st}(x_s, x_t) \geq 0$), as well as the normalization constraints (i.e., $\sum_{x_s} \tau_s(x_s) = 1$) explicitly, without Lagrange multipliers. Assigning a Lagrange multiplier $\lambda_{ts}(x_s)$ to each marginalization constraint of the form $C_{ts}(x_s) := \tau_s(x_s) - \sum_{x_t} \tau_{st}(x_s, x_t)$, we then consider the associated Lagrangian:

$$\mathcal{L}(\tau; \lambda) := \langle \theta, \, \tau \rangle + \sum_{s \in V} H_s(\tau_s) - \sum_{(s,t) \in E} \rho_{st} I_{st}(\tau_{st}) + \sum_{(s,t) \in E} \rho_{st} \big[\lambda_{ts}(x_s) C_{ts}(x_s) + \lambda_{st}(x_t) C_{st}(x_t) \big]. \tag{105}$$

(The factor $\rho_{st} > 0$ in front of the Lagrange multiplier term is simply a convenient rescaling.) By Proposition 14, the original problem is convex and feasible and $\mathrm{LOCAL}(G)$ is polyhedral. Therefore, strong duality holds [8], so that (in contrast to the ordinary Bethe problem), solving the Lagrangian formulation is equivalent to solving the original problem.

Calculations entirely analogous to the proof of Proposition 11 show that stationary points (τ, λ) of the Lagrangian are specified by equations (104a) and (104b), where the messages are defined in terms of Lagrange multipliers as $M_{st} := \exp(\lambda_{st})$ (with the exponential taken elementwise). Finally, enforcing the Lagrangian constraint $\frac{\partial \mathcal{L}}{\partial \lambda_{ts}(x_s)}(\tau; \lambda) = \rho_{st} C_{ts}(x_s) = 0$ yields the message update equation (103). □

Remark: Wiegerinck and Heskes [118] have examined the class of reweighted Bethe problems of the form (100), but without the requirement that the weights ρ_{st} belong to the spanning tree polytope $\mathbb{S}(G)$. 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.

8.4 Convex combinations of hypertrees

In analogy to the hypertree-based extensions of the Bethe variational problem described in Section 7, the definitions and analysis leading up to Propositions 14 and 15 can be extended to hypertrees as well. In this section, we sketch out this extension, and provide a simple example to illustrate.

For a given treewidth t, consider the set of all hypertrees of width less than or equal to t. The intrinsic assumption is that t is sufficiently small that performing exact computations on hypertrees of this width is feasible. It is clear that Lemma 2 generalizes naturally: for any hypertree T, the exact entropy $-A^*(\mu)$ is upper bounded by $-A^*(\mu(T))$ of a hypertree-structured distribution with matched mean parameters. As before, we consider a convex combination of such upper bounds, where the combination is based on a probability distribution $\boldsymbol{\rho} = \{\rho(T)\}$ over the set of all hypertrees of width at most t. Overall, we obtain an upper bound on the entropy of the form

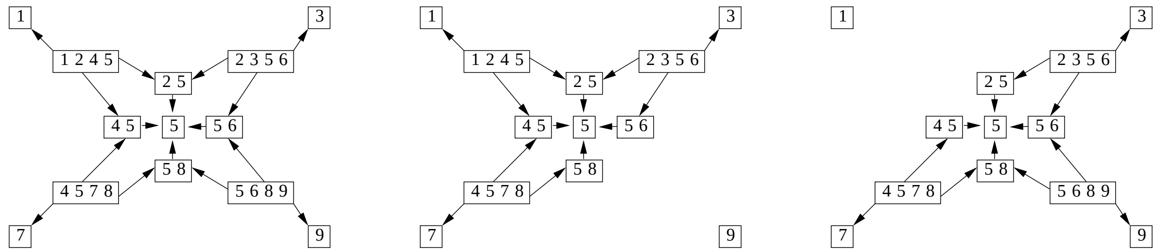
$$A^*(\mu) \quad \leq \quad -\mathbb{E}_{\boldsymbol{\rho}}[A^*(\mu(T))] \; = \; -\sum_{T} \rho(T) A^*(\mu(T)). \tag{106}$$

For a fixed $\boldsymbol{\rho}$, our strategy is to optimize the RHS of this upper bound over all pseudomarginals that are consistent on each hypertree. The resulting constraint set is precisely the polytope $\mathrm{LOCAL}_t(G)$ defined in equation (91).

With this set-up, the hypertree analog of Proposition 14 asserts that the log partition function A is upper bounded as follows:

$$A(\theta) \quad \leq \quad \max_{\tau \in \mathrm{LOCAL}(G)} \big\{ \langle \tau, \, \theta \rangle - \mathbb{E}_{\boldsymbol{\rho}}[A^*(\mu(T))] \big\}. \tag{107}$$

Moreover, the cost function in this variational problem is concave for all choices of distributions $\boldsymbol{\rho}$ over the hypertrees. Equation (107) is the hypertree analog of equation (100); in fact, it reduces to the latter equation in the special case $t = 1$.



(a) (b) (c)

Figure 22. Hyperforests embedded within augmented hypergraphs. (a) An augmented hypergraph for the 3×3 grid with maximal hyperedges of size 4 that satisfies the single counting criterion. (b) One hyperforest of width three embedded within (a). (c) A second hyperforest of width three.

Example 22 (Convex combinations of hypertrees). Let us derive an explicit form of equation (107) for a particular hypergraph and choice of hypertrees. The original graph is the 3×3 grid, as illustrated in the earlier Figure 19(a). Based on this grid, we construct the augmented hypergraph shown in Figure 22(a), which has the hyperedge set

$$E \;:=\; \{\,(1245), (2356), (4578), (5689), (25), (45), (56), (58), (5), (1), (3), (7), (9)\}. \tag{108}$$

It is straightforward to verify that it satisfies the single counting criterion.

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 22(b) is one particular acyclic substructure T^1 with hyperedge set

$$E(T^1) \;=\; \{\,(1245), (2356), (4578), (25), (45), (56), (58), (5), (1), (3), (7), (9)\},$$

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 be simplify the development to follow. Figure 22(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 $h = (1245)$, we see that hyperedges (25), (45), (5), and (1) are contained within it. Thus, using the definition in equation (84), we write (suppressing the functional dependence on $\mathbf{x}$):

$$\varphi_{1245} \;=\; \frac{\tau_{1245}}{\varphi_{25}\;\varphi_{45}\;\varphi_5 \varphi_1} \;=\; \frac{\tau_{1245}}{\frac{\tau_{25}}{\tau_5}\frac{\tau_{45}}{\tau_5}\tau_5\tau_1} \;=\; \frac{\tau_{1245}\;\tau_5}{\tau_{25}\tau_{45}\tau_1}.$$

Having calculated all the functions φ_h, we can combine them, using the hypertree equation (85), in order to obtain the following factorization for a distribution on T^1:

$$p(\mathbf{x};\tau(T^1)) \;=\; \left[\frac{\tau_{1245}\;\tau_5}{\tau_{25}\tau_{45}\tau_1}\right]\left[\frac{\tau_{2356}\;\tau_5}{\tau_{25}\tau_{56}\tau_3}\right]\left[\frac{\tau_{4578}\;\tau_5}{\tau_{45}\tau_{58}\tau_7}\right]\left[\frac{\tau_{25}}{\tau_5}\right]\left[\frac{\tau_{45}}{\tau_5}\right]\left[\frac{\tau_{56}}{\tau_5}\right]\left[\frac{\tau_{58}}{\tau_5}\right]\left[\tau_1\right]\left[\tau_3\right]\left[\tau_5\right]\left[\tau_7\right]\left[\tau_9\right]. \tag{109}$$

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. As in Proposition 13, the factorization (109) 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:

$$\sum_{i=1}^{4} \frac{1}{4} A^*(\tau(T^i)) = \frac{3}{4} \sum_{h \in E_4} \sum_{x_h} \tau_h(x_h) \log \varphi_h(x_h) + \sum_{s \in \{2,4,6,8\}} \sum_{x_{s5}} \tau_{s5}(x_{s5}) \log \frac{\tau_{s5}(x_{s5})}{\tau_5(x_5)} + \sum_{s \in \{1,3,5,7,9\}} \sum_{x_s} \tau_s(x_s) \log \tau_s(x_s). \quad (110)$$

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 (non-maximal) 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 (110) 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:

$$\frac{3}{4}\Big[H_{1245} + H_{2356} + H_{5689} + H_{4578}\Big] - \frac{1}{2}\Big[H_{25} + H_{45} + H_{56} + H_{58}\Big] + \frac{1}{4}\Big[H_1 + H_3 + H_7 + H_9\Big]. \quad (111)$$

If, on the other hand, starting from equation (110) 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 (110) is equal to the following combination of local entropies

$$\Big[H_{1245} + H_{2356} + H_{5689} + H_{4578}\Big] - \Big[H_{25} + H_{45} + H_{56} + H_{58}\Big] + H_5,$$

which is equivalent to the Kikuchi approximation derived in Example 19. 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. $\diamondsuit$

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 [82, 76], 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. With a suitable choice of convex combination, the underlying variational problem will be strictly convex, so that such hypertree-reweighted sum-product algorithms will have a unique fixed point.

9 Semidefinite relaxations for inference

Semidefinite constraints have arisen at several points in the preceding sections, particularly in the context of Gaussian problems. This section is devoted to a more in-depth development of semidefinite constraints for characterizing valid mean parameters. The use of semidefinite constraints for this purpose has a rich history [2, 61], particularly in the context of scalar random variables. The basis of our presentation is more recent work [e.g., 65, 64, 66, 84] that applies to multivariate moment problems. Much of this work applies to a fairly general class of problems, and is based on results from real algebraic geometry, which we do not discuss here. Here we limit ourselves to considering marginal polytopes, and we adopt the statistical perspective of imposing positive semidefiniteness on covariance and other moment matrices. In the course of our development, we establish various results relating the tightness of semidefinite constraints to the underlying graph structure.

We begin with some background on linear matrix inequalities, and then describe how such constraints can be applied to moment matrices. Although semidefinite constraints are more generally applicable, much of our development focuses on the multinomial case. More specifically, we describe a nested sequence of semidefinite outer bounds on the marginal polytopes, the last of which provides an exact characterization for any graph. We also address the role of graphical structure in semidefinite constraints, establishing in particular the link between treewidth and tightness of semidefinite outer bounds. We compare these sequences of semidefinite outer bounds to the hypertree-based outer bounds discussed in Section 7. Finally, to illustrate the use of semidefinite constraints, we combine semidefinite outer bounds with a Gaussian-based entropy approximation to derive a novel log-determinant relaxation for approximate inference [114].

9.1 Moment matrices and semidefinite constraints

Let $\mathcal{S}_+^n$ denote the cone of $n \times n$ symmetric positive semidefinite matrices. For two symmetric matrices A and B, we define the inner product $\langle\langle A,\ B \rangle\rangle := \text{trace}(AB)$. Given a vector $\mu \in \mathbb{R}^d$, consider a linear matrix-valued function $F(\mu) = F_0 + \sum_{i=1}^d \mu_i F_i$, where the matrices $F_i, i = 0, \ldots, d$ are $n \times n$ and symmetric. Requiring that $F(\mu)$ be positive semidefinite, which we denote by $F(\mu) \succeq 0$, is a *linear matrix inequality* (LMI). The class of constraints that can be expressed in this manner is fairly broad, including as special cases both linear and quadratic constraints [see 104]. For instance, the linear constraint $A\mu \geq b$ is equivalent to the LMI $\text{diag}\{A\mu - b\} \succeq 0$, where the diag operator places the elements of a vector on the diagonal of a matrix. In general, the constraint set carved out by a linear matrix inequality is formed of a mixture of polyhedral (i.e., linear) and curved constraints.

Given a random vector $\mathbf{y}$ with n components, let $\lambda_{st} = \mathbb{E}[y_s y_t]$ denote its second-order moments. Using these moments, we can form the following symmetric $n \times n$ matrix:

$$M[\lambda] \;=\; \mathbb{E}[\mathbf{y}\mathbf{y}^T] \;=\; \begin{bmatrix} \lambda_{11} & \lambda_{12} & \cdots & \lambda_{1n} \\ \lambda_{21} & \lambda_{22} & \cdots & \lambda_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \lambda_{n1} & \lambda_{n2} & \cdots & \lambda_{nn} \end{bmatrix} \tag{112}$$

At first sight, this definition might seem limiting, because the matrix involves only second-order moments. However, given some random vector $\mathbf{x}$ of interest, we can expose any of its moments by defining $\mathbf{y} = f(\mathbf{x})$ for a suitable choice of function f, and then considering the associated second-order moment matrix (112) for $\mathbf{y}$. For instance, by setting $\mathbf{y} = [1 \;\; \mathbf{x}]$, the moment matrix (112) will include both first and second-order moments of $\mathbf{x}$. Similarly, by including terms of the form $x_s x_t$ in the definition of $\mathbf{y}$, we can expose third moments of $\mathbf{x}$.

The significance of the moment matrix (112) lies in the following simple result:

Lemma 4 (Moment matrices). *Any valid moment matrix $M[\lambda]$ is positive semidefinite.*

Proof. We must show that $a^T M[\lambda] a \geq 0$ for an arbitrary vector $a \in \mathbb{R}^n$. If λ is a valid moment vector, then it arises by taking expectations under some distribution p. Accordingly, we can write $a^T M[\lambda] a \;=\; \mathbb{E}_p[a^T \mathbf{y}\mathbf{y}^T a] \;=\; \mathbb{E}_p[\|a^T \mathbf{y}\|^2]$, which is clearly non-negative. □

Remarks: Lemma 4 provides a *necessary* condition for a collection $\{\lambda_{st}\}$ to be a valid set of second-order moments. Such a condition is both necessary and sufficient for certain classical moment problems involving scalar random variables [e.g., 61, 54]. This condition is of course also necessary and sufficient for a Gaussian random vector, as stated in Proposition 6.

9.2 Semidefinite outer bounds on marginal polytopes

We now turn to the use of semidefinite constraints in providing outer bounds on marginal polytopes associated with multinomial random vectors.

9.2.1 Multi-index notation

Recall the exponential representation of the Ising model in Example 3, which was based on sufficient statistics of the form x_s and $x_s x_t$. The natural generalization of this representation to non-binary discrete variables is based on monomials of the form $\mathbf{x}^\alpha := \prod_{s=1}^n x_s^{\alpha_s}$, where $\alpha := (\alpha_1, \alpha_2, \ldots, \alpha_n)$ is a vector of non-negative indices α_s. We refer to α as a *multi-index.* Our convention for the all-zeros multi-index 0 is that $\mathbf{x}^0 = 1$. Given two multi-indices α and β, it will be useful to specify their component-wise sum $\alpha + \beta = (\alpha_1 + \beta_1, \ldots, \alpha_n + \beta_n)$.

Consider a multinomial random vector $\mathbf{x}$, where each x_s takes values in $\mathcal{X} := \{0, 1, \ldots, m-1\}$. (A bit more generally, we could allow the cardinality of $\mathcal{X}_s$ to vary for each vertex.) A convenient exponential representation, based on the monomials $\mathbf{x}^\alpha$, is as follows:

$$p(\mathbf{x}; \theta) \quad = \quad \exp\big\{\sum_\alpha \theta_\alpha \mathbf{x}^\alpha - A(\theta)\big\}. \tag{113}$$

Without loss of generality, the range of the sum over α in equation (113) can be restricted. In particular, observe that for any multinomial variable $x \in \mathcal{X} = \{0, 1, \ldots, m-1\}$, there always holds

$$\prod_{j=0}^{m-1} (x - j) = 0. \tag{114}$$

A minor re-arrangement of this relation yields an expression for x^m as a polynomial of degree $m-1$, which implies that any monomial x^i with $i \geq m$ can be expressed as a linear combination of lower-order monomials. Therefore, we can always assume without loss of generality that the sum is taken only over multi-indices for which the maximum degree $\|\alpha\|_\infty := \max_s \alpha_s$ is less than or equal to $m-1$. Herein all multi-indices should be understood to satisfy this restriction.

Particular classes of models are obtained by imposing constraints on the set of α. For instance, restricting α to be non-zero in at most two positions corresponds to a pairwise Markov random field. We can write this constraint compactly using the ℓ_0 norm $\|\alpha\|_0 := \#\{s \mid \alpha_s > 0\}$, which counts the number of non-zero entries. With this notation, the set of monomials $\mathbf{x}^\alpha$ associated with a pairwise Markov random field are those with multi-indices in the set $\mathcal{I}_2 = \{\alpha \mid \|\alpha\|_0 \leq 2\}$. More generally, for each integer $k = 1, \ldots, n$, we define the multi-index set $\mathcal{I}_k = \{\alpha \mid \|\alpha\|_0 \leq k\}$. This nested set of multi-index sets describes a hierarchy of Markov random field models, defined on hypergraphs with increasing sizes of hyperedges.

To calculate the cardinality of $\mathcal{I}_k$, observe that for each $i = 0, \ldots, k$, there are $\binom{n}{i}$ possible subsets of size i. Moreover, for each member of each such subset, there are $(m-1)$ possible choices of the index value, so that $\mathcal{I}_k$ has $\sum_{i=0}^{k} \binom{n}{i} \, (m-1)^i$ elements in total. The total number of all possible multi-indices (with $\|\alpha\|_\infty \leq m-1$) is given by $|\mathcal{I}_n| = \sum_{i=0}^{n} \binom{n}{i} (m-1)^i \; = \; m^n$.

9.2.2 First-order semidefinite outer bound

For any multi-index α, let $\mu_\alpha = \mathbb{E}[\mathbf{x}^\alpha]$ denote the associated mean parameter or moment. For each $k = 1, \ldots, n$, let us introduce $K_{k,n}$ to denote the hypergraph that includes *all* hyperedges of size up to k on a set of n nodes. For instance, $K_{1,n}$ is simply a disconnected graph, whereas $K_{2,n} \equiv K_n$ is the usual complete graph on n nodes. We can then consider the associated marginal polytope

$$\mathrm{MARG}(K_{k,n}) \;\; := \;\; \{\mu_\alpha \in \mathbb{R}^{|\mathcal{I}_k|} \mid \alpha \in \mathcal{I}_k \}, \tag{115}$$

which corresponds to all valid moments μ_α of order $\|\alpha\|_0 \leq k$. More generally, for any hypergraph G, we use $\mathrm{MARG}(G)$ to denote the associated marginal polytope.

We now show how to use moment matrices to develop semidefinite outer bounds on marginal polytopes. For concreteness, we focus on a pairwise Markov random field, so that the relevant singleton and pairwise moments belong to the set $\mathrm{MARG}(K_{2,n}) \equiv \mathrm{MARG}(K_n)$. Given a random vector, we denote by $M_1[\mu]$ the moment matrix corresponding to the choice $\mathbf{y} = \{\, \mathbf{x}^\alpha \mid \alpha \in \mathcal{I}_1\}$ in equation (112). Explicitly, the rows and columns of $M_1[\mu]$ are indexed by multi-indices $\alpha, \beta \in \mathcal{I}_1$, where entry (α, β) is given by

$$\big(M_1[\mu]\big)_{\alpha\beta} \;\; = \;\; \mu_{\alpha+\beta}. \tag{116}$$

Since $\|\alpha\|_0 \leq 1$ for each $\alpha \in \mathcal{I}_1$, there always holds $\|\alpha + \beta\|_0 \leq 2$.

Example 23 (Binary case). We illustrate $M_1[\mu]$ explicitly for the binary case $\mathbf{x} \in \{0,1\}^n$, for which $\{\mathbf{x}^\alpha \mid \alpha \in \mathcal{I}_1\} = (1, x_1, \ldots, x_n)$. On this basis, we calculate:

$$M_1[\mu] \;\; := \;\; \begin{bmatrix} 1 & \mu_1 & \mu_2 & \cdots & \mu_{n-1} & \mu_n \\ \mu_1 & \mu_1 & \mu_{12} & \cdots & \cdots & \mu_{1n} \\ \mu_2 & \mu_{12} & \mu_2 & \cdots & \cdots & \mu_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mu_{n-1} & \vdots & \vdots & \vdots & \vdots & \mu_{(n-1),n} \\ \mu_n & \mu_{n1} & \mu_{n2} & \cdots & \mu_{(n-1),n} & \mu_n \end{bmatrix}. \tag{117}$$

An important point to note is that in forming $M_1[\mu]$, we use the fact that $x_s^2 = x_s$ for any $x_s \in \{0,1\}$ in order to simplify the moment calculations. In particular, for each of the diagonal terms (other than the 1 in the $(1,1)$ entry), we use the fact that $\mathbb{E}[x_s^2] = \mathbb{E}[x_s] = \mu_s$. In the general multinomial case, similar simplifications follow from equation (114). $\diamondsuit$

We now use the matrix $M_1[\mu]$ to define the following semidefinite constraint set:

$$\mathrm{SDEF}_1 \;\; := \;\; \{\mu_\alpha, \alpha \in \mathcal{I}_2 \mid M_1[\mu] \succeq 0 \,\}. \tag{118}$$

The definition of $M_1[\mu]$ and Lemma 4 guarantee the following inclusion:

Lemma 5 (First-order outer bound). *The marginal polytope* $\mathrm{MARG}(K_n)$ *is contained within the semidefinite constraint set* SDEF_1.

Example 24. To illustrate Lemma 5, recall the (hitherto unproven) claim of Example 14: for the fully connected graph K_3 on three nodes, the following pseudomarginal τ lies outside $\mathrm{MARG}(K_3)$:

$$\tau_s = 0.5 \text{ for } s = 1, 2, 3, \qquad \tau_{12} = \tau_{23} = 0.4, \qquad \tau_{13} = 0.1.$$

(Note that we have translated the overcomplete canonical representation of Example 14 to a minimal representation.) We now construct the matrix M_1 for this trial set of mean parameters:

$$M_1[\tau] = \begin{bmatrix} 1 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.4 & 0.1 \\ 0.5 & 0.4 & 0.5 & 0.4 \\ 0.5 & 0.1 & 0.4 & 0.5 \end{bmatrix}.$$

A simple calculation shows that it is not positive definite, whence $\tau \notin \mathrm{SDEF}_1$. Applying Lemma 5 yields that $\tau \notin \mathrm{MARG}(K_3)$. $\diamondsuit$

9.2.3 Projections and exactness

Lemma 5 shows that the semidefinite constraint set SDEF_1 provides an outer bound on the set $\mathrm{MARG}(K_n)$ of valid second-order marginals. This same constraint set also induces an outer bound on $\mathrm{MARG}(G)$, where G is any subgraph of the complete graph K_n, in the following way. Let $\mathcal{I}(G) \subset \mathcal{I}(K_n)$ be the multi-index sets associated with G and K_n respectively. Given any outer bound $\mathrm{OUT}(K_n)$ on $\mathrm{MARG}(K_n)$, we define its *projection* onto the coordinates of $\mathcal{I}(G)$ as follows:

$$\Pi_G(\mathrm{OUT}(K_n)) = \big\{\mu_\alpha,\ \alpha \in \mathcal{I}(G) \mid \mu_\alpha = \eta_\alpha \text{ for some } \eta \in \mathrm{OUT}(K_n)\big\}. \tag{119}$$

With this definition, an immediate corollary of Lemma 5 is that $\Pi_G(\mathrm{SDEF}_1)$ is an outer bound on $\mathrm{MARG}(G)$ for any graph.

We now turn to a natural question: in which cases does SDEF_1 (or a suitable projection thereof) provide an exact description of a marginal polytope? To gain intuition, let us return to the binary case of Example 23. It can be seen that for any moment μ_s, the matrix $M_1[\mu]$ of equation (116) contains a 2×2 principal submatrix of the form

$$\begin{bmatrix} 1 & \mu_s \\ \mu_s & \mu_s \end{bmatrix}.$$

The positive semidefiniteness of this submatrix enforces the constraint $\mu_s\,(1 - \mu_s) \geq 0$, which is equivalent to the interval constraint $\mu_s \in [0, 1]$. Note that the marginal polytope $\mathrm{MARG}(K_{1,n})$, which consists only of the first-order moments μ_s, is completely characterized by these interval constraints. Therefore, we conclude that $\Pi_{K_{1,n}}(\mathrm{SDEF}_1) = \mathrm{MARG}(K_{1,n})$. This equivalence can be extended easily to the general multinomial case (i.e., $m > 2$).

This exactness breaks down for more interesting examples. For instance, SDEF_1 is a *strict* outer bound on the marginal polytope $\mathrm{MARG}(K_{2,n}) \equiv \mathrm{MARG}(K_n)$, as illustrated in the following example.

Example 25 (Strict inclusion for binary pair). Let us demonstrate the strict inclusion $\mathrm{SDEF}_1 \supset \mathrm{MARG}(K_n)$ for a pair (x_1, x_2) of binary random variables (i.e., $n = 2$). In this case, $\mathrm{MARG}(K_2)$ consists of three moments $\{\mu_1, \mu_2, \mu_{12}\}$. So as to facilitate visualization, we focus on the intersection of both the marginal polytope and the constraint set SDEF_1 with the hyperplane $\mu_1 = \mu_2$. The semidefinite constraint set is defined by the LMI constraint:

$$M_1[\mu] \;=\; \begin{bmatrix} 1 & \mu_1 & \mu_2 \\ \mu_1 & \mu_1 & \mu_{12} \\ \mu_2 & \mu_{12} & \mu_2 \end{bmatrix} \succeq 0. \tag{120}$$

In order to deduce the implied constraints, we apply Sylvester's criterion, making the substitution $\mu_1 = \mu_2$ throughout. Positivity of the $(1,1)$ subminor is trivial $(1 > 0)$, and the $(1,2)$ subminor yields the interval constraint $\mu_1 \in [0,1]$. After some simplification, non-negativity of the full determinant leads to the constraint $\mu_{12}^2 - (2\mu_1^2)\mu_{12} + (2\mu_1^3 - \mu_2^2) \leq 0$. Viewing the LHS as a quadratic in μ_{12}, we can factor it into the product $[\mu_{12} - \mu_1]\,[\mu_{12} - \mu_1\,(2\mu_1 - 1)]$. For $\mu_1 \in [0,1]$, this quadratic inequality is equivalent to the pair of constraints

$$\mu_{12} \;\leq\; \mu_1, \qquad\qquad \mu_{12} \;\geq\; \mu_1\,(2\mu_1 - 1). \tag{121}$$

The gray area in Figure 23 shows the intersection of the marginal polytope $\mathrm{MARG}(K_2)$ with

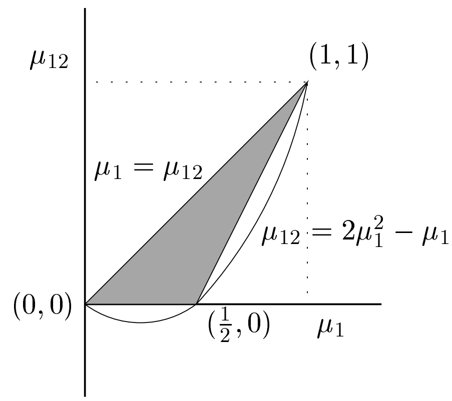


Figure 23. Nature of the semidefinite outer bound SDEF_1 on the marginal polytope $\mathrm{MARG}(K_2)$ for a pair $(x_1, x_2) \in \{0,1\}^2$. The gray area shows the cross-section of the binary marginal polytope $\mathrm{MARG}(K_2)$ corresponding to intersection with the hyperplane $\mu_1 = \mu_2$. The intersection of SDEF_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 SDEF_1 that lie strictly outside $\mathrm{MARG}(K_2)$.

the hyperplane $\mu_1 = \mu_2$. The intersection of the semidefinite constraint set SDEF_1 with this same hyperplane is characterized by the interval inclusion $\mu_1 \in [0,1]$ and the two inequalities in equation (121). Note that the semidefinite constraint set is an outer bound on $\mathrm{MARG}(K_2)$, 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 $M_1[\mu]$, but this vector certainly does not belong to $\mathrm{MARG}(K_2)$. $\diamondsuit$

9.2.4 Higher-order semidefinite constraints

The previous construction of $M_1[\mu]$ was based only on first and second-order moments of the random vector $\mathbf{x}$. Of course, we can also consider moments μ_α for higher-order multi-indices α as well; doing so leads a special case of what is known as the Lasserre sequence of relaxations [64, 66].

If the given model has higher than pairwise interactions, then considering such higher-order moments is absolutely necessary. However, it may also be useful even when considering a pairwise

Markov random field on an ordinary graph G (i.e., for which $\mathrm{MARG}(G)$ involves only pairwise moments). Indeed, suppose that we have an outer bound on $\mathrm{MARG}(K_{k,n})$ for some $k \geq 3$. For any graph G, this outer bound can be projected, as in equation (119), to obtain an outer bound on $\mathrm{MARG}(G)$.

Accordingly, for each $t = 2, \ldots, n$, we form an $|\mathcal{I}_t| \times |\mathcal{I}_t|$ matrix $M_t[\mu]$ where each row and column is associated with some multi-index $\alpha \in \mathcal{I}_t$. The entries of $M_t[\mu]$ are specified as follows:

$$\big(M_t[\mu]\big)_{\alpha\beta} \;=\; \mu_{\alpha+\beta}. \tag{122}$$

When $t = 1$, this definition reduces to our previous definition of $M_1[\mu]$ in equation (116). Note any moment $\mu_{\alpha+\beta}$ involved in $M_t[\mu]$ has order $\|\alpha+\beta\|_0 \leq \min\{2t, n\}$.

Example 26 (Higher-order semidefinite constraint). To provide a simple illustration, consider a triplet (x_1, x_2, x_3) of binary variables, so that

$$\{\mathbf{x}^\alpha \mid \alpha \in \mathcal{I}_2\} \;=\; \{1,\ x_1,\ x_2,\ x_3,\ x_1x_2,\ x_1x_3,\ x_2x_3\}$$

In this case, the matrix $M_2[\mu]$ is 7×7, and takes the following form:

$$M_2[\mu] \;=\; \begin{bmatrix} 1 & \mu_1 & \mu_2 & \mu_3 & \mu_{12} & \mu_{13} & \mu_{23} \\ \mu_1 & \mu_1 & \mu_{12} & \mu_{13} & \mu_{12} & \mu_{13} & \mu_{123} \\ \mu_2 & \mu_{12} & \mu_2 & \mu_{23} & \mu_{12} & \mu_{123} & \mu_{23} \\ \mu_3 & \mu_{13} & \mu_{23} & \mu_3 & \mu_{123} & \mu_{13} & \mu_{23} \\ \mu_{12} & \mu_{12} & \mu_{12} & \mu_{123} & \mu_{12} & \mu_{123} & \mu_{123} \\ \mu_{13} & \mu_{13} & \mu_{123} & \mu_{13} & \mu_{123} & \mu_{13} & \mu_{123} \\ \mu_{23} & \mu_{123} & \mu_{23} & \mu_{23} & \mu_{123} & \mu_{123} & \mu_{23} \end{bmatrix} \tag{123}$$

In calculating the form of $M_2[\mu]$, we use the fact that $x_s^2 = x_s$ whenever $x_s \in \{0,1\}$ in order to simplify the moment calculations. For example, in calculating the $(5,7)$ entry, we use the reduction $\mathbb{E}[(x_1x_2)\,(x_2x_3)] = \mathbb{E}[x_1x_2x_3] = \mu_{123}$. $\diamondsuit$

As with the argument preceding Lemma 5, for each $t = 1, \ldots, n$, the matrix $M_t[\mu]$ can be used to specify an outer bound[13] on the marginal polytope $\mathrm{MARG}(K_{2t,n})$. In particular, we use $M_t[\mu]$ to define the following semidefinite constraint set:

$$\mathrm{SDEF}_t \;:=\; \{\mu_\alpha, \alpha \in \mathcal{I}_{2t} \mid M_t[\mu] \succeq 0\,\}. \tag{124}$$

Note that when $t = 1$, definition (124) is equivalent to the earlier definition of SDEF_1 in equation (118). In analogy to Lemma 5, these semidefinite constraints generate outer bounds on marginal polytopes:

Lemma 6. *For each $t = 1, \ldots, n$, the set SDEF_t is an outer bound on $\mathrm{MARG}(K_{2t,n})$. Moreover, for any hypergraph G contained within $K_{2t,n}$, the projection $\Pi_G(\mathrm{SDEF}_t)$ is an outer bound on $\mathrm{MARG}(G)$.*

An important property of this sequence of outer bounds is that they are *nested.* Considering in particular $\mathrm{MARG}(K_n) \equiv \mathrm{MARG}(K_{2,n})$, we have the set of inclusions

$$\mathrm{SDEF}_1 \;=\; \Pi_{K_n}(\mathrm{SDEF}_1) \;\supseteq\; \Pi_{K_n}(\mathrm{SDEF}_2) \;\supseteq \ldots \supseteq\; \Pi_{K_n}(\mathrm{SDEF}_n). \tag{125}$$

[13]Strictly speaking, it defines an outer bound on $\mathrm{MARG}(K_{r(t),n})$ where $r(t) := \min\{2t, n\}$, but we suppress the subtlety in the interests of readability.

This nesting relation holds because for $t' < t$, the matrix $M_{t'}[\mu]$ is a principal minor of the larger matrix $M_t[\mu]$. For instance, observe that for the binary case, the matrix $M_1[\mu]$ of equation (120) is equivalent to the top 3×3 block of $M_2[\mu]$ defined in equation (123).

We have terminated the nested sequence in equation (125) at SDEF_n. The validity of this finite termination in a general setting was proved by Lasserre [64], and also by Laurent [66, 67] using different methods. Here we provide an alternative proof of finite termination for characterizing a multinomial marginal polytope:

Proposition 16 (Tightness of semidefinite constraints). *For any multinomial random vector $\mathbf{x} \in \{0, 1, \ldots, m-1\}^n$, the semidefinite constraint set SDEF_n and its projections provide an exact characterization of the marginal polytope $\mathrm{MARG}(G)$ for any hypergraph G.*

Proof. For each $J = (j_1, \ldots, j_n) \in \mathcal{X}^n$, define the indicator function $\mathbb{I}_J(\mathbf{x}) := \prod_{s=1}^n \mathbb{I}_{j_s}(x_s)$. First consider the following identities between the scalar indicator functions $\mathbb{I}_j(u)$ and monomials u^j:

$$\mathbb{I}_j(u) \;=\; \prod_{k \neq j} \frac{u-k}{j-k}, \qquad\qquad u^j \;=\; \sum_{k=0}^{m-1} k^j \mathbb{I}_k(u). \tag{126}$$

For each $\alpha \in \mathcal{I}_n$, the monomial $\mathbf{x}^\alpha$ decomposes as the product $\prod_{s=1}^n x_s^{\alpha_s}$, so that it is a linear combination of the indicators $\mathbb{I}_J(\mathbf{x}) := \prod_{s=1}^n \mathbb{I}_{j_s}(x_s)$. Conversely, for each $J \in \mathcal{X}^n$, the indicator function $\mathbb{I}_J(\mathbf{x})$ is also equal to a linear combination of the monomials $\{\mathbf{x}^\alpha, \alpha \in \mathcal{I}_n\}$. Thus, there is an invertible linear transformation (with matrix B) between the indicator functions $\{\mathbb{I}_J(\mathbf{x}), J \in \mathcal{X}^n\}$ and the monomials $\{\mathbf{x}^\alpha, \alpha \in \mathcal{I}_n\}$.

Consider the $m^n \times m^n$ moment matrix defined by the functions $\{\mathbb{I}_J(\mathbf{x}), J \in \mathcal{X}^n\}$. Its form is very simple: since the product $\mathbb{I}_J(\mathbf{x})\mathbb{I}_{J'}(\mathbf{x})$ vanishes for all $J \neq J'$, it is a diagonal matrix $D = \mathrm{diag}(\mu_J)$, where μ_J is the probability of the configuration $J \in \mathcal{X}^n$. Given the constraint $\sum_J \mathbb{I}_J(\mathbf{x}) = 1$, the positive semidefinite constraint $D \succeq 0$ is necessary and sufficient to ensure that $\{\mu_J, \quad J \in \mathcal{X}^n\}$ specifies a valid probability distribution. Moreover, by the linear bijection established above, we have $M_n[\mu] = B\,DB^T$ with B invertible, so that $D \succeq 0$ if and only if $M_n[\mu] \succeq 0$. $\square$

Remarks: This result shows that imposing a semidefinite constraint on the largest possible moment matrix $M_n[\mu]$ is sufficient to fully characterize all marginal polytopes. From a practical point of view, however, the consequences of this result are limited, because $M_n[\mu]$ is a $|\mathcal{I}_n| \times |\mathcal{I}_n|$ matrix, where $|\mathcal{I}_n| = |\mathcal{X}^n| = m^n$ is exponentially large.

To illustrate Proposition 16, we consider a very simple example.

Example 27. Consider the marginal polytope $\mathrm{MARG}(K_2)$ for a binary pair $(x_1, x_2) \in \{0,1\}^2$. In this case, the full moment matrix $M_2[\mu]$ is 4×4, corresponding to the set $\{1, x_1, x_2, x_1x_2\}$. It takes the form

$$M_2[\mu] \;=\; \begin{bmatrix} 1 & \mu_1 & \mu_2 & \mu_{12} \\ \mu_1 & \mu_1 & \mu_{12} & \mu_{12} \\ \mu_2 & \mu_{12} & \mu_2 & \mu_{12} \\ \mu_{12} & \mu_{12} & \mu_{12} & \mu_{12} \end{bmatrix}. \tag{127}$$

Positivity of the diagonal element $(4,4)$ gives the constraint $\mu_{12} \geq 0$. Positivity of the $(3,4)$ subminor, combined with the constraint $\mu_{12} \geq 0$, leads to $\mu_2 - \mu_{12} \geq 0$. By symmetry, the $(2,4)$ subminor gives $\mu_1 - \mu_{12} \geq 0$. Finally, the determinant of $M_2[\mu]$ can be calculated

$$\det M_2[\mu] \;=\; \mu_{12}\,\big[\mu_1 - \mu_{12}\big]\,\big[\mu_1 - \mu_{12}\big]\,\big[1 + \mu_{12} - \mu_1 - \mu_2\big]. \tag{128}$$

The constraint $\det M_2[\mu] \geq 0$, 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 $M_2[\mu]$, so positive semidefiniteness of $M_2[\mu]$ is equivalent to non-negativity of these four quantities.) These four inequalities provide a complete description of the marginal polytope in this simple case, as can be seen by comparison to Example 7. It is also worthwhile comparing to Example 25, where we showed that positive semidefiniteness of the 3×3 moment matrix $M_1[\mu]$, which is simply the $(1, 2, 3)$ principal submatrix of $M_2[\mu]$, provides only a partial characterization of $\mathrm{MARG}(K_2)$.

$\diamondsuit$

9.3 Link to graphical structure

Recall from our discussion in Section 4.1.3 that the complexity of a given marginal polytope $\mathrm{MARG}(G)$ depends very strongly on the structure of the (hyper)graph G. We now turn to a more detailed consideration of the role of graphical structure in semidefinite constraints. One consequence of the junction tree theorem, as stated in Proposition 1, is that marginal polytopes associated with hypertrees are straightforward to characterize. This simplicity is also apparent in the context of semidefinite characterizations.

9.3.1 Notation for graph-structured semidefinite constraints

Before turning to results, we require some additional notation. Given a hypergraph H, let $\mathcal{I}(H)$ be the set of multi-indices associated with all possible monomials $\mathbf{x}^\alpha$ defined on its hyperedges. For example, if H is simply the complete graph K_n, then the set $\mathcal{I}(K_n)$ consists of all multi-indices satisfying $\|\alpha\|_0 \leq 2$. Let $M_H[\mu]$ be the $|\mathcal{I}(H)| \times |\mathcal{I}(H)|$ moment matrix defined by $\{\mathbf{x}^{\alpha+\beta},\ \alpha, \beta \in \mathcal{I}(H)\}$. Note that $M_H[\mu]$ generalizes the previously defined moment matrix $M_t[\mu] \equiv M_{K_{t,n}}[\mu]$, where $K_{t,n}$ is the complete hypergraph including all hyperedges of size less than or equal to t.

Using these moment matrices, we define the semidefinite constraint sets

$$\mathrm{SDEF}_H \;\; := \;\; \{\mu_{\alpha+\beta},\;\; \alpha, \beta \in \mathcal{I}(H) \mid M_H[\mu] \succeq 0\}. \tag{129}$$

Observe that the set SDEF_H is a generalization of the semidefinite constraint sets SDEF_t defined in equation (124); more specifically, SDEF_t is equivalent to $\mathrm{SDEF}_{K_{t,n}}$. Finally, we use $\mathrm{SDEF}(G)$ as short-hand for the projected set $\Pi_G[\mathrm{SDEF}_G]$, where the projection is defined as in equation (119).

9.3.2 Semidefinite characterization of hypertrees

When the hypergraph G is a hypertree, then its marginal polytope is characterized by a relatively small set of semidefinite constraints:

Proposition 17 (Hypertrees). *For any hypertree G, there holds*

$$\mathrm{SDEF}(G) \;\; = \;\; \mathrm{MARG}(G). \tag{130}$$

Proof. For any hyperedge h of G, let $\mathcal{I}(h)$ denote the associated multi-indices (including $\alpha = \mathbf{0}$ for the empty subset), and define $k := |\mathcal{I}(h)|$. By definition, for every hyperedge h of the hypertree G, the moment matrix $M_G[\mu]$ includes an $k \times k$ principal submatrix, corresponding to all of moments of the form $\mu_{\alpha+\beta}$ for pairs $\alpha, \beta \in \mathcal{I}(h)$. For the subset of random variables $x_h := \{x_s \mid s \in h\}$, this principal submatrix is equivalent to the matrix $M_{K_{k,k}}[\mu]$. By Sylvester's criterion [54], the positive semidefiniteness of $M_G[\mu]$ implies that all principal submatrices must be positive semidefinite.

By Proposition 16, the positive semidefiniteness of $M_{K_{k,k}}[\mu]$ implies that the mean parameters $\{\mu_\alpha \mid \alpha \in \mathcal{I}(h)\}$ are locally consistent over the hyperedge h. The junction tree characterization of Proposition 1 then guarantees global consistency. □

Remarks: (a) This result is of an analogous nature to the junction tree sufficiency condition of Proposition 1. It is worthwhile contrasting with the earlier Proposition 16, which guarantees tightness of semidefinite constraints involving the *full* moment matrix (of size m^n). The essence of Proposition 17 is that if, in addition, G is a hypertree, a much lower order of semidefinite constraints provides a complete characterization of the marginal polytope. In particular, for a hypergraph G with maximal hyperedges of size $t+1$, the moment matrix $M_G[\mu]$ is only of size $\mathcal{O}(m^{t+1}|E|)$.
(b) As a particular example, consider the case of an ordinary tree T, which has treewidth $t = 1$. Proposition 17 then asserts that $\mathrm{SDEF}(T)$, which is defined by a moment matrix with only $\mathcal{O}(m^2 n)$ elements, is an exact characterization of the tree marginal polytope $\mathrm{MARG}(T)$.

9.3.3 Comparison to junction tree

In Section 7, we described outer bounds on the marginal polytope of an arbitrary hypergraph based on hypertree consistency. It is worthwhile understanding the connection between such outer bounds $\mathrm{LOCAL}_t(G)$, as defined[14] in equation (91), and the constraint sets $\mathrm{SDEF}(G)$ defined in the previous section.

First of all, whenever G is a hypertree, there holds

$$\mathrm{LOCAL}(G) \quad \stackrel{(a)}{=} \quad \mathrm{MARG}(G) \quad \stackrel{(b)}{=} \quad \mathrm{SDEF}(G), \tag{131}$$

where equality (a) is a consequence of Proposition 1, and equality (b) is the assertion of Proposition 17. More generally, the following relation holds:

Proposition 18. *For any hypergraph G, we have* $\mathrm{SDEF}(G) \subseteq \mathrm{LOCAL}(G)$.

Proof. The proof is similar to the proof of Proposition 17. In particular, for any hyperedge h in the hypergraph, set $k := |\mathcal{I}(h)|$, and observe that the moment matrix $M_G[\mu]$ contains a principal submatrix of the form $M_{K_{k,k}}[\mu]$, where $K_{k,k}$ is the complete k-hypergraph on the vertices in the hyperedge h. The positive semidefiniteness of this principal submatrix enforces the constraint that $\{\mu_\alpha, \alpha \in \mathcal{I}(h)\}$ defines a valid local marginal. Therefore, the constraints defining $\mathrm{SDEF}(G)$ imply those defining $\mathrm{LOCAL}(G)$, thereby establishing containment. □

Remarks: (a) The sequences $\mathrm{SDEF}(K_{t,n})$ and $\mathrm{LOCAL}(K_{t,n})$ defined by the complete hypergraphs $K_{t,n}$ (for $t = 1, \ldots, n$) correspond to particular cases of the Lasserre [64] and Sherali-Adams [98] sequences of relaxations respectively. See Laurent [66, 67] for comparison of these sequences in a more general setting.
(b) An interesting by-product of Propositions 17 and 18, or rather of their proofs, is showing that the linear constraint set $\mathrm{LOCAL}(G)$ can be viewed as an intersection of locally-defined semidefinite constraint sets. In particular, for any hyperedge g of E, let H_g be the subhypergraph with vertex set equal to g, and hyperedge set given by the power set of g (i.e., all subsets of g are hyperedges).

[14] For simplicity in notation, we omit the subscript t from hereon, understanding that it can be inferred from the underlying hypergraph G.

With this definition, the semidefinite constraint set $\mathrm{SDEF}(H_g)$ is defined by the moment matrix $M_{K_{k,k}}[\mu]$, where $k = |\mathcal{I}(g)|$. With this notation, we have the following equivalence:

$$\mathrm{LOCAL}(G) \quad = \quad \bigcap_{g \in E} \mathrm{SDEF}(H_g), \tag{132}$$

where g ranges over the hyperedge set of G. (In fact, it suffices to restrict g to maximal hyperedges.)

Example 28. Consider the case of a pairwise Markov random field, so that $\mathrm{LOCAL}(G)$ corresponds to the constraint set used in the Bethe variational problem. In this case, the maximal hyperedges of G are simply pairs (st) of nodes connected by edges. Each subhypergraph H_{st} consists of the nodes s and t joined by an edge. The semidefinite constraint set $\mathrm{SDEF}(H_{st})$ enforces the pairwise consistency of the mean parameters associated with (x_s, x_t). For instance, in the binary case, this semidefinite constraint set is enforced by a 4×4 matrix, defined in equation (127) of Example 27. The intersection of all of these constraint sets, one for each edge (s,t), is equivalent to $\mathrm{LOCAL}(G)$. $\diamondsuit$

9.4 Log-determinant relaxation

In this section, we illustrate one possible use of semidefinite constraints in approximate inference. Recall from Section 8 that a convex relaxation of the exact variational principle requires both a convex outer bound on the set of realizable mean parameters, and a concave upper bound on the entropy. The main result of this section is to show how combining a semidefinite outer bound with a Gaussian-based entropy approximation leads to a log-determinant relaxation of the exact variational principle [114].

Although the techniques described in this section can be applied more generally, for concreteness we focus on the case of a binary random vector $\mathbf{x} \in \{0,1\}^n$, with a distribution in the Ising form

$$p(\mathbf{x};\theta) \quad = \quad \exp\big\{\sum_{s \in V} \theta_s x_s + \sum_{(s,t)} \theta_{st} x_s x_t - A(\theta)\big\}. \tag{133}$$

Without loss of generality, we assume that the underlying graph is the complete graph K_n, so that the marginal polytope of interest is $\mathrm{MARG}(K_n)$. Of course, a problem defined on an arbitrary $G = (V,E)$ can be embedded into the complete graph by setting $\theta_{st} = 0$ for all $(s,t) \notin E$.

9.4.1 Gaussian entropy bound

In order to upper bound the entropy, we return to the familiar interpretation of the Gaussian as the maximum entropy distribution subject to covariance constraints [see 23]. In particular, the differential entropy $h(\widetilde{\mathbf{x}})$ of any continuous random vector $\widetilde{\mathbf{x}}$ is upper bounded by the entropy of a Gaussian with matched covariance, or in analytical terms

$$h(\widetilde{\mathbf{x}}) \quad \leq \quad \frac{1}{2} \log\det \mathrm{cov}(\widetilde{\mathbf{x}}) + \frac{n}{2}\log(2\pi e), \tag{134}$$

where $\mathrm{cov}(\widetilde{\mathbf{x}})$ is the covariance matrix of $\widetilde{\mathbf{x}}$.

The upper bound (134) is not directly applicable to a random vector taking values in a discrete space (since differential entropy in this case diverges to minus infinity). Therefore, in order to exploit this bound for the random vector $\mathbf{x} \in \{0,1\}^n$, it is necessary to construct a suitably matched continuous version of $\mathbf{x}$. One method to do so is by the addition of an independent random vector $\mathbf{u}$, such that the delta functions in the density of $\mathbf{x}$ are smoothed out.

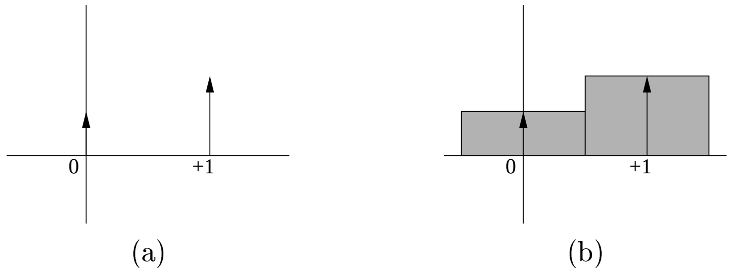


Figure 24. Illustration of the smoothing procedure. (a) Original probability mass function with impulses at $\{0, 1\}$. (b) Transformed version, where the impulses are spread out with a uniform random variable on $[-\frac{1}{2}, \frac{1}{2}]$. By construction, the (differential) entropy of the continuous random variable in (b) is equivalent to the discrete entropy of the original one in (a).

In order to do so, we use $\mathbf{x}$ to define a continuous random vector $\widetilde{\mathbf{x}} := \mathbf{x} + \mathbf{u}$, where $\mathbf{u}$ is independent of $\mathbf{x}$, with independent components distributed uniformly as $u_s \sim \mathcal{U}[-\frac{1}{2}, \frac{1}{2}]$. This construction is illustrated for the scalar case in Figure 24. A key property of this construction is that it matches the discrete entropy of $\mathbf{x}$ with the differential entropy of $\widetilde{\mathbf{x}}$.

Lemma 7. *Let h and H denote the differential and discrete entropies of $\widetilde{\mathbf{x}}$ and $\mathbf{x}$ respectively. Then $h(\widetilde{\mathbf{x}}) = H(\mathbf{x})$.*

Proof. Let $p(\cdot)$ denote the density of $\widetilde{\mathbf{x}}$ (with respect to Lebesgue measure), and let $P(\cdot)$ denote the mass function of $\mathbf{x}$ (i.e., density with respect to counting measure on $\{0,1\}^n$). Letting $\mathcal{D} := \{\widetilde{\mathbf{x}} \in \mathbb{R}^n \mid p(\widetilde{\mathbf{x}}) > 0\}$ denote the support of p, the differential entropy is given by $h(\widetilde{\mathbf{x}}) = -\int_{\mathcal{D}} p(\widetilde{\mathbf{x}}) \log p(\widetilde{\mathbf{x}}) d\widetilde{\mathbf{x}}$. By construction, $\mathcal{D}$ can be decomposed into a disjoint union of hyperboxes $\cup_{\mathbf{e}} B(\mathbf{e})$ of unit volume, one for each configuration $\mathbf{e} \in \{0,1\}^n$. Using this decomposition, h can be decomposed as

$$h(\widetilde{\mathbf{x}}) \quad = \quad -\sum_{\mathbf{e} \in \{0,1\}^n} \int_{B(\mathbf{e})} p(\widetilde{\mathbf{x}}) \log p(\widetilde{\mathbf{x}}) d\widetilde{\mathbf{x}} \stackrel{(a)}{=} -\sum_{\mathbf{e} \in \{0,1\}^n} P(\mathbf{e}) \log P(\mathbf{e}),$$

where equality (a) follows from the fact that $p(\widetilde{\mathbf{x}}) \log p(\widetilde{\mathbf{x}})$ is equal to the constant $P(\mathbf{e}) \log P(\mathbf{e})$ over each hyperbox. □

9.4.2 Log-determinant relaxation

Equipped with these building blocks, we are now ready to state a log-determinant relaxation for the log partition function. Recall the definition of SDEF_1 from equation (118).

Theorem 3. *Let $\mathbf{x} \in \{0,1\}^n$, and let $\text{OUT}(K_n)$ be any convex outer bound on $\text{MARG}(K_n)$ that is contained within $\text{SDEF}_1 \equiv \text{SDEF}_1(K_n)$. Then the log partition function $A(\theta)$ is upper bounded as follows:*

$$A(\theta) \quad \leq \quad \max_{\mu \in \text{OUT}(K_n)} \left\{ \langle \theta, \mu \rangle + \frac{1}{2} \log \det \left[M_1(\mu) + \frac{1}{12} \text{blkdiag}[0, I_n] \right] \right\} + \frac{n}{2} \log(2\pi e) \qquad (135)$$

where $\text{blkdiag}[0, I_n]$ is a $(n+1) \times (n+1)$ block-diagonal matrix.

Remarks: The inclusion $\mathrm{OUT}(K_n) \subseteq \mathrm{SDEF}_1(K_n)$ guarantees that the matrix $M_1(\mu)$ (and hence $M_1(\mu) + \frac{1}{12}\,\mathrm{blkdiag}[0, I_n]$) will always be positive semidefinite. Importantly, the optimization problem in equation (135) is a determinant maximization problem, for which efficient interior point methods have been developed [e.g., 105].

Proof of Theorem 3:
The proof is based on the variational representation of A from equation (26) of Theorem 2(b). By Theorem 1, any vector $\mu \in \mathrm{ri}\,\mathrm{MARG}(K_n)$ is realized by some distribution $p(\mathbf{x}; \theta(\mu))$. Let $\mathbf{x} \in \{0,1\}^n$ be distributed according to $p(\mathbf{x}; \theta(\mu))$. Consider the continuous-valued random vector $\widetilde{\mathbf{x}} = \mathbf{x} + \mathbf{u}$. From Lemma 7, we have $H(\mathbf{x}) = h(\widetilde{\mathbf{x}})$; combining this equality with equation (134) yields the upper bound

$$-A^*(\mu) \quad = \quad H(\mathbf{x}) \; \leq \; \frac{1}{2} \log \det \mathrm{cov}(\widetilde{\mathbf{x}}) + \frac{n}{2} \log(2\pi e). \tag{136}$$

Using the independence of $\mathbf{x}$ and $\mathbf{u}$, we can write $\mathrm{cov}(\widetilde{\mathbf{x}}) = \mathrm{cov}(\mathbf{x}) + \mathrm{cov}(\mathbf{u}) = \mathrm{cov}(\mathbf{x}) + \frac{1}{12} I_n$, where we have used the fact that $\mathrm{cov}(\mathbf{u}) = \frac{1}{12} I_n$ for the IID uniform random vector $\mathbf{u}$ on $[-1/2,\, 1/2]^n$. Combining this decomposition with equation (136) yields the upper bound

$$\begin{aligned} -A^*(\mu) \quad &\leq \quad \frac{1}{2} \log \det \left[\mathrm{cov}(\mathbf{x}) + \frac{1}{12} I_n \right] + \frac{n}{2} \log(2\pi e) \\ &= \quad \frac{1}{2} \log \det \left[M_1[\mu] + \frac{1}{12} \mathrm{blkdiag}(0, I_n) \right] + \frac{n}{2} \log(2\pi e), \end{aligned} \tag{137}$$

where the final equality follows by the Schur complement formula [54]. Finally, substituting the upper bound (137) equation (26) yields

$$\begin{aligned} A(\theta) \quad &\leq \quad \max_{\mu \in \mathrm{MARG}(K_n)} \left\{ \langle \theta,\, \mu \rangle + \frac{1}{2} \log \det \left[M_1(\mu) + \frac{1}{12} \mathrm{blkdiag}[0, I_n] \right] + \frac{n}{2} \log(2\pi e) \right\} \\ &\leq \quad \max_{\mu \in \mathrm{OUT}(K_n)} \left\{ \langle \theta,\, \mu \rangle + \frac{1}{2} \log \det \left[M_1(\mu) + \frac{1}{12} \mathrm{blkdiag}[0, I_n] \right] \right\} + \frac{n}{2} \log(2\pi e), \end{aligned}$$

where the final inequality follows because $\mathrm{OUT}(K_n)$ is an outer bound on the marginal polytope by assumption. $\square$

Remark: Just as the Bethe variational principle (75) is a tree-based approximation, the log-determinant relaxation (135) is a Gaussian-based approximation. In particular, it is worthwhile comparing the structure of the log-determinant relaxation (135) of Theorem 3 to the exact variational principle for a multivariate Gaussian, as described in Section 4.2.2. More details on the log-determinant relaxation and its performance for approximate inference can be found in the technical report [114].

10 Approximate computation of modes

The preceding sections have focused exclusively on variational methods for approximate computation of the log partition function $A(\theta)$ and mean parameters $\mu = \mathbb{E}_\theta[\boldsymbol{\phi}(\mathbf{x})]$ associated with a given density $p(\mathbf{x}; \theta)$. In this section, we turn our attention to a related but distinct problem—namely, that of computing a mode of $p(\mathbf{x}; \theta)$. It turns out that the mode problem has a variational formulation in which the set $\mathcal{M}$ once again plays a central role.

10.1 Variational formulation of computing modes

The problem of mode computation corresponds to finding a configuration $\mathbf{x}^* \in \mathcal{X}^n$ that maximizes $p(\mathbf{x};\theta)$. Note that we are assuming that at least one mode exists, so that the maximum is attained. Given the exponential form of $p(\mathbf{x};\theta)$ and the fact that the log partition function $A(\theta)$ does not depend on $\mathbf{x}$, it is equivalent to find a configuration $\mathbf{x}^* \in \arg\max_{\mathbf{x}} \langle \theta,\, \boldsymbol{\phi}(\mathbf{x}) \rangle$.

We begin by providing intuition for the more formal result to follow. Recall that the log partition function is defined as

$$A(\theta) \quad := \quad \log \int \exp\big\{\langle \theta,\, \boldsymbol{\phi}(\mathbf{x})\rangle\big\} \boldsymbol{\nu}(d\mathbf{x}), \tag{138}$$

presuming that the integral exists (i.e., $\theta \in \Theta$). Now suppose that we rescale the exponential parameter θ by some scalar $\beta > 0$. For the sake of this argument, let us assume that $\beta\theta \in \Theta$ for all $\beta > 0$. Such a rescaling will put more weight, in a relative sense, on regions of the sample space $\mathcal{X}^n$ for which $\langle \theta,\, \boldsymbol{\phi}(\mathbf{x})\rangle$ is large. Ultimately, as $\beta \to +\infty$, probability mass should remain only on configurations $\mathbf{x}^*$ in the set $\arg\max_{\mathbf{x}} \langle \theta,\, \boldsymbol{\phi}(\mathbf{x})\rangle$. This type of rescaling is equivalent to the so-called "zero-temperature limit" of statistical physics.

This intuition suggests that the behavior of the function $A(\beta\theta)$ should have a close connection to the problem of computing $\max_{\mathbf{x}} \langle \theta,\, \boldsymbol{\phi}(\mathbf{x})\rangle$. 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 and prove the following:

Theorem 4. *For all $\theta \in \Theta$, the problem of mode computation has the following alternative representations:*

$$\max_{\mathbf{x}\in\mathcal{X}^n} \langle \theta,\, \boldsymbol{\phi}(\mathbf{x})\rangle \quad \overset{(a)}{=} \quad \sup_{\mu \in \mathrm{cl}\,\mathcal{M}} \langle \theta,\, \mu\rangle \quad \overset{(b)}{=} \quad \lim_{\beta\to+\infty} \frac{A(\beta\theta)}{\beta}. \tag{139}$$

Moreover, if $\mathcal{M}$ contains no lines, then the supremum is attained at an extreme point of $\mathcal{M}$.

Proof. As pointed out earlier, the problem $\max_{\mathbf{x}\in\mathcal{X}^n} \langle \theta,\, \boldsymbol{\phi}(\mathbf{x})\rangle$ is equivalent to computing a mode for the exponential family member $p(\mathbf{x};\theta)$.

Equality (a): Let $\mathcal{P}$ be the space of all densities $p(\cdot)$, taken with respect to $\boldsymbol{\nu}$. On one hand, for any $p \in \mathcal{P}$, we have $\int \langle \theta,\, \boldsymbol{\phi}(\mathbf{x})\rangle p(\mathbf{x})\boldsymbol{\nu}(d\mathbf{x}) \leq \max_{\mathbf{x}\in\mathcal{X}^n} \langle \theta,\, \boldsymbol{\phi}(\mathbf{x})\rangle$, whence

$$\sup_{p\in\mathcal{P}} \int \langle \theta,\, \boldsymbol{\phi}(\mathbf{x})\rangle p(\mathbf{x})\boldsymbol{\nu}(d\mathbf{x}) \quad \leq \quad \max_{\mathbf{x}\in\mathcal{X}^n} \langle \theta,\, \boldsymbol{\phi}(\mathbf{x})\rangle. \tag{140}$$

Since the support of $\boldsymbol{\nu}$ is $\mathcal{X}^n$, equality is achieved in (140) by taking a sequence p^n converging to a delta function $\delta_{\mathbf{x}^*}(\mathbf{x})$, where $\mathbf{x}^* \in \arg\max_{\mathbf{x}} \langle \theta,\, \boldsymbol{\phi}(\mathbf{x})\rangle$. Finally, by linearity of expectation and the definition of $\mathcal{M}$, we have $\sup_{p\in\mathcal{P}} \int \langle \theta,\, \boldsymbol{\phi}(\mathbf{x})\rangle p(\mathbf{x})\boldsymbol{\nu}(d\mathbf{x}) = \sup_{\mu\in\mathcal{M}} \langle \theta,\, \mu\rangle$, which establishes equality (a).

Equality (b): By Proposition 2, the function A is lower semi-continuous. Therefore, for all $\theta \in \Theta$, the quantity $\lim_{\beta\to+\infty} A(\beta\theta)/\beta$ is equivalent to the recession function of A, which we denote by A_∞ (Corollary 8.5.2, [92]). 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 [92], the recession function of A corresponds to the support function of the effective domain of its dual. By Theorem 2, we have $\mathrm{cl}\,\mathrm{dom}\, A^* = \mathrm{cl}\,\mathcal{M}$, whence $A_\infty(\theta) = \sup_{\mathrm{cl}\,\mathcal{M}} \langle \theta,\, \mu\rangle$. Finally, the supremum is not affected by taking the closure.

To establish the last assertion, for a fixed $\theta \neq 0$, the function $\langle \theta, \mu \rangle$ is non-constant, linear and (hence) convex in μ. If the convex set $\mathcal{M}$ contains no lines, then the supremum must be attained at an extreme point (Cor. 32.3.2, [92]). $\square$

Remarks: (a) Theorem 4 shows that the problem of mode computation is equivalent to maximizing a linear function over the convex set $\mathcal{M}$. In fact, the function $A_\infty(\theta) := \sup_{\mu \in \mathcal{M}} \langle \theta, \mu \rangle$ corresponds to the *support function* of $\mathcal{M}$. It is clear that A_∞ is convex; moreover, it can be verified that its subdifferential $\partial A_\infty(\theta)$ has the form:

$$\mathcal{F}_{\mathcal{M}}(\theta) \quad := \quad \big\{\mu^* \in \operatorname{cl} \mathcal{M} \mid \langle \theta, \mu^* \rangle = \sup_{\mu \in \mathcal{M}} \langle \theta, \mu \rangle \big\}. \tag{141}$$

This set corresponds to the face of $\mathcal{M}$ that is exposed by the direction θ.
(b) On the basis of Theorem 2, it is possible to gain additional insight into why $\lim_{\beta \to +\infty} A(\beta\theta)/\beta$ is equivalent to the support function of $\mathcal{M}$. In particular, using Theorem 2, we write

$$\lim_{\beta \to +\infty} \frac{A(\beta\theta)}{\beta} \quad = \quad \lim_{\beta \to +\infty} \frac{1}{\beta} \sup_{\mu \in \mathcal{M}} \big\{ \langle \beta\theta, \mu \rangle - A^*(\mu) \big\} \;\; = \;\; \lim_{\beta \to +\infty} \sup_{\mu \in \mathcal{M}} \big\{ \langle \theta, \mu \rangle - \frac{1}{\beta} A^*(\mu) \big\}.$$

Equality (b) of Theorem 4 amounts to asserting that the order of the limit over β and the supremum over μ can be exchanged. The convexity of A^*, as exploited in the proof, provides justification for this exchange.
(c) In the particular case of discrete random vectors, the problem of finding a mode is an integer programming problem, and the set $\mathcal{M}$ is a polytope by Proposition 7. Thus, as a special case of Theorem 4, the integer programming problem $\max_{\mathbf{x}} \langle \theta, \boldsymbol{\phi}(\mathbf{x}) \rangle$ 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 $\mathcal{M}$. This type of transformation—i.e., from an integer program to a linear program over the convex hull of its solutions—is a frequently used technique in integer programming and combinatorial optimization [e.g., 9, 49, 79, 96]. We return to this multinomial case in Section 10.2.2.

Theorem 4 is essentially a result concerning the *value* of any mode (i.e., $\max_{\mathbf{x}} \langle \theta, \boldsymbol{\phi}(\mathbf{x}) \rangle$), and its link to rescaled forms of A. It is also of interest to investigate the limiting behavior of the mean parameters associated with $p(\mathbf{x}; \beta\theta)$, and their connection to the modes of $p(\mathbf{x}; \theta)$.

Corollary 2. *For any $\beta > 0$, let $\mu(\beta) := \mathbb{E}_{\beta\theta}[\boldsymbol{\phi}(\mathbf{x})]$ be the mean parameters associated with $p(\mathbf{x}; \beta\theta)$. If $p(\mathbf{x}; \theta)$ has at least one mode for all $\theta \in \Theta$, then $\mathcal{F}_{\mathcal{M}}(\theta)$ is non-empty for all $\theta \in \Theta$. Moreover, for all $\epsilon > 0$, there exists β_ϵ such that for all $\beta \geq \beta_\epsilon$,*

$$\mu(\beta) \quad \in \quad \big[\mathcal{F}_{\mathcal{M}}(\theta) + \mathcal{B}(0; \epsilon)\big], \tag{142}$$

where $\mathcal{B}(0; \epsilon)$ is an ϵ-ball around zero in $\mathbb{R}^d$. In the special case that $p(\mathbf{x}; \theta)$ has a unique mode $\mathbf{x}^*$, then $\mathcal{F}_{\mathcal{M}}(\theta) = \{\mu_{\mathbf{x}^*}\}$ where $\mu_{\mathbf{x}^*} := \boldsymbol{\phi}(\mathbf{x}^*)$, and $\lim_{\beta \to +\infty} \|\mu(\beta) - \mu_{\mathbf{x}^*}\| = 0$.

Proof. For each $\beta > 0$, define the function $A_\beta(\theta) := \frac{1}{\beta} A(\beta\theta)$. From Theorem 4, the sequence of functions $\{A_\beta\}$ converges to A_∞ pointwise on Θ. By Proposition 2, for each fixed $\beta < +\infty$, A_β is differentiable, and $\nabla A_\beta(\theta) = \mathbb{E}_{\beta\theta}[\boldsymbol{\phi}(\mathbf{x})]$ by the chain rule. By straightforward computations, the subdifferential of the support function $A_\infty(\theta)$ is the set $\mathcal{F}_{\mathcal{M}}(\theta)$, so that equation (142) follows from Theorem 24.5 of Rockafellar [92]. $\square$

The interpretation of Corollary 2 is quite intuitive: it guarantees that for $\beta > 0$ sufficiently large, the unique optimizer of the problem $A_\beta(\theta) = \sup_{\mu \in \mathcal{M}} \left\{ \langle \theta, \mu \rangle - \frac{1}{\beta} A^*(\mu) \right\}$ is close to the set of optimizers of the problem $A_\infty(\theta) = \sup_{\mu \in \mathrm{cl}\, \mathcal{M}} \langle \theta, \mu \rangle$. In principle, then, one could imagine attempting to compute $A_\infty(\theta)$ by computing $A_\beta(\theta)$ for an increasing sequence of β. Such a strategy can be viewed as a deterministic analog of simulated annealing [91].

10.2 Exact mode computation by variational principle

In this section, we illustrate Theorem 4 with some examples where the support function can be computed, and modes can be found exactly. To parallel our discussion in Section 4, we focus in particular on the Gaussian case, and then the multinomial case. As with the computation of mean parameters, these exact cases serve as building blocks for convex relaxations of the exact principle in more challenging cases.

10.2.1 Gaussian case

Recall our parameterization of a multivariate Gaussian random vector (of length n) on the complete graph, as presented in Sections 4.1.2 and 4.2.2. There are a total of $d = n + \binom{n}{2}$ exponential and mean parameters, one for each node and edge in the graph. It is convenient to represent the exponential and mean parameters by a pair of $(n+1) \times (n+1)$ matrices, defined as follows:

$$U(\theta) \;:=\; \begin{bmatrix} 0 & z^T(\theta) \\ z(\theta) & Z(\theta) \end{bmatrix}, \qquad\qquad W(\mu) \;:=\; \begin{bmatrix} 1 & z^T(\mu) \\ z(\mu) & Z(\mu) \end{bmatrix}. \tag{143}$$

Here $z(\mu) := [\mu_1, \mu_2, \ldots, \mu_n]^T$ is the n-vector of means, whereas $Z(\mu) = [\mu_{st}]$ is the $n \times n$ matrix of second-order moments. The analogous blocks of $U(\theta)$ are filled with the corresponding exponential parameters. Recall from Example 4 that $\Theta = \{\theta \in \mathbb{R}^d \mid Z(\theta) \prec 0\}$, whereas from Proposition 6, the set of realizable mean parameters is given by $\mathcal{M}_{Gauss} = \{\mu \in \mathbb{R}^d \mid W(\mu) \succ 0\}$. For two symmetric matrices B and C, let $\langle\!\langle B,\, C \rangle\!\rangle := \mathrm{trace}(BC)$ be the Frobenius inner product.

Semidefinite programs [104] entail maximizing a linear function subject to linear matrix inequalities (see Section 9). In the Gaussian case, the support function representation of Theorem 2 turns out to be a semidefinite program. Using Proposition 6, the constraint set of $\mathrm{cl}\, \mathcal{M}_{Gauss}$ is characterized by the linear matrix inequality $W(\mu) \succeq 0$. By the Schur complement formula [54], the LMI constraint holds if and only if $Z(\mu) - z(\mu) z^T(\mu) \succeq 0$. The cost function $\langle\!\langle U(\theta),\, W(\mu) \rangle\!\rangle$ is linear in μ, so that the overall problem is a semidefinite program (SDP).

We claim that for all $\theta \in \Theta$, this SDP has the unique optimal solution

$$z(\mu^*) \;=\; -[Z(\theta)]^{-1} z(\theta), \qquad\qquad Z(\mu^*) \;=\; z(\mu^*) z^T(\mu^*), \tag{144}$$

where $z(\mu^*) \equiv \mathbf{x}^*$ for some $\mathbf{x}^* \in \mathbb{R}^d$. The interpretation is that μ^* is realized by a Gaussian with zero covariance that places all its mass on the point $\mathbf{x}^*$. Note that the form of $\mathbf{x}^* \equiv z(\mu^*)$ coincides with the familiar expression for the mode of a Gaussian. Moreover, the optimal solution lies at an extreme point of $\mathcal{M}_{Gauss}$, which is consistent with the last assertion in Theorem 4. Figure 25 provides a geometric illustration of the result in the case $n = 1$, for which the set $\mathrm{cl}\, \mathcal{M}_{Gauss}$ is a parabola.

To establish the claim summarized in equation (144), we begin by noting that the cone of symmetric positive semidefinite matrices is self-dual [15]; hence, $B \succeq 0$ if and only if $\langle\!\langle B,\, C \rangle\!\rangle \geq 0$ for all $C \succeq 0$. Applying this fact with the choices $B := Z(\mu) - z(\mu) z^T(\mu) \succeq 0$ and $C := -Z(\theta) \succ 0$

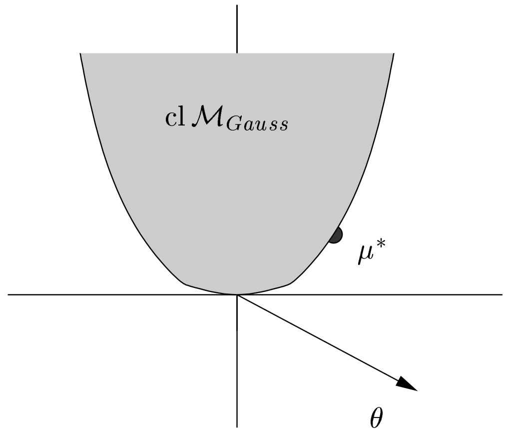


Figure 25. Illustration of the geometry of optimizing over the set $\mathrm{cl}\,\mathcal{M}_{Gauss}$ For $n = 1$, the set $\mathrm{cl}\,\mathcal{M}_{Gauss} = \{(\mu_1, \mu_2) \mid \mu_2 - \mu_1^2 \geq 0\}$. The optimum will always be attained at a boundary point of $\mathrm{cl}\,\mathcal{M}_{Gauss}$, for which $\mu_2 - \mu_1^2 = 0$, corresponding to a Gaussian with zero variance concentrated on μ_1.

yields that $\langle\langle Z(\theta),\ Z(\mu)\rangle\rangle \leq \langle\langle Z(\theta),\ z(\mu)z^T(\mu)\rangle\rangle$. Using this bound, we can write

$$\begin{aligned} \langle\langle U(\theta),\ W(\mu)\rangle\rangle &= 2\langle z(\theta),\, z(\mu)\rangle + \langle\langle Z(\theta),\ Z(\mu)\rangle\rangle \\ &\leq 2\langle z(\theta),\, z(\mu)\rangle + \langle\langle Z(\theta),\ z(\mu)z^T(\mu)\rangle\rangle. \end{aligned} \tag{145}$$

Observe that this upper bound (145) is simply a quadratic program in $z(\mu)$, with its maximum attained at $z(\mu^*) := -[Z(\theta)]^{-1}z(\theta)$. Thus, if we take the supremum over $\mu \in \mathrm{cl}\,\mathcal{M}_{Gauss}$, the bound will be met with equality, and attained at a point $W(\mu^*)$ of the form given in equation (144).

In practice, of course, one would not compute the mode of a Gaussian problem via this semidefinite formulation. However, this formulation provides valuable perspective for semidefinite relaxations of integer programming problems, as discussed in Section 10.3.3.

10.2.2 Multinomial case

Recall from Section 4.1.3 that in the finite discrete case, the set of realizable mean parameters $\mathcal{M}$ is a polytope, meaning that it is bounded and can be characterized by a finite number of linear inequality constraints. Throughout this section, we use the canonical overcomplete representation (38), so that mean parameters correspond to particular values of marginal distributions. We use $\mathrm{MARG}(G)$ to denote the set of realizable marginals associated with potentials on the cliques of G, which we refer to as a marginal polytope.

Since $\mathrm{MARG}(G)$ is a polytope, the support function representation (139) for computing modes reduces to a linear program (LP). As such, it has a particular geometry, which provides more intuition into the variational representation of Theorem 4. Figure 26 illustrates the geometry of optimizing over a marginal polytope. Extreme points of the marginal polytope are all of the form $\mu_{\mathbf{e}} = \boldsymbol{\phi}(\mathbf{e})$, for some configuration $\mathbf{e} \in \mathcal{X}^n$. The vector θ specifies a direction in the space. In order to maximize $\langle\theta,\, \mu\rangle$ over $\mathrm{MARG}(G)$, we translate a hyperplane with normal θ outwards until it is tangent to $\mathrm{MARG}(G)$. An important result in linear programming [9] is that this tangency, while it may occur at multiple points, will always involve at least one vertex of the polytope $\mathrm{MARG}(G)$. In Figure 26(a), the tangency occurs at a single vertex $\mu_{\mathbf{e}^*}$, so that $\mathbf{e}^* \in \mathcal{X}^n$ is the unique MAP configuration for the problem. In panel (b), the tangency occurs along a higher-order face of the

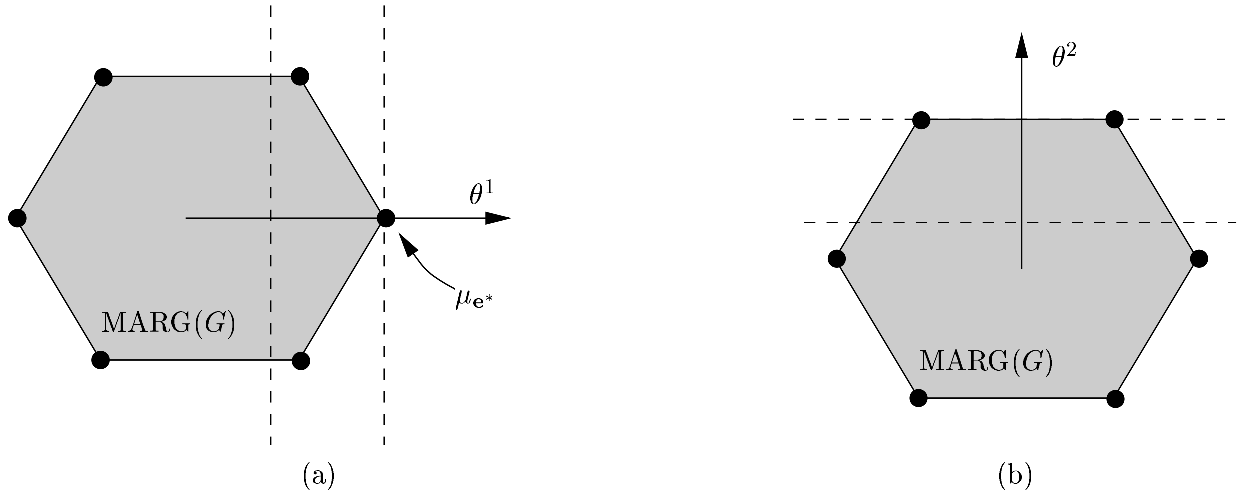


Figure 26. Geometry of optimizing over the marginal polytope MARG(G). The vector θ^i specifies the cost direction; the hyperplane with this normal is translated until it is tangent to MARG(G). (a) For the cost direction θ^1, the tangency occurs uniquely at the vertex $\mu_{\mathbf{e}^*}$, in which case $\mathbf{e}^* \in \mathcal{X}^n$ is the unique global optimum. (b) For θ^2, the tangency occurs along a higher-order face, in which case all points in the face are global optima.

polytope, and any vertex in the face will be a MAP solution. In either case, the optimal solution to the LP will be attained at a vertex of MARG(G).

Tree-structured case: As discussed in Section 4.1.3, the nature of MARG(G) depends strongly on the nature of the underlying graph G. To build on Example 8, we return to the case of a tree-structured graph $T = (V, E(T))$. Let $\mu_s(x_s)$ and $\mu_{st}(x_s, x_t)$ denote a set of marginal functions (see equation (40)) associated with the nodes and edges of T. Similarly, we also define functions of the exponential parameters as follows:

$$\theta_s(x_s) \;:=\; \sum_{j \in \mathcal{X}_s} \theta_{s;j} \mathbb{I}_j(x_s), \qquad \theta_s(x_s) \;:=\; \sum_{(j,k) \in \mathcal{X}_s \times \mathcal{X}_t} \theta_{st;jk} \mathbb{I}_{jk}(x_s, x_t).$$

In Example 8, we proved that the marginal polytope MARG(T) is characterized by the following set of local constraints:

$$\text{LOCAL}(T) \;:=\; \{\mu \geq 0 \mid \sum_{x_s} \mu_s(x_s) = 1, \qquad \sum_{x_t} \mu_{st}(x_s, x_t) = \mu_s(x_s)\}.$$

Applying Theorem 4, we conclude that finding the mode of a tree-structured problem is equivalent to solving the linear program:

$$\max_{\mu \in \text{LOCAL}(T)} \langle \mu, \theta \rangle \;=\; \max_{\mu \in \text{LOCAL}(T)} \big\{ \sum_{s \in V} \sum_{x_s} \mu_s(x_s) \theta_s(x_s) + \sum_{(s,t) \in E(T)} \sum_{x_s, x_t} \mu_{st}(x_s, x_t) \theta_{st}(x_s, x_t) \big\}. \tag{146}$$

Letting m denote $\max_s |\mathcal{X}_s|$, it can be seen that LOCAL(T) involves $\mathcal{O}(mn + m^2|E|) = \mathcal{O}(m^2 n)$ constraints. Presuming that m is not overly large, the LP of equation (146) is easily solvable by standard methods, including the simplex algorithm [9].

Of interest here is the connection between the variational problem (146) and the iterative max-product algorithm described in Section 2.5.1. Recall that the max-product algorithm is based on passing "messages", denoted by $M_{ts}(x_s)$, between nodes in the tree. These messages are updated according to the following recursion:

$$M_{ts}(x_s) \quad = \quad \kappa \max_{x_t \in \mathcal{X}_t} \Big[\exp\left\{\theta_{st}(x_s, x_t) + \theta_t(x_t)\right\} \prod_{u \in \mathcal{N}(t) \backslash s} M_{ut}(x_t) \Big]. \tag{147}$$

Note that equation (147) is the analog of the sum-product update (80) given in the proof of Proposition 11, with the summation replaced by maximization.

To understand why such a connection should exist, recall that for tree-structured problems, the exact variational principle of Theorem 2 has a concrete and tractable formulation (49), one which involves LOCAL(T) as the constraint set (see Section 4.2.3). An immediate corollary of

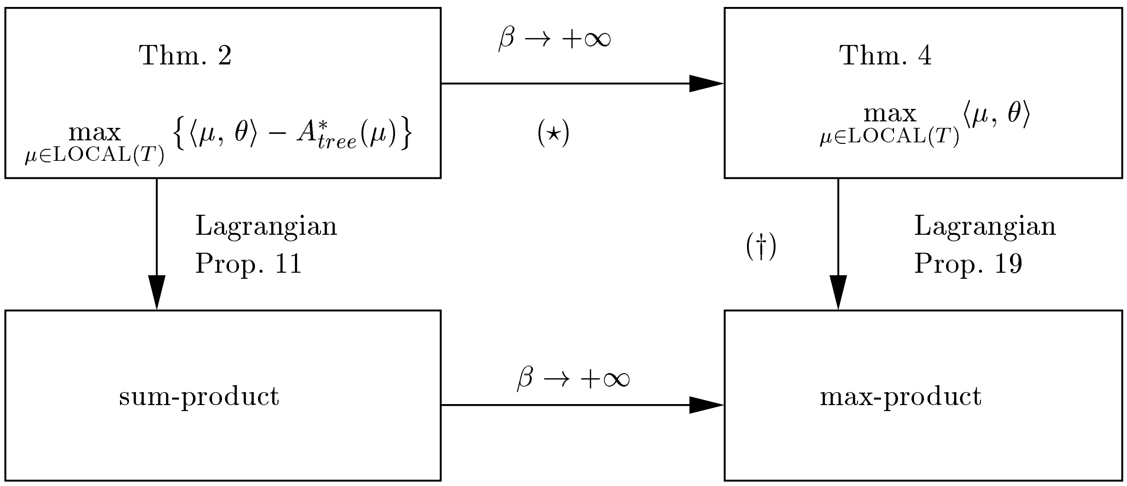


Figure 27. Block diagram of the relationships between variational principles and associated message-passing algorithms. In the tree-structured case, all the implications indicated by arrows are valid. For general graphs with cycles, implication $(\dagger)$ breaks down.

Proposition 11 is that the sum-product algorithm on trees is an iterative method for solving a Lagrangian formulation of this problem. These results and their interconnection are shown in the two left-side boxes in the block diagram of Figure 27. Next, as a special case of Theorem 4, the tree-structured linear program (146) can be obtained by taking "zero-temperature limit" of the tree-structured variational principle (49). In particular, this limiting process is described in remark (c) following the proof of Theorem 4. This implication is denoted by $(\star)$ in Figure 27.

Overall, this intuition suggests that the max-product algorithm on trees should be related to the tree-structured LP (146), which the following result [111] makes precise:

Proposition 19. *For each $x_s \in \mathcal{X}_s$, let $\lambda_{st}(x_s)$ be a Lagrange multiplier associated with the constraint $C_{ts}(x_s) = 0$, where $C_{ts}(x_s) := \tau_s(x_s) - \sum_{x_t} \tau_{st}(x_s, x_t)$. Let N be the set of τ that are non-negative and appropriately normalized:*

$$N \quad := \quad \{\tau \geq 0 \mid \sum_{x_s} \tau_s(x_s) = 1, \; \sum_{x_s, x_t} \tau_{st}(x_s, x_t) = 1 \,\}. \tag{148}$$

Consider the dual function $\mathcal{Q}$ defined by the following partial Lagrangian formulation of the tree-

structured LP (146)*:*

$$\mathcal{Q}(\lambda) \quad := \quad \max_{\tau \in N} \mathcal{L}(\tau; \lambda), \tag{149a}$$

$$\mathcal{L}(\tau; \lambda) \quad := \quad \langle \theta, \, \tau \rangle + \sum_{(s,t) \in E(T)} \big[\sum_{x_s} \lambda_{ts}(x_s) C_{ts}(x_s) + \sum_{x_t} \lambda_{st}(x_t) C_{st}(x_t) \big]. \tag{149b}$$

For any fixed point M^* of the max-product updates (147)*, 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 converting to a directed tree by first designating some node $r \in V$ 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 $\langle \theta, \, \tau \rangle$ has the alternative decomposition:

$$\sum_{x_r} \tau_r(x_r) \theta_r(x_r) + \sum_{t \to s} \sum_{x_t, x_s} \tau_{st}(x_s, x_t) \big[\theta_{st}(x_s, x_t) + \theta_s(x_s) \big].$$

With this form of the cost function, the dual function can be put into the form:

$$\mathcal{Q}(\lambda) \; := \; \max_{\tau \in N} \Big\{ \sum_{x_r} \tau_r(x_r) \nu_s(x_s) + \sum_{t \to s} \sum_{x_t, x_s} \tau_{st}(x_s, x_t) \big[\nu_{st}(x_s, x_t) - \nu_t(x_t) \big] \Big\}, \tag{150}$$

where the quantities ν_s and ν_{st} are defined in terms of λ and θ as:

$$\nu_t(x_t) \quad = \quad \theta_t(x_t) + \sum_{u \in \mathcal{N}(t)} \lambda_{ut}(x_t) \tag{151a}$$

$$\nu_{st}(x_s, x_t) \quad = \quad \theta_{st}(x_s, x_t) + \theta_s(x_s) + \theta_t(x_t) + \sum_{u \in \mathcal{N}(s) \backslash t} \lambda_{us}(x_s) + \sum_{u \in \mathcal{N}(t) \backslash s} \lambda_{ut}(x_t). \tag{151b}$$

Taking the maximum over $\tau \in N$ in equation (150) yields the explicit form for the dual function $\mathcal{Q}(\lambda) = \max_{x_r} \nu_r(x_r) + \sum_{t \to s} \max_{x_s, x_t} [\nu_{st}(x_s, x_t) - \nu_t(x_t)]$.

Any vector of messages M in the max-product algorithm defines 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 (147) if and only if the associated ν_s^* and ν_{st}^*, as defined by $\lambda^* := \log M^*$, satisfy the *edgewise consistency* condition $\max_{x_s} \nu_{st}^*(x_s, x_t) = \nu_t^*(x_t) + C_{st}$ for all $x_t \in \mathcal{X}_t$, where C_{st} is a constant independent of $\mathbf{x}$. We now show that any such λ^* is a dual optimal solution.

Under the edgewise consistency condition on a tree-structured graph, we can always find at least one configuration $\mathbf{x}^*$ that satisfies

$$x_s^* \in \arg\max_{x_s} \nu_s^*(x_s) \quad \forall \, s \in V, \qquad\qquad (x_s^*, x_t^*) \in \arg\max_{x_s, x_t} \nu_{st}^*(x_s, x_t) \quad \forall \, (s,t) \in E$$

The edgewise consistency condition also guarantees the following equalities:

$$\max_{x_s, x_t} [\nu_{st}^*(x_s, x_t) - \nu_t^*(x_t)] \quad = \quad \max_{x_t} [\nu_t^*(x_t) + C_{st} - \nu_t^*(x_t)] \; = \; C_{st} \; = \; \nu_{st}^*(x_s^*, x_t^*) - \nu_t^*(x_t^*).$$

Combining these two relations yields the following expression for the dual value at λ^*:

$$\mathcal{Q}(\lambda^*) \quad = \quad \nu_r^*(x_r^*) + \sum_{t \to s} \big[\nu_{st}^*(x_s^*, x_t^*) - \nu_t^*(x_t^*) \big] \quad \stackrel{(a)}{=} \quad \theta_r(x_r^*) + \sum_{t \to s} \; \big[\theta_{st}(x_s^*, x_t^*) + \theta_s(x_s^*) \big], \tag{152}$$

where equality (a) follows by applying the definition of $\{\nu_s^*, \nu_{st}^*\}$ from equation (151) and simplifying. (The Lagrange multipliers λ^* all cancel out in this simplification.)

Now consider the primal solution defined by $\tau_s^*(x_s) := \mathbb{I}_{x_s^*}[x_s]$ and $\tau_{st}^*(x_s, x_t) = \mathbb{I}_{x_s^*}[x_s]\,\mathbb{I}_{x_t^*}[x_t]$, where $\mathbb{I}_{x_s^*}[x_s]$ is an indicator function for the event $\{x_s = x_s^*\}$. It is clear that τ^* is primal feasible; moreover, the primal cost is equal to

$$\sum_{x_r} \tau_r^*(x_r)\theta_r(x_r) + \sum_{t \to s} \sum_{x_t, x_s} \tau_{st}^*(x_s, x_t)\big[\theta_{st}(x_s, x_t) + \theta_s(x_s)\big] \;\; = \;\; \theta_r(x_r^*) + \sum_{t \to s} \big[\theta_{st}(x_s^*, x_t^*) + \theta_s(x_s^*)\big],$$

which is precisely equal to $\mathcal{Q}(\lambda^*)$. Therefore, by strong duality for linear programs [9], the pair (τ^*, λ^*) is primal-dual optimal. $\square$

Remark: A careful examination of the proof of Proposition 19 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 will discuss in the following section.

10.3 Relaxations of the exact principle

We now consider relaxations of the exact variational principle of Theorem 4. The development of this section is specialized to the multinomial case, for which the set of realizable mean parameters is a marginal polytope $\mathrm{MARG}(G)$.

10.3.1 Relaxations from zero-temperature limits

In remark (b) following Theorem 4, we discussed how the support function representation of computing modes arises as a zero-temperature limit of the variational principle from Theorem 2(b). In analogy to this result, we begin by showing how taking the zero-temperature limit of any convex relaxation for inference leads to a corresponding relaxation for MAP estimation.

Proposition 20. *Consider a relaxation for computing approximate mean parameters based on the variational problem*

$$B(\theta) \;\; := \;\; \max_{\tau \in \mathrm{OUT}(G)} \big\{\langle \theta,\, \tau\rangle - B^*(\tau)\big\}, \tag{153}$$

where $\mathrm{OUT}(G)$ *is a compact and convex outer bound on* $\mathrm{MARG}(G)$*, and* B^* *is a convex approximation to the dual function* A^**. In the zero-temperature limit, we obtain the following relaxation for approximate mode computation:*

$$\max_{\mu \in \mathrm{MARG}(G)} \langle \theta,\, \mu\rangle \;=\; A_\infty(\theta) \;\;\leq\;\; B_\infty(\theta) \;\;:=\;\; \max_{\tau \in \mathrm{OUT}(G)} \langle \theta,\, \tau\rangle. \tag{154}$$

Proof. In the finite discrete case, we have $\operatorname{dom} A = \mathbb{R}^d$; moreover, by the compactness of $\mathrm{OUT}(G)$, we also have $\operatorname{dom} B = \mathbb{R}^d$. Consequently, the respective recession functions are defined for all $\theta \in \mathbb{R}^d$ by the following limits:

$$A_\infty(\theta) \;:=\; \lim_{\beta \to +\infty} \frac{A(\beta\theta)}{\beta}, \quad \text{and} \qquad B_\infty(\theta) \;:=\; \lim_{\beta \to +\infty} \frac{B(\beta\theta)}{\beta}.$$

By Theorem 4, we have $A_\infty(\theta) = \sup_{\mu \in \mathrm{MARG}(G)}\langle \theta,\, \mu\rangle$. Moreover, observe that B, as defined in equation (153), can be interpreted as the conjugate dual to the extended real-valued function B^*, where $B^*(\tau) := +\infty$ for $\tau \notin \mathrm{OUT}(G)$. With this definition, the (effective) domain of B^* is simply the set $\mathrm{OUT}(G)$. As a conjugate function, B is lower semi-continuous; therefore, by Theorem 13.3 of Rockafellar [92], the recession function of B is given by the support function of $\operatorname{dom} B^*$; in analytical terms, we have $B_\infty(\theta) = \max_{\tau \in \mathrm{OUT}(G)} \langle \theta, \tau \rangle$. The bound $A_\infty(\theta) \leq B_\infty(\theta)$ follows because $\mathrm{OUT}(G)$ is an outer bound on $\mathrm{MARG}(G)$ by assumption. $\square$

Remark: Of course, the inequality $\max_{\mu \in \mathrm{MARG}(G)} \langle \theta, \mu \rangle \leq \max_{\tau \in \mathrm{OUT}(G)} \langle \theta, \tau \rangle$ could be obtained more directly, simply by observing that $\mathrm{OUT}(G)$ is a convex outer bound on $\mathrm{MARG}(G)$. Nonetheless, it is interesting to obtain it as a zero-temperature limit of a corresponding convex relaxation for computing mean parameters. It should also be noted that the proof of Proposition 20 relies on the convexity of A^*. For instance, it does not apply directly to the Bethe entropy approximation or any of its non-convex extensions.

Proposition 20 gives a straightforward way to transform any convex relaxation for computing approximate mean parameters into a corresponding relaxation for approximate mode computation. We illustrate in the following sections with a number of examples.

10.3.2 Linear programming relaxations

We begin by considering linear programming (LP) relaxations of the exact principle, wherein the exact marginal polytope $\mathrm{MARG}(G)$ is replaced by an outer bound formed entirely of linear constraints. For various classes of problems in combinatorial optimization, such LP relaxations have been studied extensively; see the books [49, 79] for further details.

The case of a pairwise Markov random field suffices to illustrate the basic notion of an LP relaxation. It is convenient to use the canonical overcomplete representation based on indicator functions, as defined in equation (38) of Section 4.1.3. The constraint set $\mathrm{LOCAL}(G)$, first discussed in Example 8, constitutes an outer bound on $\mathrm{MARG}(G)$. Recall from Example 14 that it is a strict outer bound on $\mathrm{MARG}(G)$, unless G is actually tree-structured. The set $\mathrm{LOCAL}(G)$ specifies the following relaxation of mode computation for a multinomial distribution defined a pairwise Markov random field:

$$\max_{\mathbf{x} \in \mathcal{X}^n} \langle \theta, \boldsymbol{\phi}(\mathbf{x}) \rangle \quad = \quad \max_{\mu \in \mathrm{MARG}(G)} \langle \theta, \mu \rangle \quad \leq \quad \max_{\tau \in \mathrm{LOCAL}(G)} \langle \theta, \mu \rangle. \tag{155}$$

This relaxation can also be derived as a Lagrangian dual formulation of finding the tightest upper bound on the support function $A_\infty(\theta) = \max_{\mathbf{x} \in \mathcal{X}^n} \langle \theta, \boldsymbol{\phi}(\mathbf{x}) \rangle$ based on a convex combination of trees [111].

Since the relaxed constraint set $\mathrm{LOCAL}(G)$ (like $\mathrm{MARG}(G)$) is a polytope, the relaxation on the RHS of equation (155) is a linear program. Consequently, by standard properties of linear programs [9], the relaxed optimum must be attained at a vertex (possibly more than one) of the polytope $\mathrm{LOCAL}(G)$. We say that a vertex of $\mathrm{LOCAL}(G)$ is *integral* if all of its components are zero or one, and *fractional* otherwise. The following result characterizes the vertices of $\mathrm{LOCAL}(G)$:

Proposition 21. *All vertices of* $\mathrm{MARG}(G)$ *are also vertices of the relaxed polytope* $\mathrm{LOCAL}(G)$. *In addition, when G is not tree-structured, then* $\mathrm{LOCAL}(G)$ *includes additional fractional vertices that lie outside of* $\mathrm{MARG}(G)$.

Proof. In the canonical overcomplete representation of a multinomial ($\mathcal{X} = \{0, 1, \ldots, m-1\}$) on a pairwise MRF, the polytope $\mathrm{LOCAL}(G)$ lies within $\mathbb{R}^d$, where $d = mn + m^2|E|$. The set $\mathrm{LOCAL}(G)$ is defined by the d inequality constraints $\mu_\alpha \geq 0$ for all $\alpha \in \mathcal{I}$, and the normalization and marginalization equality constraints (see equation (41)). By Proposition 7, every vertex of

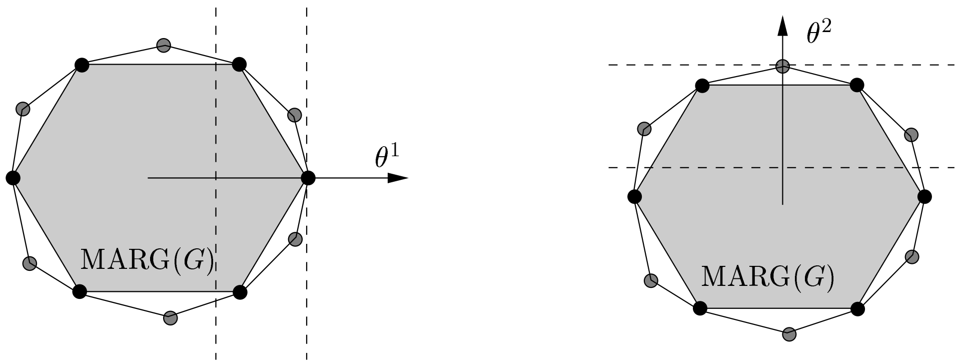


Figure 28. The constraint set LOCAL(G) is an outer bound on the exact marginal polytope. Its vertex set includes all the vertices of MARG(G), which are in one-to-one correspondence with optimal solutions of the integer program. It also includes additional fractional vertices, which are *not* vertices of MARG(G).

MARG(G) is of the form μ_J for some configuration $J \in \mathcal{X}^n$. This vector has components $[\mu_J]_s(x_s) = \mathbb{I}_{j_s}(x_s)$, and $[\mu_J]_{st}(x_s, x_t) = \mathbb{I}_{j_s}(x_s)\mathbb{I}_{j_t}(x_t)$. To show that μ_J is also a vertex of LOCAL(G), it suffices [9] to show that there are d constraints of LOCAL(G) that are active at μ_J and linearly independent. For any $J \in \mathcal{X}^n$, we have $\mathbb{I}_k(x_s) = 0$ for all $k \in \mathcal{X}\backslash\{j_s\}$, and $\mathbb{I}_k(x_s)\mathbb{I}_l(x_t) = 0$ for all $(k,l) \in (\mathcal{X} \times \mathcal{X})\backslash\{j_s, j_t\}$. All of these active inequality constraints are linearly independent, and there are a total of $d' = (m-1)n + (m^2-1)|E|$. All of the normalization and marginalization constraints are also satisfied by μ_J, but not all of them are linearly independent (when added to the active inequality constraints). However, we can add the normalization constraints for each $s = 1, \ldots, n$ and for each $(s,t) \in E$, while still preserving linear independence. Adding these $n + |E|$ equality constraints to the d' inequality constraints yields a total of d linearly independent constraints of LOCAL(G) that are satisfied by μ_J, so that it is a vertex. Note that each of these vertices has $0-1$ components, and so is integral.

The set LOCAL(G) is a polytope, so that it is equal to the convex hull of its vertices [92]. Moreover, it is a strict outer bound on MARG(G), so that it must contain additional vertices that are not members of MARG(G). Any such vertex must be fractional; otherwise, it could be identified with a unique configuration $J \in \mathcal{X}^n$, and hence would belong to MARG(G) by Proposition 7. $\square$

The distinction between fractional and integral vertices is crucial, because it determines whether or not the LP relaxation (155) specified by LOCAL(G) is tight. In particular, there are only two possible outcomes to solving the relaxation:

(a) the optimum is attained at a vertex of MARG(G), in which case the upper bound in equation (155) is tight, and a mode can be obtained.

(b) the optimum is attained only at one or more fractional vertices of LOCAL(G), which lie strictly outside MARG(G). In this case, the upper bound of equation (155) is loose, and the relaxation does not output the optimal configuration.

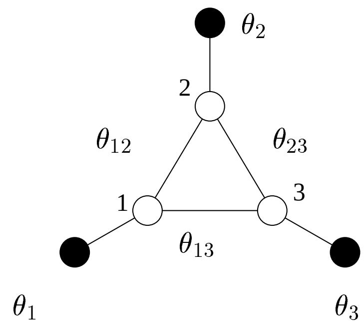
Figure 28 illustrates both of these possibilities. The vector θ^1 corresponds to case (a), in which the optimum is attained at a vertex of MARG(G). The vector θ^2 represents a less fortunate setting, in which the optimum is attained only at a fractional vertex of LOCAL(G). In simple cases, one can explicitly demonstrate a fractional vertex of the polytope LOCAL(G).

Example 29. Here we explicitly construct a fractional vertex for a binary problem $\mathbf{x} \in \{0,1\}^3$ on the complete graph K_3. Consider the exponential parameter θ shown in matrix form in Figure 29(a).

$$\theta_s = \begin{bmatrix} \theta_{s;0} & \theta_{s;1} \end{bmatrix}$$

$$\theta_{st} = \begin{bmatrix} \beta_{st} & 0 \\ 0 & \beta_{st} \end{bmatrix}$$

(a)



(b)

Figure 29. The smallest graph $G = (V, E)$ on which the relaxation (155) 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 $\theta_s, s \in V$.

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 that the relaxation (155) is tight, regardless of the choice of $\theta_s, s \in V$. 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 exponential parameter corresponds to a direction for which the relaxation (155) is *not* tight, and hence exposes a fractional vertex. First choose $\theta^*_s = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$ for $s = 1, 2, 3$, and then set $\beta_{st} = \beta < 0$ for all edges (s,t) to define θ^*_{st} via the construction in Figure 29(a). Observe that for any configuration $\mathbf{x} \in \{0,1\}^3$, we must have $x_s \neq x_t$ for at least one edge $(s,t) \in E$. Therefore, any $\mu \in \mathrm{MARG}(G)$ must place non-zero mass on at least one term of θ^* involving β, whence $A_\infty(\theta^*) = \max_{\mu \in \mathrm{MARG}(G)} \langle \theta, \mu \rangle < 0$. In fact, the optimal value $A_\infty(\theta^*)$ is exactly equal to $\beta < 0$.

On the other, consider the pseudomarginal $\tau^* \in \mathrm{LOCAL}(G)$ defined as follows:

$$\tau^*_s := \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}^T \quad \text{for } s \in V, \qquad \tau^*_{st} := \begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix} \quad \text{for } (s,t) \in E.$$

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 $\mathrm{LOCAL}(G)$. Moreover, the relaxation (155) is *not* tight because $\max_{\mu \in \mathrm{MARG}(G)} \langle \theta^*, \mu \rangle < 0$. Finally, to establish that τ^* is a vertex of $\mathrm{LOCAL}(G)$, we will 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, τ^* is a fractional vertex. $\diamondsuit$

Remarks: (a) In analogy to Proposition 11, one might postulate that Proposition 19 could be extended to graphs with cycles—specifically, that the max-product algorithm solves the dual of the

tree-based relaxation (155). This conjecture is false, since it is possible to construct problems (on graphs with cycles) for which the max-product algorithm will output a non-optimal configuration. Consequently, the max-product algorithm does not specify solutions to the dual problem, since any LP relaxation will either output a configuration with a guarantee of correctness, or a fractional vertex. Wainwright et al. [111] derive a tree-reweighted analog of the max-product algorithm, analogous to the tree-reweighted sum-product algorithm of Section 8.3. Under certain conditions, fixed points of this algorithm provably correspond to dual-optimal solutions of the tree-based relaxation (155).
(b) The tree-based relaxation (155) can be extended to hypertrees of higher width, by using the hypertree-based constraint sets $\text{LOCAL}_t(G)$ described in Section 7. This extension produces a sequence of progressively tighter LP relaxations. In the binary $\{0,1\}$ case, this sequence has been proposed previously by various researchers [50, 98], although without the connections to the underlying graphical structure.
(c) Feldman et al. [36, 37] have applied the tree-based relaxation (155) to the task of decoding turbo and low-density parity check (LDPC) codes, and provided analytical results to characterize its decoding performance. For this case of error-correcting codes, the marginal polytope is equivalent to a codeword polytope (i.e., the convex hull of all possible codewords). Moreover, the fractional vertices of the linear relaxation have a very specific interpretation as *pseudocodewords* [e.g., 39, 116, 43] of the underlying code.

10.3.3 Semidefinite relaxations for mode computation

It is also possible to develop relaxations for computing modes based on semidefinite outer approximations to $\text{MARG}(G)$, as described in Section 9. The resulting optimization problem is a *semidefinite program* [104], since it entails optimizing a linear function subject to linear matrix inequalities. Such semidefinite relaxations are widely-used in combinatorial optimization [e.g., 49, 48, 71], as well as for programs involving semialgebraic constraints more generally [e.g., 64, 84].

For the sake of brevity, we limit ourselves to describing a well-known semidefinite programming relaxation that applies to the Ising model, as described in Example 3. In particular, the problem of computing the mode of a model in Ising form is equivalent to solving the following quadratic binary integer program:

$$\max_{\mathbf{x}\in\{-1,1\}^n} \big\{ \sum_{s\in V} \theta_s x_s + \sum_{(s,t)} \theta_{st} x_s x_t \big\}. \tag{156}$$

It is convenient to use "spin" variables $\mathbf{x} \in \{-1,+1\}^n$, and to assume that the problem is formulated on the complete graph K_n.[15] By applying Theorem 4, we conclude that the the binary quadratic program (156) is equivalent to a linear program over the marginal polytope $\text{MARG}(K_n)$, represented in this case in terms of the spin variables. Since the marginal polytope is difficult to characterize, it is natural to replace it with the first-order semidefinite outer bound SDEF_1, as defined in equation (118). Doing so leads to the following semidefinite relaxation of the integer program:

$$\max_{\mu\in\text{MARG}(K_n} \langle\theta,\, \mu\rangle \quad\leq\quad \max_{\mu\in\text{SDEF}_1} \langle\theta,\, \mu\rangle, \tag{157}$$

where the RHS corresponds to a semidefinite program (SDP). Recall from our discussion of the exact computation of Gaussian modes in Section 10.2.1 the $n\times n$ matrices of exponential parameters

[15] This assumption entails no loss of generality, since a problem defined on an arbitrary graph can be put in this form by setting $\theta_{st} = 0$ for all $(s,t) \notin E$.

$U(\theta)$ and mean parameters $W(\mu)$. It is instructive to re-write the semidefinite program (SDP) in terms of these quantities as follows:

$$\max_{\mu \in \mathrm{SDEF}_1} \langle \theta, \mu \rangle \quad = \quad \frac{1}{2} \max_{W(\mu) \succeq 0, \mu_{ss}=1} \langle\langle U(\theta), \, W(\mu) \rangle\rangle. \tag{158}$$

Note that the constraints $\mu_{ss} = 1$ on the diagonal of $W(\mu)$ arise because $x_s^2 = 1$ for any spin variable $x_s \in \{-1, +1\}$. From the form of the relaxation in (158), it can be seen that the relaxation is essentially Gaussian-based. In particular, any optimal solution μ^* can be associated with the covariance matrix of a multivariate Gaussian random vector, where each element of the vector is constrained to unit variance (i.e., $\mu_{ss} = 1$).

Studying the SDP relaxation (157) in application to the MAX-CUT problem,[16] Goemans and Williamson [48] provided a random sampling scheme for generating solutions, and proved a strong guarantee on its expected performance. Although not originally described in these terms, their method can be understood as solving the SDP relaxation, thereby obtaining a solution μ^* that specifies the covariance of an optimal zero-mean Gaussian. The sampling scheme itself entails drawing a random sample from this zero-mean Gaussian, and then taking the sign of each element of this random n-vector, thereby generating an integral vector (i.e., an element of $\{-1, +1\}^n$). Goemans and Williamson proved the following remarkable fact: the expected value of a randomized solution generated in this way is *at worst* a factor $\alpha \approx 0.878$ less than the value of the optimal cut. In subsequent work, researchers have developed approximation algorithms based on semidefinite constraints for a variety of other problems [e.g., 80, 60].

11 Discussion

The core of this paper 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 mean field methods, the sum-product and max-product algorithms, 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, which in turn lead to new algorithms for approximate inference.

Many of the algorithms described in this paper are already essential tools in various practical applications (e.g., the sum-product algorithm in error-correcting coding). 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 a particular subclass of problems. For instance, it remains to be seen whether techniques used to obtain performance guarantees for relaxations of combinatorial optimization problems can be adapted to analyze other types of inference problems (e.g., computing approximate marginal distributions). Another major area with various open issues is the application of variational methods to parameter estimation. Although mean field methods are already widely used for parameter estimation in directed graphical models, open questions include

[16] The MAX-CUT problem is a particular case of the general binary quadratic program(156), in which $\theta_s = 0$ for all $s \in V$ and $\theta_{st} \leq 0$ for all $(s,t) \in E$.

how to exploit more powerful variational methods, and also how to deal with undirected graphical models. Variational methods that provide upper bounds on the cumulant generating function are likely to be useful for parameter estimation in the undirected setting.

Finally, it should be emphasized that the variational approach provides a set of techniques that are complementary to Monte Carlo methods. One interesting 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 analyze the trade-offs in complexity versus accuracy inherent to each method. It is also worthwhile pursuing the development of hybrid methods, which could combine the virtues of both variational techniques and Monte Carlo methods.

Acknowledgements

A large number of people contributed to the gestation of this paper, and it is a pleasure to acknowledge them here. The intellectual contributions and support of Alan Willsky and Tommi Jaakkola have been particularly significant in the development of the ideas presented here, and we wish to express our sincere gratitude to both of them. In addition, we thank the following individuals for their comments and insights along the way: Constantine Caramanis, Laurent El Ghaoui, Jon Feldman, G. David Forney Jr., David Karger, Adrian Lewis, Michal Rosen-Zvi, Lawrence Saul, Nathan Srebro, Sekhar Tatikonda, Romain Thibaux, Yee Whye Teh, Lieven Vandenberghe, Yair Weiss and Jonathan Yedidia.

A Proofs

A.1 Proof of Proposition 2

The proof of the results in this proposition are straightforward; see, for example, Brown [17] for more details. Lower semi-continuity follows from Fatou's lemma [93]. Interchanging the order of differentiation and integration can be justified via a standard argument using the dominated convergence theorem [93], from which derivatives can be calculated by chain rule. To establish the last claim, let θ^b be a boundary point, and let $\theta^0 \in \Theta$ be arbitrary. By the convexity and openness of Θ, the line $\theta^t := t\theta^b + (1-t)\theta^0$ is contained in Θ for all $t \in [0, 1)$ (see Thm. 6.1, [92]). Using the differentiability of A on Θ and its convexity (Corollary 1), 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 yields that $A(\theta^t) - A(\theta^0) \leq \|\theta^t - \theta^0\| \, \|\nabla A(\theta^t)\|$. Now as $t \to 1^-$, the LHS tends to infinity by the lower semi-continuity of A. Consequently, the RHS must also tend to infinity; since $\|\theta^t - \theta^0\|$ is bounded, we conclude that $\|\nabla A(\theta^t)\| \to +\infty$, as claimed.

A.2 Proof of Corollary 1

From equation (20b), the Hessian $\nabla^2 A(\theta)$ is a Gram matrix and hence must be positive semidefinite on the open set Θ, which ensures convexity (Thm. 4.3.1, [53]). If the representation is minimal, there is no vector $a \in \mathbb{R}^d$ and constant $b \in \mathbb{R}$ such that $\langle a,\, \boldsymbol{\phi}(\mathbf{x}) \rangle = b$ holds $\boldsymbol{\nu}$-almost-everywhere. This condition implies $\mathrm{var}_\theta[\langle a,\, \boldsymbol{\phi}(\mathbf{x}) \rangle] = a^T \nabla^2 A(\theta) a > 0$ for all $a \in \mathbb{R}^d$ and $\theta \in \Theta$; this strict positive definiteness of the Hessian on the open set Θ implies strict convexity [53].

A.3 Proof of Proposition 3

If the representation is not minimal, then there exists a distinct pair $\theta^1 \neq \theta^2$ for which $p(\mathbf{x};\theta^1) = p(\mathbf{x};\theta^2)$. For this pair, we have $\Lambda(\theta^1) = \Lambda(\theta^2)$, so that Λ is not one-to-one. Conversely, if the representation is minimal, then A must be strictly convex by Corollary 1. For any strictly convex function, the inequality $\langle \nabla A(\theta^1) - \nabla A(\theta^2),\, \theta^1 - \theta^2 \rangle > 0$ holds for all $\theta^1 \neq \theta^2$, which is equivalent to Λ being one-to-one.

A.4 Proof of Theorem 1

We prove the result first for a minimal representation, and then discuss its extension to the overcomplete case. By definition, a convex subset of $\mathbb{R}^d$ is *full-dimensional* if its affine hull is equal to $\mathbb{R}^d$. As shown in Proposition 5, $\mathcal{M}$ is a full-dimensional convex set whenever the representation is minimal. Consequently, we can deal with the interior (as opposed to relative interior). Our proof makes use of the following properties of a full-dimensional convex set [see 53, 92]: (a) $\operatorname{int} \mathcal{M}$ is non-empty, and $\operatorname{int}[\operatorname{cl}(\mathcal{M})] = \operatorname{int}(\mathcal{M})$; and (b) the vector $0 \in \operatorname{int}(\mathcal{M}) \iff$ for all non-zero $\gamma \in \mathbb{R}^d$, there exists some $\mu \in \mathcal{M}$ with $\langle \gamma,\, \mu \rangle > 0$.

$\underline{\Lambda(\Theta) \subseteq \operatorname{int} \mathcal{M}}$: By shifting the potential $\boldsymbol{\phi}$ by a constant vector if necessary, it suffices to consider the case $0 \in \Lambda(\Theta)$. Let $\theta^0 \in \Theta$ be the associated exponential parameter satisfying $\Lambda(\theta^0) = 0$. We prove that for all non-zero directions $\gamma \in \mathbb{R}^d$, there is some $\mu \in \mathcal{M}$ such that $\langle \gamma,\, \mu \rangle > 0$, which implies $0 \in \operatorname{int}(\mathcal{M})$ by property (b).

For any $\gamma \in \mathbb{R}^d$, the openness of Θ ensures the existence of some $\delta > 0$ such that $(\theta^0 + \delta\gamma) \in \Theta$. Using the strict convexity and differentiability of A on Θ and the fact that $\Lambda(\theta^0) = 0$ by assumption, there holds $A(\theta^0 + \delta\gamma) > A(\theta^0) + \langle \Lambda(\theta^0),\, \delta\gamma \rangle \;=\; A(\theta^0)$. Similarly, defining $\mu^\delta := \Lambda(\theta^0 + \delta\gamma)$, 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 $\mu^\delta \in \Lambda(\Theta) \subseteq \mathcal{M}$ and $\gamma \in \mathbb{R}^d$ was arbitrary, this establishes that $0 \in \operatorname{int}(\mathcal{M})$.

$\underline{\operatorname{int} \mathcal{M} \subseteq \Lambda(\Theta)}$: As in the preceding argument, we may take $0 \in \operatorname{int} \mathcal{M}$ without loss of generality. Then, we must establish the existence of $\theta \in \Theta$ such that $\Lambda(\theta) = \nabla A(\theta) = 0$. By convexity, it is equivalent to show that $\inf_{\theta \in \Theta} A(\theta)$ 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 non-zero direction $\gamma \in \mathbb{R}^d$ and $\epsilon > 0$, consider the half space $H_{\gamma,\epsilon} := \{\mathbf{x} \in \mathcal{X}^n \mid \langle \gamma,\, \boldsymbol{\phi}(\mathbf{x}) \rangle \geq \epsilon\}$. Since $0 \in \operatorname{int} \mathcal{M}$, this half-space must have positive measure under $\boldsymbol{\nu}$ for all sufficiently small $\epsilon > 0$. Otherwise, the inequality $\langle \gamma,\, \boldsymbol{\phi}(\mathbf{x}) \rangle \leq 0$ would hold $\boldsymbol{\nu}$-a.e., which implies that $\langle \gamma,\, \mu \rangle \leq 0$ for all $\mu \in \operatorname{cl}(\mathcal{M})$. By the convexity of $\mathcal{M}$, this inequality would imply that $0 \notin \operatorname{int} \operatorname{cl}(\mathcal{M}) = \operatorname{int}(\mathcal{M})$, which contradicts our starting assumption.

For an arbitrary $\theta^0 \in \Theta$, we now write

$$A(\theta^0 + t\gamma) \;\;\geq\;\; \log \int_{H_{\gamma,\epsilon}} \exp\left\{\langle \theta^0 + t\gamma,\, \boldsymbol{\phi}(\mathbf{x}) \rangle\right\} \boldsymbol{\nu}(d\mathbf{x}) \;\geq\; t\epsilon + \underbrace{\log \int_{H_{\gamma,\epsilon}} \exp\left\{\langle \theta^0,\, \boldsymbol{\phi}(\mathbf{x}) \rangle\right\} \boldsymbol{\nu}(d\mathbf{x})}_{C(\theta^0)}.$$

Note that we must have $C(\theta^0) > -\infty$, because $\exp\{\langle \theta^0,\, \boldsymbol{\phi}(\mathbf{x}) \rangle\} > 0$ for all $\mathbf{x} \in \mathcal{X}^n$, and $\boldsymbol{\nu}(H_{\gamma,\epsilon}) > 0$. 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.

Extension to overcomplete case: For any overcomplete representation ϕ, 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 Λ_φ and Λ_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, Λ_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 Λ_ϕ is onto the relative interior of $\mathcal{M}_\phi$.

A.5 Proof of Theorem 2

(a) Case (i) $\mu \in \operatorname{ri} \mathcal{M}$: In this case, Theorem 1 guarantees that the inverse image $\Lambda^{-1}(\mu)$ is nonempty. Any point in this inverse image attains the supremum in equation (23). In a minimal representation, there is only one optimizing point, whereas there is an affine subset for an overcomplete representation. Nonetheless, for any $\theta(\mu) \in \Lambda^{-1}(\mu)$, the value of the optimum is: $A^*(\mu) = \langle \theta(\mu), \mu \rangle - A(\theta(\mu))$. We conclude by observing that

$$-H(p(\mathbf{x};\theta(\mu))) \;=\; \mathbb{E}_\theta[\langle \theta(\mu), \phi(\mathbf{x}) \rangle - A(\theta(\mu))] \;=\; \langle \theta(\mu), \mu \rangle - A(\theta(\mu)).$$

Case (ii) $\mu \notin \operatorname{cl} \mathcal{M}$: 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 $\operatorname{cl} \mathcal{M} \supseteq \operatorname{dom} A^*$. From Proposition 2, the function A is essentially smooth and lower semi-continuous. From Theorem 1, we have $\nabla A(\Theta) = \operatorname{ri} \mathcal{M}$. By Corollary 26.4.1 of Rockafellar [92], these conditions guarantee that $\operatorname{ri} \operatorname{dom} A^* \subseteq \operatorname{ri} \mathcal{M} \subseteq \operatorname{dom} A^*$. Since both $\mathcal{M}$ and $\operatorname{dom} A^*$ are convex sets, taking closures in the these inclusions yields that $\operatorname{cl} \operatorname{dom} A^* = \operatorname{cl} \operatorname{ri} \mathcal{M} = \operatorname{cl} \mathcal{M}$, where the second equality follows by the convexity of $\mathcal{M}$. Therefore, by definition of the effective domain, $A^*(\mu) = +\infty$ for any $\mu \notin \operatorname{cl} \mathcal{M}$.

Case (iii) $\mu \in \operatorname{cl} \mathcal{M} \backslash \operatorname{ri} \mathcal{M}$: Since A^* is defined as a conjugate function, it is lower semi-continuous. Therefore, the value of $A^*(\mu)$ for any boundary point $\mu \in \operatorname{cl} \mathcal{M} \backslash \operatorname{ri} \mathcal{M}$ is determined by the limit over a sequence approaching μ from inside $\operatorname{ri} \mathcal{M}$, as claimed.

(b) From Proposition 2, A is lower semi-continuous, which ensures that $(A^*)^* = A$ so that we can write $A(\theta) = \sup_{\mu \in \operatorname{cl} \operatorname{dom} A^*} \{\langle \theta, \mu \rangle - A^*(\mu)\}$. Part (a) shows that $\operatorname{cl} \operatorname{dom} A^* = \operatorname{cl} \mathcal{M}$, so that equation (26) follows. Whether the supremum is taken over $\mathcal{M}$ or over $\operatorname{cl} \mathcal{M}$ is inconsequential.

A.6 Proof of Proposition 4

For a minimal representation, Proposition 3 and Theorem 1 guarantee that the gradient mapping ∇A is a bijection between Θ and $\operatorname{ri} \mathcal{M}$. On this basis, it follows that the gradient mapping ∇A^* also exists and is bijective [92], whence the supremum (32) is attained at a unique point whenever $\theta \in \Theta$. The analogous statement for an overcomplete representation can be proved via reduction to a minimal representation.

A.7 Proof of Proposition 5

(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 $\langle a, \phi(\mathbf{x}) \rangle = b$ holds $\boldsymbol{\nu}$-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) By Theorem 4 of Section 10, 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 (hence finite) on all of $\mathbb{R}^d$ (Prop. 3.2.7; [53]).

A.8 Proof of Proposition 6

By the Schur complement formula [54], the $(n+1) \times (n+1)$ matrix $W(\mu)$ is positive definite if and only if the $n \times n$ matrix $Z(\mu) - z(\mu)z^T(\mu)$ is positive definite. But this latter matrix can be interpreted as the covariance of $\mathbf{x}$. Any Gaussian random vector gives rise to a positive definite covariance. Conversely, given a matrix $W(\mu)$ that is positive definite, we can construct a Gaussian with mean $z(\mu)$ and covariance $Z(\mu) - z(\mu)z^T(\mu)$.

A.9 Proof of Proposition 7

The definition (33) shows that $\mathcal{M}$ is given by the convex hull of the vectors $\{\boldsymbol{\phi}(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}^n\}$. Regardless of the specific choice of $\boldsymbol{\phi}$, there are only a finite number of vectors $\boldsymbol{\phi}(\mathbf{x})$ in this convex hull. Therefore, the Minkowski-Weyl theorem [92, 53] guarantees that $\mathcal{M}$ has a representation as in equation (36). By definition, any $\mu \in \mathcal{M}$ has a representation of the form $\mu = \sum_{\mathbf{x}} p(\mathbf{x})\boldsymbol{\phi}(\mathbf{x})$. If μ is not of the form $\mu_{\mathbf{e}} = \boldsymbol{\phi}(\mathbf{e})$, then $p(\mathbf{x}^i) > 0$ for at least two distinct $\mathbf{x}^1, \mathbf{x}^2$. Hence, it is not an extreme point.

A.10 Proof of Proposition 8

The convexity and lower semi-continuity (l.s.c.) follow because A^* is the supremum of collection of functions linear in μ. Statements (a)—(c) are equivalent to the assertion that A^* is strictly convex and essentially smooth. Since both A and A^* are l.s.c., A^* has these properties if and only if A is strictly convex and essentially smooth (Thm. 26.3; [92]). For a minimal representation, A is strictly convex by Corollary 1, and it is essentially smooth by Proposition 2, so that the result follows.

B Affine hulls and relative interior

The interior of a convex set $C \subseteq \mathbb{R}^d$ consists of all vectors $\mathbf{z} \in C$ for which there exists some $\epsilon > 0$ such that the ϵ-ball $B_\epsilon(\mathbf{z}) := \{\mathbf{y} \in \mathbb{R}^d \mid \|\mathbf{y} - \mathbf{z}\| < \epsilon\}$ is contained within C. The relative interior is defined similarly, except that the interior is taken with respect to the affine hull of C, denoted $\operatorname{aff} C$. More formally, the *relative interior* of C, denoted $\operatorname{ri} C$, is the set of all points $\mathbf{z}$ such that for some $\epsilon > 0$, the set $B_\epsilon(\mathbf{z}) \cap \operatorname{aff}(C)$ is contained within C. 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; the affine hull of $[0,1]$ is the real line, so that the relative interior is the open interval $(0,1)$.

A key property of any convex set C is that its relative interior is always non-empty [92]. A convex set $C \subseteq \mathbb{R}^d$ is *full-dimensional* if its affine hull $\operatorname{aff} C$ is equal to $\mathbb{R}^d$. In this case, the notion of interior and relative interior coincide.

C 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 $\psi_{As}(z, x_s) := \psi_{123}(z_1, z_2, z_3)\mathbb{I}[z_s = x_s]$. 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 among the triplet $\{x_1, x_2, x_3\}$.

D Möbius inversion

This appendix provides a brief overview of the Möbius function associated with a partially-ordered set (poset); see Stanley [101] for a thorough treatment. The *zeta function* $\zeta(g, h)$ of a poset is defined as:

$$\zeta(g,h) \;=\; \begin{cases} 1 & \text{if } g \leq h \\ 0 & \text{otherwise} \end{cases} \tag{159}$$

The Möbius function ω arises as the multiplicative inverse of this zeta function. It is defined in a recursive fashion, by first specifying $\omega(g, g) = 1$ for all g. Once $\omega(g, f)$ has been defined for all f such that $g \leq f < h$, we then define:

$$\omega(g,h) = -\sum_{\{f \mid g \leq f < h\}} \omega(g,f) \tag{160}$$

With this definition, it can be seen that ω and ζ are multiplicative inverses, in the sense that

$$\sum_{\{f \mid g \leq f \leq h\}} \omega(g,f)\zeta(f,h) \;=\; \delta(g,h)$$

where $\delta(g, h)$ is the Kronecker delta.

Lemma 8 (Möbius inversion formula). *Let $\Upsilon(h)$ be a real-valued function defined for h in a poset. Define a new real-valued function Ω via:*

$$\Omega(h) \;=\; \sum_{g \in \mathcal{D}^+(h)} \Upsilon(g) \tag{161}$$

where $\mathcal{D}^+(h) := \{g \mid g \leq h\}$ is the set of descendants of h. Then we have the relation

$$\Upsilon(h) \;=\; \sum_{g \in \mathcal{D}^+(h)} \omega(g,h)\Omega(g) \tag{162}$$

where ω is the associated Möbius function.

References

[1] S. Aji and R. McEliece. The generalized distributive law. *IEEE Trans. Info. Theory*, 46:325–343, March 2000.

[2] N. I. Akhiezer. *The classical moment problem and some related questions in analysis.* Hafner Publishing Co., New York, NY, 1966.

[3] S. Amari. Differential geometry of curved exponential families — curvatures and information loss. *Annals of Statistics*, 10(2):357–385, 1982.

[4] S. Amari and H. Nagaoka. *Methods of information geometry.* American Mathematical Society, Providence, RI, 2000.

[5] O. E. Barndorff-Nielson. *Information and exponential families.* Wiley, Chichester, 1978.

[6] R. J. Baxter. *Exactly solved models in statistical mechanics.* Academic Press, New York, 1982.

[7] D. Bertsekas. *Dynamic programming and stochastic control*, volume 1. Athena Scientific, Belmont, MA, 1995.

[8] D. Bertsekas. *Nonlinear programming.* Athena Scientific, Belmont, MA, 1995.

[9] D. Bertsimas and J. Tsitsikilis. *Introduction to linear optimization.* Athena Scientific, Belmont, MA, 1997.

[10] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48(3):259–279, 1986.

[11] J. Besag and P. J. Green. Spatial statistics and Bayesian computation. *J. R. Stat. Soc. B*, 55(1):25–37, 1993.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[13] H. Bodlaender. A tourist guide through treewidth. *Acta Cybernetica*, 11:1–21, 1993.

[14] J. Borwein and A. Lewis. *Convex Analysis.* Springer-Verlag, New York, NY, 1999.

[15] S. Boyd and L. Vandenberghe. *Convex optimization.* Courses notes; to be published. Stanford University, Palo Alto, CA, 2002.

[16] 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*, 7:191–204, 1967.

[17] L. Brown. *Fundamentals of statistical exponential families.* Institute of Mathematical Statistics, Hayward, CA, 1986.

[18] C. B. Burge and S. Karlin. Finding the genes in genomic dna. *Current Opinion in Structural Biology*, 8:346–354, 1998.

[19] Y. Censor and S. A. Zenios. *Parallel Optimization: Theory, Algorithms, and Applications.* Numerical Mathematics and Scientific Computation. Oxford University Press, 1988.

[20] D. Chandler. *Introduction to modern statistical mechanics.* Oxford University Press, Oxford, 1987.

[21] S. Chopra. On the spanning tree polyhedron. *Operations Research Letters*, 8:25–29, 1989.

[22] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* MIT Press, Cambridge, MA, 1990.

[23] T. Cover and J. Thomas. *Elements of Information Theory.* John Wiley and Sons, New York, 1991.

[24] G. Cross and A. Jain. Markov random field texture models. *IEEE Trans PAMI*, 5:25–39, 1983.

[25] I. Csiszár and G. Tusnády. Information geometry and alternating minimization procedures. In E. J. D. et al., editor, *Recent results in estimation theory and related topics.* 1984.

[26] I. Csiszár. I-divergence geometry of probability distributions and minimization problems. *Annals of Probability,*, 3(1):146–158, Feb. 1975.

[27] I. Csiszár. A geometric interpretation of Darroch and Ratcliff's generalized iterative scaling. *Annals of Statistics*, 17(3):1409–1413, Sep. 1989.

[28] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, 43:1470–1480, 1972.

[29] A. P. Dawid. Applications of a general propagation algorithm for probabilistic expert systems. *Statistics and Computing*, 2:25–36, 1992.

[30] J. Demmel. *Applied numerical linear algebra.* SIAM, Philadelphia, 1997.

[31] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. Royal Stat. Soc. B*, 39:1–38, 1977.

[32] M. Deza and M. Laurent. *Geometry of cuts and metric embeddings*. Springer-Verlag, New York, 1997.
[33] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, editors. *Biological Sequence Analysis*. Cambridge University Press, Cambridge, 1998.
[34] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1:127–136, 1971.
[35] B. Efron. The geometry of exponential families. *Annals of Statistics*, 6:362–376, 1978.
[36] J. Feldman, D. R. Karger, and M. J. Wainwright. Linear programming-based decoding of turbo codes and its relation to iterative approaches. In *Proc. Allerton Conf. Communication, Control and Computing*, October 2002.
[37] J. Feldman, D. R. Karger, and M. J. Wainwright. Using linear programming to decode LDPC codes. In *Conference on Information Science and Systems*, March 2003.
[38] J. Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution*, 17:368–376, 1981.
[39] G. 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*, pages 101–112. Springer, 2001.
[40] G. D. Forney, Jr. The Viterbi algorithm. *Proc. IEEE*, 61:268–277, March 1973.
[41] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *Intl. J. Computer Vision*, 40(1):25–47, 2000.
[42] B. Frey, R. Koetter, and N. Petrovic. Very loopy belief propagation for unwrapping phase images. In *NIPS 14*. MIT Press, 2001.
[43] B. J. Frey, R. Koetter, and A. Vardy. Signal-space characterization of iterative decoding. *IEEE Trans. Info. Theory*, 47:766–781, 2001.
[44] R. G. Gallager. *Low-density parity check codes*. MIT Press, Cambridge, MA, 1963.
[45] A. E. Gelfand and A. F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409, 1990.
[46] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Pat. Anal. Mach. Intell.*, 6:721–741, 1984.
[47] Z. Ghahramani and M. Jordan. Factorial hidden Markov models. *Machine Learning*, 29:245–273, 1997.
[48] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.
[49] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*. Springer-Verlag, Berlin, Germany, 1993.
[50] P. Hammer, P. Hansen, and B. Simeone. Roof duality, complementation, and persistency in quadratic 0-1 optimization. *Mathematical Programming*, 28:121–155, 1984.
[51] M. Hassner and J. Sklansky. The use of Markov random fields as models of texture. *Comp. Graphics Image Proc.*, 12:357–370, 1980.
[52] T. Heskes, K. Albers, and B. Kappen. Approximate inference and constrained optimization. In *Uncertainty in Artificial Intelligence*, volume 13, page to appear, 2003.
[53] J. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms*, volume 1. Springer-Verlag, New York, 1993.
[54] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1985.
[55] X. Huang, A. Acero, H.-W. Hon, and R. Reddy. *Spoken Language Processing*. Prentice Hall, New York, 2001.
[56] T. S. Jaakkola. Tutorial on variational approximation methods. In M. Opper and D. Saad, editors, *Advanced mean field methods: Theory and practice*, pages 129–160. MIT Press, 2001.
[57] M. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. In *Learning in graphical models*, pages 105–161. MIT Press, 1999.
[58] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, New Jersey, 2000.
[59] H. Kappen and P. Rodriguez. Efficient learning in Boltzmann machines using linear response theory. *Neural Computation*, 10:1137–1156, 1998.
[60] D. Karger, R. Motwani, and M. Sudan. Appoximate graph coloring by semidefinite programming. In *IEEE Symposium Foundations of Computer Science*, 1994.
[61] S. Karlin and W. Studden. *Tchebycheff systems, with applications in analysis and statistics*. Interscience Publishers, New York, NY, 1966.

[62] 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*, 305(3):567–580, 2001.

[63] F. Kschischang and B. Frey. Iterative decoding of compound codes by probability propagation in graphical models. *IEEE Sel. Areas Comm.*, 16(2):219–230, February 1998.

[64] J. B. Lasserre. An explicit equivalent positive semidefinite program for nonlinear 0-1 programs. *SIAM Journal on Optimization*, 12:756–769, 2001.

[65] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[66] M. Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for 0-1 programming. *Mathematics of Operations Research*, To appear, 2002.

[67] M. Laurent. Semidefinite relaxations for max-cut. In *The Sharpest Cut: Fetschrift in Honor of M. Padberg's 60th Birthday*, page To appear, 2002.

[68] S. L. Lauritzen. *Graphical models*. Oxford University Press, Oxford, 1996.

[69] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society B*, 50:155–224, January 1988.

[70] M. Leisink and H. Kappen. A tighter bound for graphical models. In *NIPS 13*, pages 266–272. MIT Press, 2001.

[71] L. Lovasz. On the Shannon capacity of a graph. *IEEE Trans. Information Theory*, IT-25:1–8, 1979.

[72] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. Spielman. Improved low-density parity check codes using irregular graphs. *IEEE Trans. Info. Theory*, 47:585–598, February 2001.

[73] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.

[74] J. McAuliffe, L. Pachter, and M. I. Jordan. Multiple-sequence functional annotation and the generalized hidden Markov phylogeny. Technical report, Department of Statistics, University of California, 2003.

[75] R. McEliece, D. McKay, and J. Cheng. Turbo decoding as an instance of Pearl's belief propagation algorithm. *IEEE Jour. Sel. Communication*, 16(2):140–152, February 1998.

[76] R. J. McEliece and M. Yildirim. Belief propagation on partially ordered sets. In D. Gilliam and J. Rosenthal, editors, *Mathematical Theory of Systems and Networks*. Institute for Mathematics and its Applications, 2002.

[77] T. P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, MIT, January 2001.

[78] R. Neal and G. E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*. MIT Press, Cambridge, MA, 1999.

[79] G. L. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization*. Wiley-Interscience, New York, 1999.

[80] Y. Nesterov. Semidefinite relaxation and non-convex quadratic optimization. *Optimization Methods and Software*, 12:1–20, 1997.

[81] M. Opper and D. Saad. Adaptive TAP equations. In M. Opper and D. Saad, editors, *Advanced mean field methods: Theory and practice*, pages 85–98. MIT Press, 2001.

[82] P. Pakzad and V. Anantharam. Iterative algorithms and free energy minimization. In *CISS*, March 2002.

[83] G. Parisi. *Statistical field theory*. Addison-Wesley, 1988.

[84] P. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematics of Operations Research*, To appear, 2002.

[85] J. Pearl. *Probabilistic reasoning in intelligent systems*. Morgan Kaufman, San Mateo, 1988.

[86] J. S. Pedersen and J. Hein. Gene finding with a hidden Markov model of genome structure and evolution. *Bioinformatics*, 19:219–227, 2003.

[87] T. Plefka. Convergence condition of the TAP equation for the infinite-ranged Ising model. *Journal of Physics A*, 15(6):1971–1978, 1982.

[88] L. R. Rabiner and B. H. Juang. *Fundamentals of speech recognition*. Prentice Hall, Englewood Cliffs, N.J., 1993.

[89] T. Richardson and R. Urbanke. The capacity of low-density parity check codes under message-passing decoding. *IEEE Trans. Info. Theory*, 47:599–618, February 2001.
[90] B. D. Ripley. *Spatial statistics*. Wiley, New York, 1981.
[91] C. P. Robert and G. Casella. *Monte Carlo statistical methods*. Springer texts in statistics. Springer-Verlag, New York, NY, 1999.
[92] G. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, 1970.
[93] H. Royden. *Real Analysis*. Prentice-Hall, New Jersey, 1988.
[94] L. K. Saul and M. Jordan. Exploiting tractable substructures in intractable networks. In *NIPS 8*, pages 486–492. MIT Press, 1996.
[95] L. K. Saul and M. I. Jordan. Boltzmann chains and hidden Markov models. In *Advances in Neural Information Processing Systems 7*. MIT Press, Cambridge, MA, 1995.
[96] A. Schrijver. *Theory of linear and integer programming*. Wiley-Interscience Series in Discrete Mathematics, Wiley, NY, 1989.
[97] G. R. Shafer and P. P. Shenoy. Probability propagation. *Annals of Mathematics and Artificial Intelligence*, 2:327–352, 1990.
[98] 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*, 3:411–430, 1990.
[99] 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*, pages 277–286, 2003.
[100] T. P. Speed and H. T. Kiiveri. Gaussian Markov distributions over finite graphs. *Annals of Statistics*, 14(1):138–150, March 1986.
[101] R. P. Stanley. *Enumerative combinatorics*, volume 1. Cambridge University Press, Cambridge, UK, 1997.
[102] S. Tatikonda and M. I. Jordan. Loopy belief propagation and Gibbs measures. In *Proc. Uncertainty in Artificial Intelligence*, volume 18, pages 493–500, August 2002.
[103] A. Thomas, A. Gutin, V. Abkevich, and A. Bansal. Multilocus linkage analysis by blocked Gibbs sampling. *Statistics and Computing*, 10:259–269, 2000.
[104] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.
[105] L. Vandenberghe, S. Boyd, and S. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19:499–533, 1998.
[106] S. Verdú and H. V. Poor. Abstract dynamic programming models under commutativity conditions. *SIAM J. Control and Optimization*, 25(4):990–1006, July 1987.
[107] S. R. W. Gilks and D. Spiegelhalter, editors. *Markov Chain Monte Carlo in practice*. Chapman and Hall, New York, NY, 1996.
[108] M. J. Wainwright. *Stochastic processes on graphs with cycles: geometric and variational approaches*. PhD thesis, MIT, January 2002.
[109] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. In *Uncertainty in Artificial Intelligence*, volume 18, pages 536–543, August 2002.
[110] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree consistency and bounds on the max-product algorithm and its generalizations. Appeared as LIDS Tech. report, P-2554 MIT; Available online at `http://www.eecs.berkeley/edu/~wainwrig`, July 2002.
[111] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Exact MAP estimates via agreement on (hyper)trees: Linear programming and message-passing approaches. Technical report, UC Berkeley, UCB/CSD-3-1269, August 2003.
[112] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree-based reparameterization framework for analysis of sum-product and related algorithms. *IEEE Trans. Info. Theory*, 49(5):1120–1146, May 2003.
[113] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudomoment matching. In *Workshop on Artificial Intelligence and Statistics*, January 2003.
[114] M. J. Wainwright and M. I. Jordan. Semidefinite relaxations for approximate inference on graphs with cycles. Technical report, UC Berkeley, UCB/CSD-3-1226, January 2003.

[115] M. Welling and Y. Teh. Belief optimization: A stable alternative to loopy belief propagation. In *Uncertainty in Artificial Intelligence*, July 2001.

[116] N. Wiberg. *Codes and decoding on general graphs.* PhD thesis, University of Linkoping, Sweden, 1996.

[117] W. Wiegerinck. Variational approximations between mean field theory and the junction tree algorithm. In *UAI 2000*, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

[118] W. Wiegerinck and T. Heskes. Fractional belief propagation. In *NIPS*, volume 12, page to appear, 2002.

[119] A. S. Willsky. Multiresolution Markov models for signal and image processing. *Proceedings of the IEEE*, 90(8):1396–1458, 2002.

[120] J. Woods. Markov image modeling. *IEEE Transactions on Automatic Control*, 23:846–850, October 1978.

[121] J. Yedidia. An idiosyncratic journey beyond mean field theory. In M. Opper and D. Saad, editors, *Advanced mean field methods: Theory and practice*, pages 21–36. MIT Press, 2001.

[122] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS 13*, pages 689–695. MIT Press, 2001.

[123] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. Technical Report TR2001-22, Mitsubishi Electric Research Labs, January 2002.

[124] A. Yuille. CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14:1691–1722, 2002.