Chapter 5

# Tree-based reparameterization for approximate estimation

## ■ 5.1 Introduction

Given a distribution $p(\mathbf{x})$ defined by a graphical model, one important problem is computing marginal distributions of variables at each node on the graph. For tree-structured graphs, standard and highly efficient algorithms exist for this task; see Appendix A for description of one such algorithm. In contrast, exact solutions are prohibitively complex for more general graphs of any substantial size [40]. As a result, there has been considerable interest and effort aimed at developing approximate inference algorithms for large graphs with cycles.

A widely-studied approximation method [e.g., 3, 131, 147, 175, 183] is that known variously as *belief propagation* in the graphical model community [138], and the *sum-product algorithm* in coding theory [e.g., 118, 131]. The interest in this algorithm has been fueled in part by its use in fields such as artificial intelligence and computer vision [e.g., 67,68,134], and also by the success of turbo codes and other graphical codes, for which the decoding algorithm is a particular instantiation of belief propagation [e.g., 72, 118, 131]. While there are various equivalent forms for belief propagation [138], the best known formulation, which we refer to here as the synchronous BP algorithm, entails the exchange of statistical information among neighboring nodes via message-passing. If the graph is a tree, the resulting algorithm can be shown to produce exact solutions in a finite number of iterations. The message-passing formulation is thus equivalent to other techniques for optimal inference on trees, some of which involve more global and efficient computational procedures. On the other hand, if the graph contains cycles, then it is the local message-passing algorithm that is most generally applicable. It is well-known that the resulting algorithm may not converge; moreover, when it does converge, the quality of the resulting approximations varies substantially.

Recent work has yielded some insight into the dynamics and convergence properties of BP. For example, several researchers [2, 8, 107, 175] have analyzed the single cycle case, where belief propagation can be reformulated as a matrix powering method. For the special case of graphs corresponding to turbo codes, Richardson [147] developed a geometric approach, through which he was able to establish the existence of fixed points, and give conditions for their stability. More recently, Yedidia et al. [183] showed that BP can be viewed as performing a constrained minimization of the so-called Bethe free energy associated with the graphical distribution,[1] which inspired other researchers [e.g., 177, 184] to develop more sophisticated algorithms for the minimization of the Bethe free energy. Yedidia et al. also proposed extensions to BP based on cluster variational methods [115]; related extensions using higher order structures have been proposed by Minka [132]. These advances notwithstanding, much remains to be understood about the behavior of this algorithm, and more generally about other (perhaps superior) approximation algorithms.

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

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

For a graph with cycles, on the other hand, exact factorizations exposing these marginals do not generally exist. Nevertheless, it is always possible to reparameterize certain *portions* of any factorized representation — namely, any subset of factors corresponding to a cycle-free subgraph of the original graph. We are thus led to consider iterative reparameterization of different subsets, each corresponding to an acyclic subgraph. As we will show, the synchronous form of BP can be interpreted in exactly this manner, in which each reparameterization takes place over the extremely simple tree consisting of a pair of neighboring nodes. This interpretation also applies to a broader class of updates, in which reparameterization is performed over arbitrary cycle-free subgraphs. As a vehicle for studying the concept of reparameterization, the bulk of this paper will focus on updates over *spanning trees*, which we refer to as tree-based reparameterization (or TRP). However, the class of reparameterization algorithms is broad, including not only TRP, sum-product, BP and variants thereof, but also various generalizations [132, 183] (see Chapter 6).

[1] Several researchers have investigated the utility of Bethe tree approximations for graphical models; we refer the reader to [e.g., 164, 181].

At one level, just as BP message-passing can be reformulated as a particular sequence of reparameterization updates, the more global updates of TRP are equivalent to a schedule for message-passing based on spanning trees. We find that tree-based updates often lead to faster convergence, and can converge on problems for which synchronous BP fails. At another level, the reparameterization perspective provides conceptual insight into the nature of belief propagation and related algorithms. In particular, a fact highlighted by reparameterization, yet not obvious from the traditional message-passing viewpoint, is that the overall distribution on the graph with cycles is never altered by such algorithms. Moreover, from the perspective of tree-based updates arises a simple and intuitive characterization of BP fixed points, and more broadly any constrained minimum of the Bethe free energy, in terms of consistency over all trees of the graph. These two properties, when applied in conjunction, allow us to characterize the approximation error for an arbitrary graph with cycles.

In the next section, we introduce the background and notation that underlies our development. In the process, we illustrate how distributions over cycle-free graphs can be reparameterized in terms of local marginal distributions. In Section 5.3, we introduce the class of TRP algorithms. In this context, it is convenient to represent distributions in an exponential form using an overcomplete basis. Our choice of an overcomplete basis, though unorthodox, makes the idea of reparameterization more transparent, and easily stated. In this section, we also show an equivalent formulation of synchronous BP as a sequence of local reparameterizations.

Section 5.4 contains analysis of geometry of TRP updates, as well as the nature of the fixed points. We begin by formalizing the defining characteristic of all reparameterization algorithms — namely, they do not change the distribution on the graph with cycles, but simply yield an alternative factorization. In geometric terms, this invariance means that successive iterates are confined to a linear subspace of exponential parameters (i.e., an $e$-flat manifold in terms of information geometry [e.g., 5, 34].) We then show how each TRP update can be viewed as a projection onto an $m$-flat manifold formed by the constraints associated with each tree, where projection is specified by a cost function $\mathcal{G}$ that is an approximation to the Kullback-Leibler divergence. We prove a Pythagorean result that links TRP iterates, and establishes interesting links between tree-based reparameterization and successive projection algorithms for constrained minimization of Bregman distances [e.g., 31]. The Pythagorean result enables us to show that fixed points of the TRP algorithm satisfy the necessary conditions to be a constrained local minimum of $\mathcal{G}$, thereby making contact with the work of Yedidia et al. [183]. Specifically, we show that although the cost function $\mathcal{G}$ minimized by our TRP algorithm is not the same as the Bethe free energy, TRP fixed points do coincide with extremal points of the Bethe free energy (i.e., with the fixed points of BP). An important benefit of our formulation is a new and intuitive characterization of the fixed points: in particular, any fixed point of BP/TRP must be consistent, in a suitable sense to be defined, with respect to any singly-connected subgraph; and at least one such fixed point of this type is guaranteed to exist. By adapting the invariance and fixed point characterization to the Gaussian (as opposed to discrete) case, we obtain a short and elementary proof of the exactness of the means when BP/TRP converges.

The final topic of this chapter is the analysis of the approximation error arising from application of TRP or BP. Previous results on this error have been obtained in certain special cases. For a single cycle, Weiss [175] derived a relation between the exact marginals and the BP approximations, and for a binary processes showed how local corrections could be applied to compute the exact marginals. Empirically, he also observed that in certain cases, approximation accuracy is correlated with the convergence rate of BP. In the context of turbo decoding, Richardson [147] provided an approximate analysis of the associated error. Despite these encouraging results, a deep and broadly applicable understanding of the approximation error remains a challenging and important problem. Our characterization of the TRP/BP fixed points, in conjunction with the invariance property, allows us to contribute to this goal by analyzing the approximation error for arbitrary graphs. In particular, our development in Section 5.5 begins with the derivation of an *exact* relation between the correct marginals and the approximate marginals computed by TRP or BP. We then exploit this exact relation to derive both upper and lower bounds on the approximation error. The interpretation of these bounds provides an understanding of the conditions that govern the performance of algorithms like TRP or BP. Using results from Chapter 7, these bounds can be computed efficiently. We illustrate their performance on some sample problems. The chapter concludes in Section 5.6 with a summary. Portions of the work presented in this chapter have appeared previously in a conference paper [172] and technical report [171].

## ■ 5.2 Estimation in graphical models

The focus of this chapter is the (approximate) computation of marginal distributions associated with a graph-structured distribution $p(\mathbf{x})$. In particular, the distribution $p(\mathbf{x})$ is defined by a product of compatibility functions $\psi_C$ over the cliques of a graph $G$, as in equation (2.3). Throughout this chapter, we consider graphs in which the clique set $\mathbf{C}$ consists only of singletons and edges (i.e., $\mathbf{C} = V \cup E$), an assumption which is implicit in the synchronous message-passing form of BP presented in Section 5.2.2. In Chapter 6, we consider extensions of reparameterization that operate over higher order cliques. Given only singleton and pairwise cliques, the clique index set ranges over all edges $(s,t) \in E$, as well as the singleton cliques $\{s\}$.

Under these assumptions, the prior distribution $p(\mathbf{x})$ is defined by a product of singleton and edge terms as follow:

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

Here the compatibility functions $\psi_s$ and $\psi_{st}$ denote real-valued functions of $x_s$ and $(x_s, x_t)$ respectively. With a minor abuse of notation, we will often use the same notation to refer to vectors and matrices respectively. In particular, for an $m-$state discrete process, the quantity $\psi_{st}$ lying on the edges $(s,t)$ can be also thought of as a $m \times m$ matrix, where the $(j,k)$ element $\psi_{st;jk}$ is equal to the function value of $\psi_{st}$ for $\{x_s = j, x_t = k\}$. Similarly, the single node functions $\psi_s$ can be thought of as an $m$-vector, where the $j^{th}$ component $\psi_{s;j}$ equals the value of $\psi_s$ for $\{x_s = j\}$. Throughout this chapter, we will switch between these two interpretations, depending on which is more convenient; the interpretation should be clear from the context.

The specific goal of this chapter is to (approximately) compute the marginal probabilities $P_s(x_s)$ of $p(\mathbf{x})$ at each node of the graph. For general graphs with cycles, this task requires summations involving exponentially many terms; indeed, it can be shown to be a NP-hard problem [40]. For tree-structured graphs, there exist direct algorithms for optimal estimation. For graphs with cycles, suboptimal algorithms (such as BP) are used in an attempt to compute approximations to the desired marginals. In the following sections, we elaborate on both of these topics.

## ■ 5.2.1 Exact estimation on trees as reparameterization

Algorithms for optimal estimation on trees have appeared in the literature of various fields, including coding theory [118], artificial intelligence [138], and system theory [14]. See Appendix A for a detailed derivation of one algorithm for optimal estimation on trees. As described in Section 2.1.5, such tree inference algorithms can, in principle, be applied to any graph by clustering nodes so as to form a junction tree. However, in many cases of interest, the aggregated nodes of the junction tree have exponentially large state cardinalities, meaning that applying tree algorithms is prohibitively complex. This explosion in the state cardinality is another demonstration of the intrinsic complexity of exact computations for graphs with cycles.

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

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

where $P_s(x_s)$ and $P_{st}(x_s, x_t)$ are the single node and joint marginal distributions respectively. As an illustration, Figure 5.1(a) shows a simple example of a tree-structured distribution, parameterized in terms of compatibility functions $\psi_s$ and $\psi_{st}$, which leads to the factorization in equation (5.1). Figure 5.1(b) shows this same tree, with the distribution now *reparameterized* in terms of the local marginal distributions $P_s$ and $P_{st}$. The representation of equation (5.2) can be deduced from a more general factorization result on junction trees [e.g. 102,123]. Alternatively, equation (5.2) can be seen as a symmetrized generalization of the well-known factorization(s) of Markov chains. For example, the variables at the three nodes $\{1, 2, 4\}$ in Figure 5.1(b) form a simple

Markov chain, meaning that the joint distribution can be written as

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

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

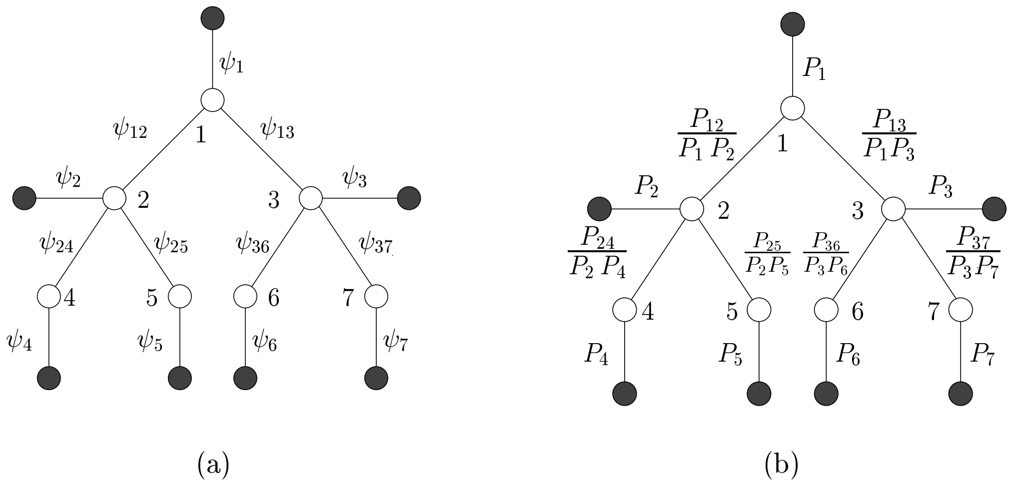



**Figure 5.1.** A simple example of a graphical model; circles correspond to state variables $x_s$, whereas squares correspond to observations. (a) Original parameterization of distribution $p(\mathbf{x}) = \frac{1}{Z}\prod_{s\in V}\psi_s(x_s)\ \prod_{(s,t)\in E}\psi_{st}(x_s,x_t)$ on the tree in terms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

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

## ■ 5.2.2 Belief propagation for graphs with cycles

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

$\mathcal{N}(s) = \{t \in V \mid (s,t) \in E\}$ be the set of neighbors of $s$ in $G$. With this notation, the message at iteration $(n+1)$ is updated based on the messages at the previous iteration $n$ as follows:

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

where $\kappa$ denotes a normalization constant.[2] At any iteration, the "beliefs" — that is, approximations to the marginal distributions — are given by

$$B^n_{s;j} \quad = \quad \kappa\psi_{s;j} \prod_{u \in \mathcal{N}(s)} M^n_{us;j} \tag{5.4}$$

## ■ 5.3 Tree-based reparameterization framework

In this section, we introduce the class of tree-based reparameterization (TRP) updates. Key to TRP is the concept of a tree-structured subgraph of an arbitrary graph $G$ with cycles — i.e., a tree formed by removing edges from the graph. A *spanning tree* is an acyclic subgraph that connects all the vertices of the original graph.

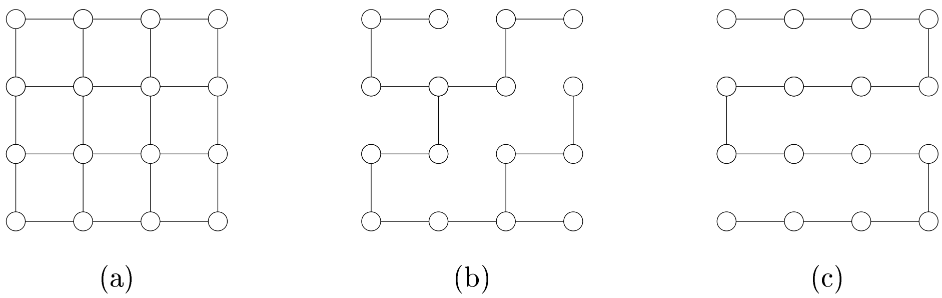



**Figure 5.2.** A graph with cycles has a (typically large) number of spanning trees. (a) Original graph is a nearest neighbor grid. Panels (b) and (c) show two of the $100,352$ spanning trees of the graph in (a).

Figure 5.2 illustrates these definitions: panel (a) shows a nearest neighbor grid, whereas panels (b) and (c) illustrate spanning trees. Of course, these are just two examples of such embedded spanning trees. Indeed, a graph generally has a (large) number of spanning trees, and we exploit this fact in our work.

Now suppose that $\mathcal{T}^0, \ldots, \mathcal{T}^{L-1}$ (with corresponding edge sets $E^0, \ldots, E^{L-1} \subset E$) is a given set of spanning trees for the graph $G$. Then for any $i \in \{0, \ldots, L-1\}$, the distribution $p(\mathbf{x})$ defined by compatibility functions on the graph $G$, as in equation (5.1),

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

can be factored as:

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

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

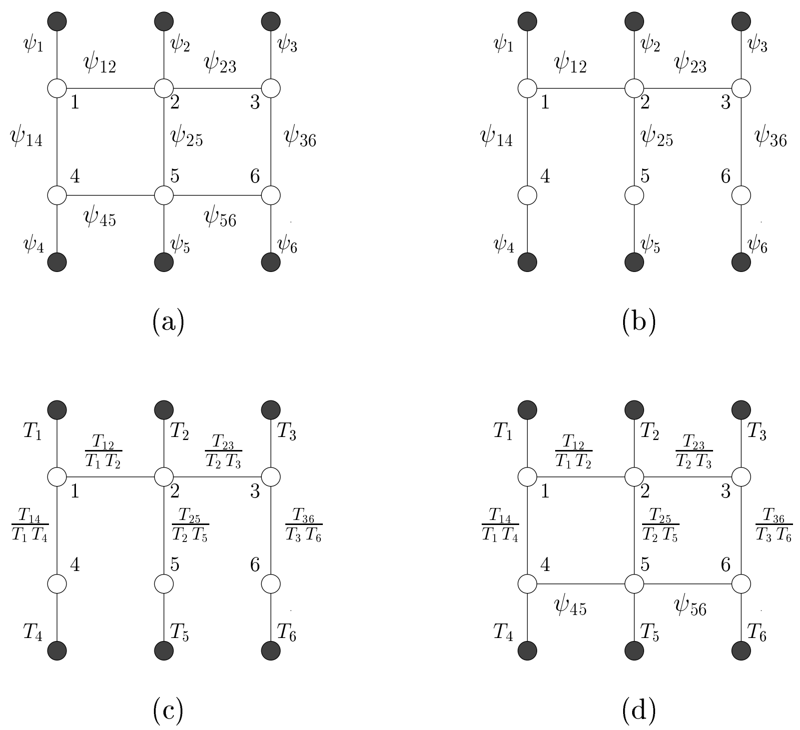



**Figure 5.3.** Illustration of a tree-based reparameterization update. (a) Original parameterization in terms of compatibility functions $\psi_s$ and $\psi_{st}$. (b) Isolate terms corresponding to a tree $\mathcal{T}^i$. (c) Reparameterize the tree-structured component $p^i(\mathbf{x})$ of the distribution in terms of its marginals $\{T_s, T_{st}\}$. (d) New parameterization of the full distribution after a single iteration.

Because the distribution $p^i(\mathbf{x})$ is tree-structured, it can be reparameterized in terms of its marginals, as in equation (5.2). With reference to the full graph $G$ and distribution $p(\mathbf{x})$, this operation simply specifies a new choice of compatibility function for each vertex $s \in V$, and each edge $(s,t) \in E \backslash E^i$, but does not modify the full distribution $p(\mathbf{x})$. In a subsequent update using this new set of functions and choosing a *different* tree $\mathcal{T}^j$, we can write $p(\mathbf{x}) = p^j(\mathbf{x}) r^j(\mathbf{x})$, where $p^j(\mathbf{x})$ includes compatibility functions over cliques in $\mathcal{T}^j$. We can then perform reparameterization for $p^j(\mathbf{x})$, and repeat the process, choosing one of the trees $\mathcal{T}^i$ at each step of the iteration.

Figure 5.3 illustrates the basic steps of this procedure for a simple graph with cycles. Panel (a) shows the original parameterization of $p(\mathbf{x})$ in terms of compatibility functions $\psi_s$ and $\psi_{st}$, as in equation (5.1). A spanning tree, formed by removing edges $(4,5)$ and $(5,6)$, is shown in panel (b); specifically, we have $r^i(\mathbf{x}) = \psi_{45}(x_4, x_5)\, \psi_{56}(x_5, x_6)$ in this case. The tree distribution $p^i(\mathbf{x})$, corresponding to the product of all the other compatibility functions, is reparameterized in terms of marginals $T_s$ and $T_{st}$ computed from the tree-structured distribution $p^i(\mathbf{x})$. The quantities $\{T_s, T_{st}\}$, though exact marginals for this tree-structured distribution, represent approximations to the actual marginals $\{P_s, P_{st}\}$ of the graph with cycles. After this tree reparameterization, the compatibility functions in the residual term $r^i(\mathbf{x})$ are re-instated; the full set of graph compatibility functions following one update are shown in panel (d). In a subsequent update, a different tree is chosen over which reparameterization is to be performed.

At one level, the sequence of updates just described is equivalent to a particular tree-based schedule for message-passing. In particular, a given tree update can be performed by fixing all messages on edges not in the tree, and updating messages on the tree edges until convergence. However, thinking about tree reparameterization instead of message-passing highlights a fundamental property: each step of the algorithm[3] entails specifying alternative factorization of the distribution $p(\mathbf{x})$, and therefore leaves the full distribution intact. To formalize this basic idea, in this section we introduce a particular parameterization of distributions $p(\mathbf{x}; \theta)$, such that iterations of the type just described can be represented as explicit functional updates $\theta^n \mapsto \theta^{n+1}$ on these parameters. We also show that synchronous BP iterations can be interpreted as reparameterization operations using especially simple non-spanning embedded trees, and we present experimental results illustrating the potential advantages of tree-based updates over synchronous BP.

## ■ 5.3.1 Exponential families of distributions

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

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

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

where $\Phi$ is the *log partition function* that normalizes the distribution.

It is standard to specify an exponential family of the form in equation (5.6a) using a set of functions $\boldsymbol{\phi} = \{\phi_\alpha \mid \alpha \in \mathcal{A}\}$ that are linearly independent. This gives rise to a so-called minimal representation [e.g., 13]. However, in this chapter, we will find it convenient to use a non-minimal set of functions. Specifically, let $s, t \in V$ be indices parameterizing the nodes of the graph, and let the indices $j,\ k$ run over the $m$ possible

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

states of the discrete random variables. We then take the index set for $\alpha$, denoted by $\mathcal{A}$, to be the set of pairs $(s;j)$ or 4-tuples $(st;jk)$, and choose the potentials $\phi_\alpha$ as indicator functions for $x$ to take on the indicated value (or values) at the indicated node (or pair of nodes). That is,

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

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

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

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

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

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

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

## ■ 5.3.2 Basic operators

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

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

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

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

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

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

On this basis, we conclude that $\mathbf{P}$ constitutes a set of *mean parameters* dual to the exponential parameters $\theta$. These two parameters are coupled via the relation:

$$\mathbf{P} \quad = \quad \Lambda(\theta) \tag{5.11}$$

where $\Lambda$ is the Legendre transform. (See Section 2.2.4 for more information about the Legendre transform and its properties). Therefore, the vector $\mathbf{P}$ can be viewed as an alternative set of parameters for the distribution $p(\mathbf{x};\theta)$.

Note that the range of $\Lambda$, denoted $Ra(\Lambda)$, is a highly constrained set of vectors. First of all, any $\mathbf{T} \in Ra(\Lambda)$ must belong to the unit hypercube $(0,1)^{d(\theta)}$. Secondly, there are normalization constraints (single-node and joint marginal probabilities must sum to one); and marginalization constraints (pairwise joint distributions, when marginalized, must be consistent with the single node marginals). That is, $Ra(\Lambda) \subseteq \mathbb{C}$, where

$$\mathbb{C} \;=\; \big\{\mathbf{T} \;\big|\; \mathbf{T} \in (0,1)^{d(\theta)};\; \sum_{j=0}^{m-1} T_{s;j} = 1 \text{ for } s \in V;\; \sum_{k=0}^{m-1} T_{st;jk} = T_{s;j} \text{ for } (s,t) \in E\;\big\} \tag{5.12}$$

The elements of $\mathbf{T} \in \mathbb{C}$ define a *locally consistent* set of pairwise and single node marginal distributions on the graph. When $G$ is a tree, then any $\mathbf{T} \in \mathbb{C}$ can be extended (via the tree factorization of equation (5.2)) to a unique distribution $p(\mathbf{x};\theta)$ such that $\mathbf{T} = \Lambda(\theta)$. Thus, for tree-structured graphs, $Ra(\Lambda) = \mathbb{C}$.

For a graph with cycles, in contrast, $Ra(\Lambda)$ is strictly contained within $\mathbb{C}$. Indeed, for graphs with cycles, there exist elements of $\mathbb{C}$ that cannot be realized as the marginals of any distribution (Markov or otherwise). This strict containment reflects the fact that for a graph with cycles, the local consistency conditions defining $\mathbb{C}$ are no longer sufficient to guarantee the existence of a globally consistent distribution.

For a general graph with cycles, of course, the computation of $\Lambda(\theta)$ in equation (5.11) is very difficult. Indeed, algorithms like BP and TRP can be formulated as iteratively generating approximations to $\Lambda(\theta)$. To make a sharp distinction from exact marginal vectors $\mathbf{P} \in Ra(\Lambda) \subseteq \mathbb{C}$, we use the symbol $\mathbf{T}$ to denote such *pseudomarginal vectors.*

We will also make use of the following mapping that is defined for any such $\mathbf{T}$:

$$\big[\Theta(\mathbf{T})\big]_\alpha \;=\; \begin{cases} \log T_{s;j} & \text{if } \alpha = (s;j) \in \mathcal{A} \\ \log\Big[T_{st;jk}/(\sum_j T_{st;jk})(\sum_k T_{st;jk})\Big] & \text{if } \alpha = (st;jk) \in \mathcal{A} \end{cases} \tag{5.13}$$

The quantity $\Theta(\mathbf{T})$ can be viewed as an exponential parameter vector that indexes a distribution $p(\mathbf{x};\Theta(\mathbf{T}))$ on the graph $G$. In fact, consider a marginal vector $\mathbf{P} \in Ra(\Lambda)$. If $G$ is a tree, then not only is the computation of (5.11) simple, but we are also guaranteed $\Theta(\mathbf{P})$ indexes the same graphical distribution as that corresponding to the marginal vector $\mathbf{P}$ — that is:

$$\Lambda(\Theta(\mathbf{P})) = \mathbf{P} \tag{5.14}$$

This equality is simply a restatement of the factorization of equation (5.2) for any tree-structured distribution in terms of its single-node and joint pairwise marginals. However, if $G$ has cycles, then in general the marginal distributions of $p(\mathbf{x};\Theta(\mathbf{P}))$ need not agree with the original marginals $\mathbf{P}$ (i.e., the equality of equation (5.14) does not hold). In fact, determining the exponential parameters corresponding to $\mathbf{P}$ for a graph with cycles is as difficult as the computation of $\Lambda(\theta)$ in equation (5.11). Thus, the composition of operators $\Lambda \circ \Theta$, mapping one marginal vector to another, is the identity for trees but not for general graphs.

Alternatively, we can consider composing $\Theta$ and $\Lambda$ in the other order:

$$\mathcal{R}(\theta) \quad = \quad \big(\Theta \circ \Lambda\big)(\theta) \tag{5.15}$$

which defines a mapping from one exponential parameter vector to another. For a general graph, the operator $\mathcal{R}$ will alter the distribution (that is, $p(\mathbf{x};\theta) \neq p(\mathbf{x};\mathcal{R}(\theta))$). For a tree-structured graph, while $\mathcal{R}$ is not the identity mapping, it does leave the probability distribution unchanged; indeed, applying $\mathcal{R}$ corresponds to shifting from the original parameterization of the tree distribution in terms of $\theta$ to a new exponential parameter $\mathcal{R}(\theta)$ that corresponds directly to the factorization of equation (5.2). As a result, in application to trees, the operator $\mathcal{R}$ is idempotent (i.e., $\mathcal{R} \circ \mathcal{R} = \mathcal{R}$).

## ■ 5.3.3 Tree-based reparameterization updates

The basic idea of TRP is to perform reparameterization updates on a set of spanning trees $\mathcal{T}^0, \ldots, \mathcal{T}^{L-1}$ in succession. The update on any given spanning tree $\mathcal{T}^i$ involves only a subset $\mathcal{A}^i = \{(s;j),\ (st;jk) \mid s \in V,\ (s,t) \in E^i\}$ of all the elements of $\theta$. To move back and forth between parameter vectors on the full graph and those on spanning tree $\mathcal{T}^i$, we define projection and injection operators

$$\Pi^i(\theta) = \{\theta_\alpha \mid \alpha \in \mathcal{A}^i\ \} \tag{5.16a}$$

$$\mathcal{I}^i(\Pi^i(\theta)) = \begin{cases} \theta_\alpha & \text{if } \alpha \in \mathcal{A}^i \\ 0 & \text{if } \alpha \notin \mathcal{A}^i \end{cases} \tag{5.16b}$$

We let $\Lambda^i$, $\Theta^i$ and $\mathcal{R}^i$ denote operators analogous to those in equations (5.11), (5.13) and (5.15) respectively, but as defined for $\mathcal{T}^i$.

Each TRP update acts on the full-dimensional vector $\theta$, but changes only the lower-dimensional subvector $\Pi^i(\theta) = \{\theta_\alpha \mid \alpha \in \mathcal{A}^i\}$. For this reason, it is convenient to use the underbar notation to define operators of the following type:

$$\underline{\mathcal{R}}^i(\theta) \quad = \quad \mathcal{I}^i(\mathcal{R}^i(\Pi^i(\theta))) \tag{5.17a}$$

$$\underline{\Lambda}^i(\theta) \quad = \quad \mathcal{I}^i(\Lambda^i(\Pi^i(\theta))) \tag{5.17b}$$

For instance, $\underline{\Lambda}^i$ projects the exponential parameter vector $\theta$ onto spanning tree $\mathcal{T}^i$; computes the corresponding marginal vector for the distribution $p(\mathbf{x};\Pi^i(\theta))$ induced on the tree; and then injects back to the higher dimensional space by inserting zeroes for elements of edges not in $\mathcal{T}^i$ (i.e., for indices $\alpha \in \mathcal{A}/\mathcal{A}^i$). Moreover, analogous to $\mathbb{C}$, we define a constraint set $\mathbb{C}^i$ by imposing marginalization constraints only for edges in the spanning tree (i.e., as in equation (5.12) with $E$ replaced by $E^i$). Note that $\mathbb{C}^i \supseteq \mathbb{C}$, and since every edge is included in at least one spanning tree, we have that $\cap_i \mathbb{C}^i = \mathbb{C}$.

Using this notation, the operation of performing tree-reparameterization on spanning tree $\mathcal{T}^i$ can be written compactly as transforming a parameter vector $\theta$ into the new vector given by:

$$\begin{aligned}\mathcal{Q}^i(\theta) &= \underline{\mathcal{R}}^i(\theta) + \left[I - \mathcal{I}^i \circ \Pi^i\right](\theta) &(5.18a)\\ &= \theta + \left[\underline{\mathcal{R}}^i(\theta) - \mathcal{I}^i(\Pi^i(\theta))\right] &(5.18b)\end{aligned}$$

where $I$ is the identity operator. The two terms in equation (5.18a) parallel the decomposition of equation (5.5): namely, the operator $\underline{\mathcal{R}}^i$ performs reparameterization of the distribution $p^i(\mathbf{x})$, whereas the operator $[I - \mathcal{I}^i \circ \Pi^i]$ corresponds to leaving the residual term $r^i(\mathbf{x})$ unchanged. Thus, equation (5.18a) is a precise statement of a spanning tree update, such as that illustrated in Figure 5.3, specified in terms of the exponential parameter $\theta$.

Given a parameter vector $\theta$, computing $\mathcal{Q}^i(\theta)$ is straightforward, since it only involves operations on the spanning tree $\mathcal{T}^i$. The tree-based reparameterization algorithm generates a sequence of parameter vectors $\{\theta^n\}$ by successive application of these operators $\mathcal{Q}^i$. The sequence is initialized[4] at $\theta^0$ using the original set of graph functions $\{\psi_s\}$ and $\{\psi_{st}\}$ as follows:

$$\theta^0_\alpha = \begin{cases} \log \psi_{s;j} & \text{if } \alpha = (s;j) \\ \log \psi_{st;jk} & \text{if } \alpha = (st;jk) \end{cases}$$

At each iteration $n$, we choose some spanning tree index $i(n)$ from $\{0, \ldots, L-1\}$, and then update using the operator on spanning tree $\mathcal{T}^i$:

$$\theta^{n+1} = \mathcal{Q}^{i(n)}(\theta^n) \tag{5.19}$$

In the sequel, we will also consider a relaxed iteration, involving a step size $\lambda^n \in (0, 1]$ for each iteration:

$$\theta^{n+1} = \lambda^n \mathcal{Q}^{i(n)}(\theta^n) + (1 - \lambda^n)\theta^n \tag{5.20}$$

where $\lambda^n = 1$ recovers the unrelaxed version.

The only restriction that we impose on the set of spanning trees is that each edge of the full graph $G$ is included in at least one spanning tree (i.e., $\cup_i \mathcal{A}^i = \mathcal{A}$). It is also necessary to specify an order in which to apply the spanning trees — that is, how to choose the index $i(n)$. A natural choice is the *cyclic ordering*, in which we set $i(n) \equiv n \pmod{L}$. More generally, any ordering — possibly random — in which each spanning tree occurs infinitely often is acceptable. A variety of possible orderings for successive projection algorithms are discussed in [31].

[4]Other initializations are also possible. More generally, $\theta^0$ can be chosen as any exponential parameter that induces the same distribution as the original compatibility functions $\{\psi_s\}$ and $\{\psi_{st}\}$.

## 5.3.4 Belief propagation as reparameterization

Just as TRP can be viewed as a particular tree-based schedule for message-passing, we show in this section how to reformulate synchronous BP in "message-free" manner as a sequence of local rather than global reparameterization operations. Specifically, in each step, new compatibility functions are determined by performing exact calculations over extremely simple (non-spanning) trees formed of two nodes and the corresponding edge joining them.

We denote by $M^0_{st} = \{M_{st;k} \mid k \in \mathcal{X}\}$ the $m-$vector corresponding to the chosen initialization of the messages. This choice is often the vector of all ones, but any initialization with strictly positive components is permissible. The message-free version of BP iteratively updates approximations to the exact marginals $\mathbf{P} = \{P_s, P_{st}\}$. Initial values of the approximations $\mathbf{T} = \{T_s, T_{st}\}$ are determined from the initial messages $M^0_{st}$ and the original compatibility functions of the graphical model as follows:

$$T^0_{s;j} = \kappa\, \psi_{s;j} \prod_{u \in \mathcal{N}(s)} M^0_{us;j} \quad \text{for all } s \in V \tag{5.21a}$$

$$T^0_{st;jk} = \kappa\, \psi_{st;jk}\, \psi_{s;j}\, \psi_{t;k} \prod_{u \in \mathcal{N}(s)/t} M^0_{us;j} \prod_{u \in \mathcal{N}(t)/s} M^0_{ut;k} \quad \text{for all } (s,t) \in E \tag{5.21b}$$

where $\kappa$ denotes a normalization factor.

At iteration $n$, these quantities are updated according to the following recursions:

$$T^n_{s;j} = \kappa T^{n-1}_{s;j} \prod_{t \in \mathcal{N}(s)} \frac{1}{T^{n-1}_{s;j}} \sum_{k=0}^{m-1} T^{n-1}_{st;jk} \tag{5.22a}$$

$$T^n_{st;jk} = \kappa \frac{T^{n-1}_{st;jk}}{\left(\sum_{j=0}^{m-1} T^{n-1}_{st;jk}\right)\left(\sum_{k=0}^{m-1} T^{n-1}_{st;jk}\right)} T^n_{s;j} T^n_{t;k} \tag{5.22b}$$

The update in equation (5.22b) is especially noteworthy: it corresponds to performing optimal estimation on the very simple two-node tree formed by edge $(s,t)$. As an illustration, Figure 5.4(b) shows the decomposition of a single-cycle graph into such two-node trees. This simple reparameterization algorithm operates by performing optimal estimation on this set of non-spanning trees, one for each edge in the graph, as in equation (5.22b). The single-node marginals from each such tree are merged via equation (5.22a).

We now claim that this reparameterization algorithm is equivalent to belief propagation, summarizing the result as follows:

**Proposition 5.3.1.** *The reparameterization algorithm specified in the equations* (5.21) *and* (5.22) *is equivalent to the message-passing form of BP given in equations* (5.3) *and* (5.4). *In particular, for each iteration $n = 0, 1, \ldots$ and initial message vector $M^0_{st}$,*

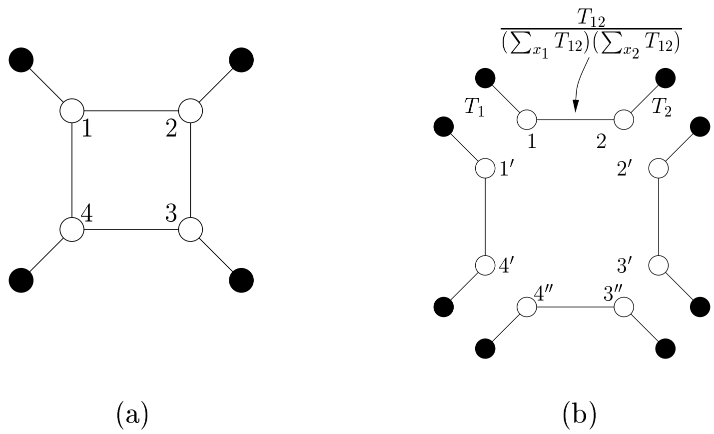



**Figure 5.4.** (a) Toy example of original graph. (b) Two node trees used for updates in message-free version of belief propagation. Computations are performed exactly on each two-node tree formed by a single edge and the two associated observation potentials as in equation (5.22b). The node marginals from each two-node tree are merged via equation (5.22a).

*we have the following relations:*

$$M^{n+1}_{st;k} = \kappa M^0_{st;k} \prod_{i=0}^{n} \frac{1}{T^i_{t;k}} \sum_{j=0}^{m-1} T^i_{st;jk} \quad \text{for all } (s,t) \in E \tag{5.23a}$$

$$B^n_{s;j} = T^n_{s;j} \quad \text{for all } s \in V \tag{5.23b}$$

*where $\kappa$ denotes a normalization factor.*

*Proof.* See Appendix C.1. □

## ■ 5.3.5 Empirical comparisons of BP versus TRP

Given that a spanning tree reaches every node of the graph, one might expect tree-based updates, such as those of TRP, to have convergence properties superior to those of local updates such as synchronous BP. As stated previously, a single TRP update on a given spanning tree can be performed by fixing all the messages on edges not in tree, and updating messages on edges in the tree until convergence. Such tree-based message updating schedules are used in certain applications of BP, such as turbo decoding [131], for which there are natural choices of trees over which to perform updates. In this section, we provide experimental evidence supporting the claim that tree-based updates have superior convergence properties for other problems. An interesting but open question raised by these experiments is how to optimize the choice of trees (not necessarily spanning) over which to perform the updates.

### Convergence rates

In this section, we report the results of experiments on the convergence rates of TRP and BP on three graphs: a single cycle with 15 nodes, a $7 \times 7$ nearest neighbor grid, and a larger $40 \times 40$ grid. At first sight, the more global nature of TRP might suggest that each TRP iteration is more complex computationally than the corresponding BP iteration. In fact, the opposite statement is true. Each TRP update corresponds to solve a tree problem exactly, and therefore requires $\mathcal{O}(m^2(N-1))$ operations.[5] In contrast, each iteration of synchronous BP requires $\mathcal{O}(m^2|E|)$ operations, where $|E| \geq N-1$ is the number of edges in the graph. In order to make comparisons fair in terms of actual computation required, whenever we report iteration numbers, they are rescaled in terms of relative cost (i.e., for each graph, TRP iterations are rescaled by the ratio $(N-1)/|E| < 1$). In all cases, we used unrelaxed updates for both BP and TRP.

For each graph, we performed simulations under three conditions: edge potentials that are *repulsive* (i.e., that encourage neighboring nodes to take opposite values); *attractive* (that encourage neighbors to take the same value); and *mixed* (in which some potentials are attractive, while others are repulsive). For each of these experimental conditions, each run involved a random selection of the initial parameter vector $\theta^0$ defining the distribution $p(\mathbf{x}; \theta^0)$. In all experiments reported here, we generated the single node parameters $\theta_{s;j}$ as follows:[6] for each node $s \in V$, sample $a_s \sim \mathcal{N}(0, (0.25)^2)$, and set $[\theta_{s;0} \;\; \theta_{s;1}] = [a_s \;\; -a_s]$. To generate the edge potential components $\theta_{st;jk}$, we begin by sampling $b_{st} \sim \mathcal{N}(0,1)$ for each edge $(s,t)$. With $\delta_{jk}$ denoting the Kronecker delta for $j,k$, we set the edge potential components in one of three ways depending on the experimental condition. For the *repulsive condition*, we set $\theta_{st;jk} = -(2\delta_{jk}-1)\,|b_{st}|$; for the *attractive condition*, $\theta_{st;jk} = (2\delta_{jk}-1)\,|b_{st}|$; whereas for the *mixed condition* $\theta_{st;jk} = (2\delta_{jk}-1)\,b_{st}$.

For each experimental condition, we performed a total of 500 trials for each of the single cycle and $7 \times 7$ grid, comparing the performance of TRP to BP. On any given run, an algorithm was deemed to converge when the mean $L^2$ difference between successive node elements $(\frac{1}{n}\sum_s \|\theta_s^{n+1} - \theta_s^n\|^2)$ reached a threshold of $\epsilon = 1 \times 10^{-16}$. A run in which which a given algorithm failed to reach this threshold within 3000 iterations was classified as a failure to converge. In each condition, we report the total number of convergent trials (out of 500); and also the mean number of iterations required to converge, rescaled by the ratio $(N-1)/|E|$ and based only on trials where both TRP and BP converged.

Table 5.1 shows some summary statistics for the two graphs used in these experiments. For the single cycle, we implemented TRP with two spanning trees, whereas we used four spanning trees for the grid. Although both algorithms converged on all trials for the single cycle, the rate of TRP convergence was significantly (roughly 3 times) faster. For the grid, algorithm behavior depends more on the experimental condition.

[5] Here we are using the fact that a tree problem can be solved efficiently by a two-pass sweep, where exactly two messages are passed along each edge of the graph.

[6] The notation $\mathcal{N}(0, \sigma^2)$ denotes a zero-mean Gaussian with variance $\sigma^2$.

| Graph | Single 15-cycle | | | | | | 7 × 7 grid | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | R | | A | | M | | R | | A | | M | |
| BP | 500 | 23.2 | 500 | 23.4 | 500 | 23.6 | 455 | 62.3 | 457 | 65.8 | 267 | 310.1 |
| TRP | 500 | 8.1 | 500 | 8.0 | 500 | 8.2 | 500 | 30.5 | 500 | 30.8 | 282 | 103.2 |

**Table 5.1.** Comparison of convergence behavior of TRP versus BP for a single cycle of 15 nodes; and a $7 \times 7$ grid. Potentials were chosen randomly in each of the three conditions: repulsive potentials (R); attractive potentials (A); mixed potentials (M). First and second numbers in each box denote the number of convergent runs out of 500; and the mean number of iterations (rescaled by relative cost and computed using only runs where both TRP and BP converged) respectively.

The repulsive and attractive conditions are relatively easy, though still difficult enough for BP that it failed to converge on roughly 10% of the trials, in contrast to the perfect convergence percentage of TRP. In terms of mean convergence rates, TRP converged more than twice as quickly as BP. The mixed condition is difficult for suitably strong edge potentials on a grid: in this case both algorithms failed to converge on almost half the trials, although TRP converged more frequently than BP. Moreover, on runs where both algorithms converged, the TRP mean rate of convergence was roughly three times faster than BP. Although mean convergence rates were faster, we did find individual problems on the grid for which the version of TRP with four trees converged more slowly than BP. However, one possibility (which we did not take advantage of here) is to optimize the choice of trees in an adaptive manner.

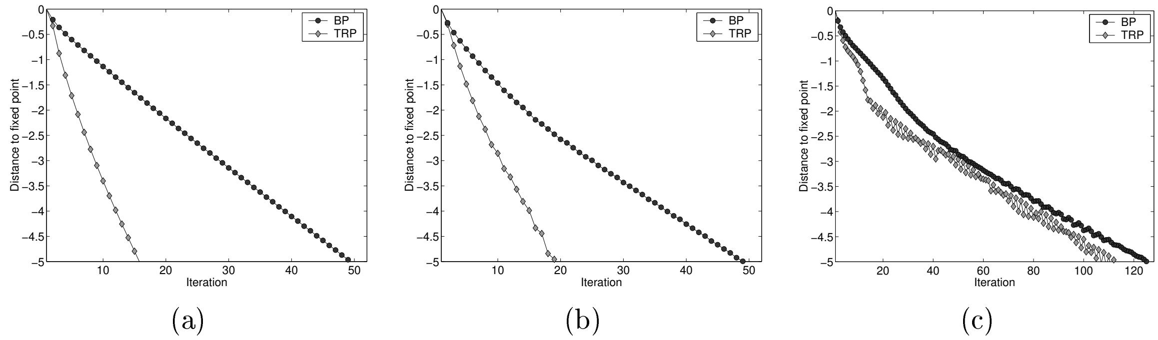



**Figure 5.5.** Convergence rates for TRP versus BP on a $40 \times 40$ grid. Plotted on a log scale is the $L^2$ distance $(\sum_{s,j} |\theta^n_{s;j} - \theta^*_{s;j}|^2)$ from current iterate $\theta^n$ to fixed point $\theta^*$ versus iteration number $n$. In all cases, both BP and TRP converge to the same fixed point $\theta^*$. (a) Repulsive potentials. (b) Mixed potentials. (c) Particular choice of mixed potentials that causes difficulty for TRP.

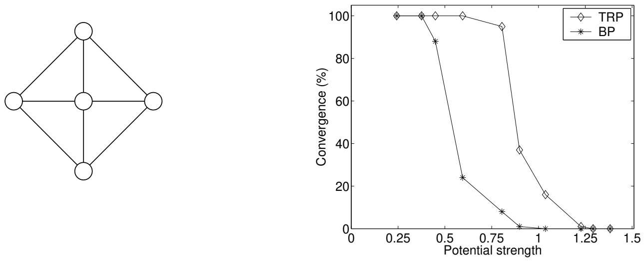
We also examined convergence behavior for the $40 \times 40$ grid with 1600 nodes, using a version of TRP updates over two spanning trees. Figure 5.5 provides an illustration of the convergence behavior of the two algorithms. Plotted on a log scale is the $L^2$ distance between the single-node elements of $\theta^n$ and $\theta^*$ at each iteration, where $\theta^*$ is a fixed point common to BP and TRP, versus the iteration number. Again, the TRP iteration are rescaled by their cost relative to BP iterations ($(N-1)/|E|$), which for this large grid is very close to 0.5. Panel (a) illustrates a case with repulsive potentials, for which the TRP updates converge quite a bit faster than BP updates. Examples in the attractive condition show similar convergence behavior. Panels (b) and (c) show two different examples, each with a mixed set of potentials. The mixed condition on the grid is especially difficult due to the possibility of conflicting or frustrated interactions between the nodes. For the problem in (b), the two spanning trees used for this particular version of TRP are a good choice, and again lead to faster convergence. The potentials for the problem in (c), in contrast, cause difficulties for this pair of spanning trees; note the erratic convergence behavior of TRP.

Each TRP update ignores some local interactions corresponding to the edges removed to form the spanning tree. These edges are covered by other spanning trees in the set used; however, it remains an open question how to choose trees so as to maximize the rate of convergence. In this context, one could imagine a hybrid algorithm in which pure synchronous BP iterations are interspersed with iterations over more global structures like trees (not necessarily spanning). The exploration of such issues remains for future research.

#### Domain of convergence

We have also found that tree-based updates can converge for a wider range of potentials than synchronous BP. The simple 5-node graph shown in Figure 5.6(a) serves to illustrate this phenomenon. We simulated a binary process over a range of potential strengths $\mu$ ranging from $-0.3$ to $-1.0$. Explicitly, for each value of $\mu$, we made a deterministic assignment of the potential for each edge $(s,t)$ of the graph as $\theta_{st;jk} = (2\delta_{jk} - 1)\mu$. For each potential strength we conducted 100 trials, where on each trial the single-node potentials were set randomly by sampling $a_s \sim \mathcal{N}(0, (0.25)^2)$ and setting $[\theta_{s;0} \quad \theta_{s;1}] = [a_s \quad -a_s]$. On any given trial, the convergence of a given algorithm was assessed as in Section 5.3.5. Plotted in Figure 5.6(b) is the percentage of successfully converged trials versus potential strength for TRP and BP. Both algorithms exhibit a type of threshold behavior, in which they converge with 100% success up to a certain potential strength, after which their performance degrades rapidly. However, the tree-based updates extend the effective range of convergence.[7] To be fair, recently proposed alternatives to BP for minimizing the Bethe free energy [e.g., 177,184], though they entail greater computational cost than the updates considered here, are guaranteed to converge to a stationary point.

[7] This result is not dependent on the symmetry of the problem induced by our choice of edge potentials; for instance, the results are similar if edge potentials are perturbed from their nominal strengths by small random quantities.




(a) (b)

**Figure 5.6.** (a) Simple 5-node graph. (b) Comparison of BP and TRP convergence percentages versus function of potential strength on graph in (a). Plotted along the abscissa as a measure of potential strength is the multi-information $D(p(\mathbf{x};\theta) \parallel \prod_{s=1}^{N} p(x_s;\theta))$. Both TRP and BP exhibit a threshold phenomenon, with TRP converging for a wider range of potentials.

## ■ 5.4 Analysis of fixed points and convergence

In this section, we present a number of results related to the geometry and fixed points of reparameterization algorithms like TRP and BP. The defining characteristic of a reparameterization algorithm is that the original distribution is never altered. Accordingly, we begin in Section 5.4.1 with a formal statement of this property in terms of exponential parameters, and then establish its validity for the more general class of relaxed updates. We also develop the information-geometric interpretation of this result. Motivated by this geometric view, we show that a TRP update can be viewed as a projection onto the tree constraint set $\mathbb{C}^i$. This projection is defined by a particular cost function, defined in Section 5.4.2, that arises as an approximation to the Kullback-Leibler (KL) divergence. In Section 5.4.3, we show that successive TRP iterates satisfy a Pythagorean relation with respect to this cost function. This result is of independent interest because it establishes links to successive projection techniques for constrained minimization of Bregman distances [e.g., 31]. In Section 5.4.4, we use this Pythagorean relation to prove that fixed points of the TRP algorithm satisfy necessary conditions to be a constrained minimum of this cost function. By combining our results with those of Yedidia et al. [183], we conclude that fixed points of the TRP algorithm coincide with those of BP. The Pythagorean result also allows us to formulate a set of sufficient conditions for convergence of TRP in the case of two spanning trees, which we present in Section 5.4.5. In Section 5.4.6, we provide an elementary proof of the result originally developed in [152,176] concerning the behavior of BP for jointly Gaussian distributions. Finally, in Section 5.4.7, we address the question of when (other than for tree-structured distributions) the TRP/BP approximations can be exact.

## 5.4.1 Geometry and invariance of TRP updates

Highlighted by our informal set-up in Section 5.3 is a fundamental property of reparameterization algorithms like TRP and BP — specifically, they do not change the original distribution on the graph with cycles. In this section, we formalize this notion of invariance, and show that it also holds for the more general class of relaxed updates in equation (5.20). On the basis of this invariance, we provide an illustration of the TRP updates in terms of information geometry [7, 34, 45]. This geometric perspective provides intuition and guides our subsequent analysis of reparameterization algorithms and their fixed points.

From the perspective of reparameterization, a crucial feature of the exponential parameterization defined in equation (5.7) is its overcompleteness. For this reason, given a fixed exponential parameter $\widetilde{\theta}$, it is interesting to consider the following subset of $\mathbb{R}^{d(\theta)}$:

$$\mathcal{M}(\widetilde{\theta}) \triangleq \left\{ \theta \in \mathbb{R}^{d(\theta)} \;\mid\; p(\mathbf{x};\theta) \equiv p(\mathbf{x};\widetilde{\theta}) \right\} \tag{5.24}$$

where $d(\theta)$ denotes the length of $\theta$ as defined in equation (5.8). This set can be seen to be a closed submanifold of $\mathbb{R}^{d(\theta)}$— in particular, note that it is the inverse image of the point $\widetilde{\theta}$ under the continuous mapping $\theta \mapsto p(\mathbf{x};\theta)$. In order to further understand the structure of $\mathcal{M}(\widetilde{\theta})$, we need to link the overcomplete parameterization to a minimal parameterization, specified by a linearly independent collection of functions.

We begin with the special case of binary-valued nodes ($m = 2$). Recall from Example 2.2.2 of Section 2.2.1 that the standard minimal representation of a distribution on a binary vector with pairwise potentials has the form:

$$p(\mathbf{x};\gamma) = \exp\Big\{ \sum_{s} \gamma_s x_s + \sum_{s,t \in E} \gamma_{st} x_s x_t - \Phi(\gamma) \Big\} \tag{5.25}$$

Here we use the parameter $\gamma$ to distinguish this minimal representation from the overcomplete parameter $\theta$ used in TRP updates. Similarly, as shown by the discussion of Section 2.2.1, an $m$-ary process on a graph with pairwise potentials has a minimal representation in terms of the collection of functions:

$$\mathcal{R}(s) \triangleq \{x_s^a \mid a = 1, \ldots, m-1\} \qquad \text{for } s \in V \tag{5.26a}$$

$$\mathcal{R}(s,t) = \{x_s^a x_t^b \mid a, b = 1, \ldots, m-1\} \qquad \text{for } (s,t) \in E \tag{5.26b}$$

As in the binary case illustrated above, we let $\gamma$ be a parameter vector of weights on these functions.

In contrast to the overcomplete case, the minimal representation induces a one-to-one correspondence between parameter vectors $\gamma$ and distributions $p(\mathbf{x};\gamma)$. Therefore, associated with the distribution $p(\mathbf{x};\widetilde{\theta})$ is a unique vector $\widetilde{\gamma}$ such that $p(\mathbf{x};\widetilde{\theta}) \equiv p(\mathbf{x};\widetilde{\gamma})$. The dimension of the exponential family [see 5] is given by the length of $\gamma$, which we denote by $d(\gamma)$. From equation (5.26), we see that this dimension is given by $d(\gamma) = [\,(m-1)\,N + (m-1)^2\,|E|\,]$. On the basis of these equivalent representations, the set $\mathcal{M}(\widetilde{\theta})$ can be characterized as follows:

**Proposition 5.4.1.** *The set $\mathcal{M}(\widetilde{\theta})$ of equation (5.24) is a linear (e-flat) submanifold of $\mathbb{R}^{d(\theta)}$ of dimension $d(\theta) - d(\gamma)$. It has the form $\{\theta \in \mathbb{R}^{d(\theta)} \mid A\,\theta = \widetilde{\gamma}\,\}$, where $A$ is an appropriately defined $d(\gamma) \times d(\theta)$ matrix of constraints.*

*Proof.* See Appendix C.5. □

Based on this proposition, we can provide a geometric statement and proof of the invariance of TRP updates:

**Theorem 5.4.1 (Invariance of distribution).** *Consider a sequence of TRP iterates $\{\theta^n\}$ generated by the relaxed updates:*

$$\theta^{n+1} \;=\; \lambda^n \mathcal{Q}^{i(n)}(\theta^n) + (1-\lambda^n)\theta^n \tag{5.27}$$

*Then the distribution on the full graph with cycles is invariant under the updates: that is, $\theta^n$ belongs to the set $\mathcal{M}(\theta^0) = \{\theta \in \mathbb{R}^{d(\theta)} \mid p(\mathbf{x};\theta) \equiv p(\mathbf{x};\theta^0)\}$ for all $n = 1, 2, \ldots$. Moreover, any limit point $\theta^*$ of the sequence also belongs to $\mathcal{M}(\theta^0)$.*
*In addition, the same statements hold for the reparameterization form of BP presented in Section 5.3.4.*

*Proof.* As previously described, the unrelaxed TRP update of equation (5.19) does indeed leave the distribution unchanged, so that $\mathcal{Q}^i(\theta) \in \mathcal{M}(\theta)$ for all $\theta$. The relaxed update of equation (5.27) is nothing more than a convex combination of two exponential vectors ($\theta^n$ and $\mathcal{Q}^{i(n)}(\theta^n)$) that parameterize the same distribution, so that by recourse to Proposition 5.4.1, the proof of the first statement is complete. As noted earlier, $\mathcal{M}(\theta^0)$ is a closed submanifold, so that any limit point of the sequence $\{\theta^n\}$ must also belong to $\mathcal{M}(\theta^0)$. An inductive argument establishes that the reparameterization form of BP also leaves invariant the distribution on the full graph.

□

These results together also lead to a geometric understanding of the TRP updates in the exponential domain (i.e., in terms of the parameter vector $\theta$). In order to describe this geometry, we begin by defining an exponential analog of the constraint set $\mathbb{C}$ as follows:

$$\mathbb{D} \;=\; \{\theta \mid \theta = \Theta(\mathbf{T}) \text{ for some } \mathbf{T} \in \mathbb{C}\} \;=\; \Theta(\mathbb{C}) \tag{5.28}$$

If a vector $\theta$ belongs to the set $\mathbb{D}$, then it must satisfy certain nonlinear convex constraints (e.g., $\log[\sum_j \exp(\theta_{s;j})] = 0$ for all $s \in V$; and $\log[\sum_j \exp(\theta_{st;jk} + \theta_{s;j})] = 0$ for all $(s,t) \in E$). For each spanning tree constraint set $\mathbb{C}^i$, we also define the set $\mathbb{D}^i$ in an analogous manner. Note that for any $\theta$, the updated $\mathcal{Q}^i(\theta)$ is guaranteed to belong to $\mathbb{D}^i$. Moreover, the set-up of the algorithm ensures that $\mathbb{D} = \cap_i \mathbb{D}^i$.

Figure 5.7 illustrates the geometry of TRP updates. The sequence of iterates $\{\theta^n\}$ remains within the linear manifold $\mathcal{M}(\theta^0)$. In terms of information geometry [5], this manifold is $e$-flat, since it is linear in the exponential parameters. Note that $\mathbb{C}$ and each

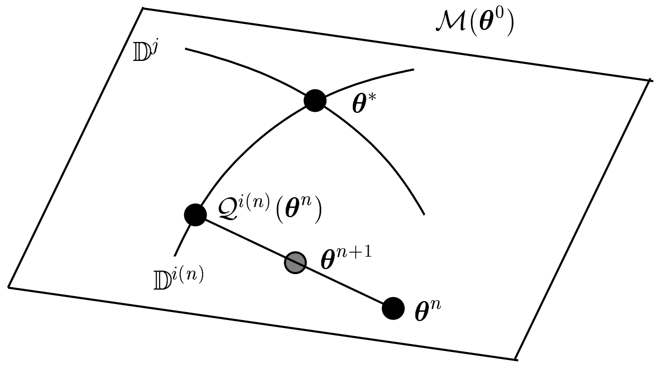



**Figure 5.7.** Geometry of tree-reparameterization updates in the exponential domain. Iterates are confined to the linear manifold $\mathcal{M}(\theta^0)$. Curved lines within $\mathcal{M}(\theta^0)$ correspond to the intersection $\mathbb{D}^i \cap \mathcal{M}(\theta^0)$, for a particular spanning tree constraint set $\mathbb{D}^i$. Each update entails moving along the line between $\theta^n$ and the point $\mathcal{Q}^{i(n)}(\theta^n)$ on $\mathbb{D}^{i(n)}$. Any fixed point $\theta^*$ belongs to the intersection of $\mathbb{D} = \cap_i \mathbb{D}^i$ with $\mathcal{M}(\theta^0)$.

$\mathbb{C}^i$ are defined by linear constraints in terms of the (pseudo)marginal $\mathbf{T}$, and so are $m$-flat manifolds. Each set $\mathbb{D}^i$, as defined in equation (5.28), is the image under $\Theta$ of the constraint set $\mathbb{C}^i$, and so is a curved manifold in terms of exponential parameters. Therefore, the intersection $\mathbb{D}^i \cap \mathcal{M}(\theta^0)$ forms a curved line, as illustrated in Figure 5.7. Each update consists of moving along the straight line between the current iterate $\theta^n$, and the point $\mathcal{Q}^{i(n)}(\theta^n)$ obtained by applying the tree reparameterization operator $\mathcal{Q}^{i(n)}$. By construction, the vector $\mathcal{Q}^{i(n)}(\theta^n)$ belongs to the constraint set $\mathbb{D}^{i(n)}$. The ultimate goal of a reparameterization algorithm is obtain a point $\theta^*$ in the intersection $\cap_i \mathbb{D}^i$ of all the tree constraint sets.

## ■ 5.4.2 Approximation to the Kullback-Leibler divergence

Based on the geometric view illustrated in Figure 5.7, an unrelaxed TRP update corresponds to moving from $\theta^n$ to the point $\mathcal{Q}^{i(n)}(\theta^n)$ in the constraint set $\mathbb{D}^{i(n)} = \Theta(\mathbb{C}^{i(n)})$. We now show that this operation corresponds to a type of projection; that is, it can be formulated as finding the point in $\mathbb{C}^{i(n)}$ that is closest to $\theta^n$, where "distance" is measured by a cost function to be defined. In particular, the cost function $\mathcal{G}$ central to our analysis arises as an approximation to the Kullback-Leibler (KL) divergence [42], one which is exact for a tree.

Let $\mathbf{T} \in (0,1)^{d(\theta)}$ be a pseudomarginal vector, and let $\theta$ be a parameter vector for the original graph $G$ with cycles. As building blocks for defining the full cost function, we define functions for each node $s$ and edge $(s,t)$ as follows:

$$\mathcal{G}_{st}(T_{st};\theta_{st}) = \sum_{j,k} T_{st;jk}\Big\{ \log\big[T_{st;jk}/(\sum_j T_{st;jk})(\sum_k T_{st;jk})\big] - \theta_{st;jk}\Big\} \quad (5.29a)$$

$$\mathcal{G}_{s}(T_{s};\theta_{s}) = \sum_{j} T_{s;j}\big[\log T_{s;j} - \theta_{s;j}\big] \quad (5.29b)$$

We then define the cost function as:

$$\mathcal{G}(\mathbf{T};\theta) \triangleq \sum_{s\in V}\mathcal{G}_s(T_s;\theta_s) + \sum_{(s,t)\in E}\mathcal{G}_{st}(T_{st};\theta_{st}) \equiv \sum_{\alpha\in\mathcal{A}} T_\alpha\big[\Theta(\mathbf{T})-\theta\big]_\alpha \quad (5.30)$$

It can be seen that this cost function is equivalent to the Bethe free energy [183] when $\mathbf{T}$ belongs to the constraint set $\mathbb{C}$, but distinct for vectors $\mathbf{T}$ that do not satisfy the marginalization constraints defining membership in $\mathbb{C}$.

To see how $\mathcal{G}$ is related to the KL divergence as defined in equation (2.31), consider the analogous function defined on spanning tree $\mathcal{T}^i$ for a vector $\mathbf{T}\in\mathbb{C}^i$:

$$\mathcal{G}^i(\Pi^i(\mathbf{T});\Pi^i(\theta)) = \sum_{s\in V}\mathcal{G}_s(T_s;\theta_s) + \sum_{(s,t)\in E^i}\mathcal{G}_{st}(T_{st};\theta_{st}) \equiv \sum_{\alpha\in\mathcal{A}^i} T_\alpha\big[\Theta^i(\Pi^i(\mathbf{T}))-\theta\big]_\alpha \quad (5.31)$$

where $\Pi^i(\theta)$ and $\Pi^i(\mathbf{T})$ are exponential parameter vectors and marginal vectors, respectively, defined on $\mathcal{T}^i$. With the exponential parameterization of equation (5.7) applied to any tree, we have $T_\alpha = \mathbb{E}_{\Theta^i(\Pi^i(\mathbf{T}))}[\phi_\alpha(\mathbf{x})]$ for all indices $\alpha\in\mathcal{A}^i$. As a result, the function $\mathcal{G}^i$ is related to the KL divergence as follows:

$$D(\Theta^i(\Pi^i(\mathbf{T}))\;\|\;\Pi^i(\theta)) = \mathcal{G}^i(\Pi^i(\mathbf{T});\Pi^i(\theta)) + \Phi(\Pi^i(\theta)) \quad (5.32)$$

In establishing this equivalence, we have used the fact that the partition function of the factorization in equation (2.6) is unity, so that the corresponding log partition function is zero (i.e., $\Phi(\Theta^i(\Pi^i(\mathbf{T}))) = 0$). Therefore, aside from an additive constant $\Phi(\Pi^i(\theta))$ independent of $T$, the quantity $\mathcal{G}^i(\Pi^i(\mathbf{T});\Pi^i(\theta))$, when viewed as a function of $\Pi^i(\mathbf{T})$, is equivalent to the KL divergence.

Now consider the problem of minimizing the KL divergence as a function of $\mathbf{T}$, subject to the constraint $\mathbf{T}\in\mathbb{C}^i$. The KL divergence in equation (5.32) assumes its minimum value of zero at the vector of correct marginals on the spanning tree — namely, $\mathbf{P}^i = \underline{\Lambda}^i(\Pi^i(\theta))\in\mathbb{C}^i$. By the equivalence shown in equation (5.32), minimizing the function $\mathcal{G}^i(\Pi^i(\mathbf{T});\Pi^i(\theta))$ over $\mathbf{T}\in\mathbb{C}^i$ will also yield the same minimizing argument $\mathbf{P}^i$.

For the original graph $G$ with cycles, the cost function $\mathcal{G}$ of equation (5.30) is not equivalent to the KL divergence. The argument leading up to equation (5.32) cannot be applied because $\Lambda(\Theta(\mathbf{T}))\neq\mathbf{T}$ for a general graph with cycles. Nevertheless, this cost function lies at the core of our analysis of TRP. Indeed, we show in Section 5.4.3

how the TRP algorithm can be viewed as a successive projection technique for constrained minimization of the cost function $\mathcal{G}$, in which the reparameterization update on spanning tree $\mathcal{T}^i$ as in equation (5.19) corresponds to a projection onto constraint set $\mathbb{C}^i$. Moreover, since $\mathcal{G}$ agrees with the Bethe free energy [183] on the constraint set $\mathbb{C}$, this allows us to establish equivalence of TRP fixed points with those of BP.

## 5.4.3 Tree reparameterization updates as projections

Given a linear subspace $\mathcal{L} \subset \mathbb{R}^n$ and a vector $\mathbf{y} \in \mathbb{R}^n$, it is well-known [20] that the projection $\widehat{\mathbf{x}}$ under the Euclidean norm (i.e., $\widehat{\mathbf{x}} \triangleq \arg\min_{\mathbf{x}\in\mathcal{L}} \|\mathbf{x}-\mathbf{y}\|$) is characterized by an orthogonality condition, or equivalently a Pythagorean relation. The main result of this section is to show that a similar geometric picture holds for TRP updates with respect to the cost function $\mathcal{G}$. As a consequence, each iteration of TRP can be viewed as projecting the current iterate $\theta$ onto the constraint set $\mathbb{C}^{i(n)}$, where $\mathcal{G}$ plays the role of a squared distance function:

**Proposition 5.4.2 (Pythagorean relation).** *Assume that the sequence $\{\theta^n\}$ generated by equation (5.20) with step sizes $\lambda^n$ remains bounded. Let $i = i(n)$ be the tree index used at iteration $n$. Then for all $\mathbf{U} \in \mathbb{C}^i$:*

$$\mathcal{G}(\mathbf{U};\theta^n) = \mathcal{G}(\mathbf{U};\theta^{n+1}) + \lambda^n \mathcal{G}(\underline{\Lambda}^i(\mathcal{Q}^i(\theta^n));\theta^n) \tag{5.33}$$

*where $\underline{\Lambda}^i$ is defined in equation (5.17b).*

*Proof.* See Appendix C.2. □

An important special case of Proposition 5.4.2 is the unrelaxed update ($\lambda^n = 1$), in which case equation (5.33) simplifies to

$$\mathcal{G}(\mathbf{U};\theta^n) = \mathcal{G}(\mathbf{U};\theta^{n+1}) + \mathcal{G}(\underline{\Lambda}^i(\theta^{n+1});\theta^n) \tag{5.34}$$

A result analogous to equation (5.34) holds for the minimum of a Bregman distance over a linear constraint set [e.g., 31]. Well-known examples of Bregman distances include the $L^2$ norm, as well as the Kullback-Leibler (KL) divergence. Choosing the KL divergence as the Bregman distance leads to the I-projection in information geometry [e.g., 7,34,45].

Even when the distance is not the $L^2$ norm, results of the form in equation (5.34) are still called Pythagorean, because the function $\mathcal{G}$ plays the role (in a loose sense) of the squared Euclidean distance. This geometric interpretation is illustrated in Figure 5.8. For the unrelaxed updates, we are guaranteed the existence[8] of a pseudomarginal $\mathbf{T}^n$ such that $\theta^n = \Theta(\mathbf{T}^n)$. The three points $\mathbf{T}^n$, $\underline{\Lambda}^i(\theta^{n+1})$ and $\mathbf{U}$ analogous to the vertices of a right triangle, as drawn in Figure 5.8. We project the point $\mathbf{T}^n$ onto the constraint

[8]The image of the unit hypercube $(0,1)^{d(\theta)}$ under the map $\Theta$ is not all of $\mathbb{R}^{d(\theta)}$, since, for example, given any pseudomarginal $\mathbf{T} \in (0,1)^{d(\theta)}$, we have $[\Theta(\mathbf{T})]_{s;j} = \log T_{s;j} < 0$. Nonetheless, for unrelaxed updates producing iterates $\theta^n$, it can be seen that the inverse image of a point $\theta^n$ under $\Theta$ will be non-empty as soon as each edge has been updated at least once.

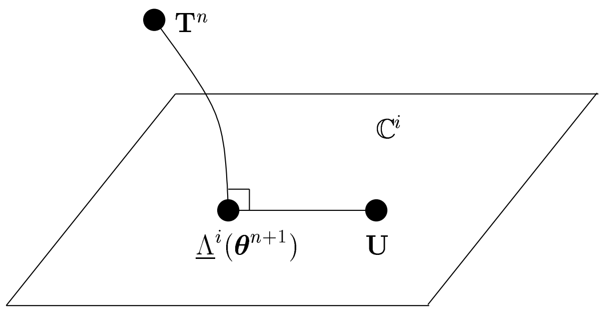



**Figure 5.8.** Illustration of the geometry of Proposition 5.4.2. The pseudomarginal vector $\mathbf{T}^n$ is projected onto the linear constraint set $\mathbb{C}^i$. This yields the point $\underline{\Lambda}^i(\theta^{n+1})$ that minimizes the cost function $\mathcal{G}^i$ over the constraint set $\mathbb{C}^i$.

set $\mathbb{C}^i$, where the function $\mathcal{G}^i$ serves as the distance measure. This projection yields the point $\underline{\Lambda}^i(\theta^{n+1}) \in \mathbb{C}^i$, and we have depicted its relation to an arbitrary $\mathbf{U}$ also in $\mathbb{C}^i$.

It is worthwhile comparing Figure 5.8 to Figure 5.7, which represent the same geometry in the two different co-ordinate systems. Figure 5.8 gives a picture of a single TRP update in terms of pseudomarginal vectors $\mathbf{T}$; in this co-ordinate system, the constraint set $\mathbb{C}^i$ is linear and hence illustrated as a plane. Figure 5.7 provides a similar picture in terms of the exponential parameters. The non-linear mapping $\Theta^i$ transforms this constraint set $\mathbb{C}^i$ to its analog $\mathbb{D}^i$ in the exponential domain (i.e., $\mathbb{D}^i = \Theta^i(\mathbb{C}^i)$). As a consequence of the non-linearity, the sets $\mathbb{D}^i$ in Figure 5.7 are represented by curved lines in exponential co-ordinates. In Figure 5.7, a single TRP update corresponds to moving along the straight line in exponential parameters between $\theta^n = \Theta(\mathbf{T}^n)$ and the point $\mathcal{Q}^{i(n)}(\theta^n)$ that belongs that $\mathbb{D}^i \cap \mathcal{M}(\theta^0)$. Conversely, in Figure 5.8, this same update is represented by moving along the curved line between $\mathbf{T}^n$ and $\underline{\Lambda}^i(\theta^{n+1}) = \mathcal{I}^i(\Lambda^i(\Pi^i(\mathcal{Q}^{i(n)}(\theta^n))))$.

## 5.4.4 Characterization of fixed points

Returning to the Euclidean projection example at the start of Section 5.4.3, consider again the problem of projecting $\mathbf{y} \in \mathbb{R}^n$ onto the linear constraint set $\mathcal{L} \subset \mathbb{R}^n$. Suppose that constraint set $\mathcal{L}$ can be decomposed as the intersection $\mathcal{L} = \cap_i \mathcal{L}^i$. Whereas it may be difficult to compute directly the projection $\widehat{\mathbf{x}} \in \mathcal{L}$, performing projections onto the larger linear constraint sets $\mathcal{L}^i$ is often easier. In this scenario, one possible strategy for finding the optimal projection $\widehat{\mathbf{x}} \in \mathcal{L}$ is to start at $\mathbf{y}$, and then perform a series of projections onto the constraint sets $\{\mathcal{L}^i\}$ in succession. In fact, such a sequence of projections is guaranteed [e.g., 31] to converge to the optimal approximation $\widehat{\mathbf{x}} \in \mathcal{L}$.

More generally, a wide class of algorithms can be formulated as successive projection techniques for minimizing a Bregman distance over a set formed by an intersection of linear constraints [e.g., 31]. An example that involves a Bregman distance other than the Euclidean norm is the generalized iterative scaling algorithm [52], used to compute projections involving the Kullback-Leibler divergence. A Pythagorean relation analogous equation (5.34) is instrumental in establishing the convergence of such techniques [31,45].

The problem of interest here is similar, since we are interested in finding a point belonging to a constraint set formed as an intersection of linear constraint sets (i.e., $\mathbb{C} = \cap_i \mathbb{C}^i$). However, the function $\mathcal{G}$ is certainly not a Bregman distance since, for instance, it can assume negative values. Nonetheless, the Pythagorean result in Proposition 5.4.2 allows us to show that any fixed point $\theta^*$ of the TRP algorithm satisfies the necessary conditions for it to be a local minimum of $\mathcal{G}(\mathbf{T}; \theta^0)$ over the constraint set $\mathbb{C}$. Although the result extends to other orderings, for concreteness we state it here for a *cyclic ordering* of spanning trees $\mathcal{T}^0, \ldots, \mathcal{T}^{L-1}$: i.e., the tree index for iteration $n$ is chosen as $i(n) = n (\mathrm{mod}\, L)$.

**Theorem 5.4.2 (Characterization of fixed points).** *Let $\theta^*$ be a fixed point of a sequence of iterates $\{\theta^n\}$ generated by equation (5.20) with a cyclic tree ordering, and using step sizes $\lambda^n \in [\epsilon, 1]$ for some $\epsilon > 0$.*

*(a) The point $\theta^*$ is fixed under all the tree operators $\mathcal{Q}^i$. (I.e., $\theta^* = \mathcal{Q}^i(\theta^*)$ for all indices $i = 0, \ldots, L-1$.) Therefore, each fixed point $\theta^*$ is associated with a unique pseudomarginal vector $\mathbf{T}^* \in \mathbb{C}$.*

*(b) The pseudomarginal vector $\mathbf{T}^*$ satisfies the necessary conditions for it to be a local minimum of $\mathcal{G}(\mathbf{T}; \theta^0)$ over the constraint set $\mathbb{C}$:*

$$\sum_{\alpha \in \mathcal{A}} \frac{\partial \mathcal{G}}{\partial T_\alpha}(\mathbf{T}^*; \theta^0) \left[\mathbf{U} - \mathbf{T}^*\right]_\alpha \quad = \quad 0$$

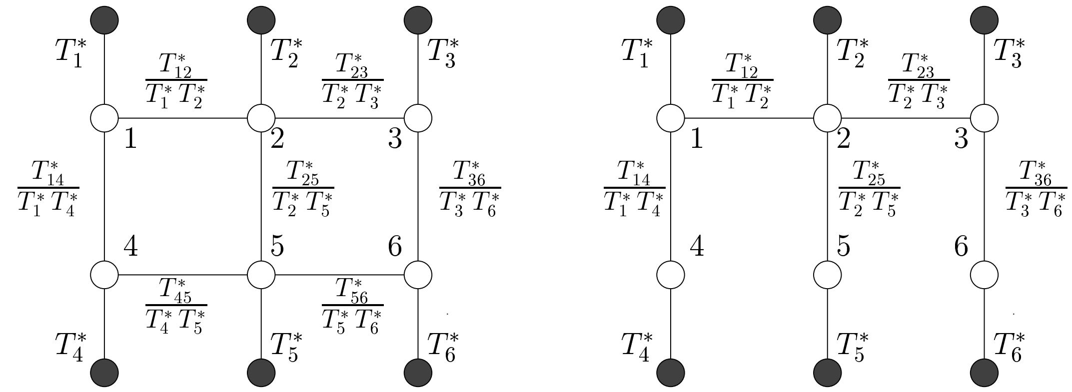
*for all $\mathbf{U}$ in the constraint set $\mathbb{C}$.*

*(c) Fixed points of the TRP algorithm always exist, and coincide with those of BP.*

*Proof.* See Appendix C.2. □

A few remarks about Theorem 5.4.2 are in order. First of all, to clarify the result stated in (a), the unique pseudomarginal vector $\mathbf{T}^*$ associated with $\theta^*$ can be constructed explicitly as follows. For an arbitrary index $\alpha$, pick a spanning tree $\mathcal{T}^i$ such that $\alpha \in \mathcal{A}^i$. Then define $T^*_\alpha = [\Lambda^i(\Pi^i(\theta^*))]_\alpha$; that is, $T^*_\alpha$ is the value of this (single node or pairwise) marginal for the tree-structured distribution $p(\mathbf{x}; \Pi^i(\theta^*))$. Note that this is a consistent definition of $T^*_\alpha$, because the condition of part (a) means that $[\Lambda^i(\Pi^i(\theta^*))]_\alpha$ is the same for all spanning tree indices $i \in \{0, \ldots, L-1\}$ such that $\alpha \in \mathcal{A}^i$. Moreover, this construction ensures that $\mathbf{T}^* \in \mathbb{C}$, since it must satisfy the normalization and marginalization constraints associated with every node and edge.

Figure 5.9 illustrates this characterization of fixed points in terms of $\mathbf{T}^*$. Shown



(a) (b)

**Figure 5.9.** Illustration of fixed point consistency condition. (a) Fixed point $\{T^*_s,\, T^*_{st}\}$ on the full graph with cycles. (b) Illustration of consistency condition on an embedded tree. The quantities $\{T^*_s,\, T^*_{st}\}$ must be a consistent set of marginal probabilities for any tree embedded within the full graph.

in panel (a) is an example of a graph $G$ with cycles, parameterized according to the approximate marginals $T^*_{st}$ and $T^*_s$. Consider the set of edges that must be removed from $G$ in order to form a given tree $\mathcal{T}$. For instance, Figure 5.9(b) shows the tree obtained by removing edges $(4,5)$ and $(5,6)$ from the graph in (a). Suppose that we remove the functions $T^*_{st}/(T^*_s T^*_t)$ that sit on non-tree edges, and use the remaining functions to form a tree-structured distribution as follows:

$$p^{\mathcal{T}}(\mathbf{x};\mathbf{T}^*) \quad = \quad \prod_{s\in V} T^*_s(x_s) \prod_{(s,t)\in E(\mathcal{T})} \frac{T^*_{st}(x_s,x_t)}{T^*_s(x_s)T^*_t(x_t)} \tag{5.35}$$

Here $E(\mathcal{T})$ is the edge set of the tree $\mathcal{T}$. The consistency condition of Theorem 5.4.2 then guarantees that the pseudomarginals $T^*_{st}$ and $T^*_s$ correspond to exact marginal distributions for the tree-structured distribution $p^{\mathcal{T}}(\mathbf{x};\mathbf{T}^*)$. In this case, we say that the pseudomarginal vector $\mathbf{T}^*$ is *tree-consistent* with respect to the tree $\mathcal{T}$.

In fact, this tree-consistency holds for *any* acyclic substructure embedded within the full graph with cycles — not just the spanning trees used to implement the algorithm. Thus, Theorem 5.4.2 provides an alternative and very intuitive view of BP or TRP: such algorithms attempt to reparameterize a distribution on a graph with cycles so that it is consistent with respect to each tree. In this regard, part (c) of Theorem 5.4.2 is noteworthy, in that it guarantees that any positive distribution on a graph can be reparameterized in terms of pseudomarginals $\mathbf{T}^*$ that satisfies the tree-based consistency condition of part (a). Although the existence of such a reparameterization is well-known for trees, it is by no means obvious for an arbitrary graph with cycles.

In the context of Theorem 5.4.2, the equivalence of TRP and BP fixed points follows because $\mathcal{G}$ and the Bethe free energy coincide on the constraint set $\mathbb{C}$. It is also possible to establish this equivalence in a more constructive manner. In particular, any fixed point $\{M^*_{st}\}$ of the message updates in equation (5.3) can be transformed into a pseudomarginal vector $\mathbf{T}^* = \{T^*_s,\, T^*_{st}\}$ as follows:

$$T^*_{s;j} \;=\; \kappa\, \psi_{s;j} \prod_{u \in \mathcal{N}(s)} M^*_{us;j} \tag{5.36a}$$

$$T^*_{st;jk} \;=\; \kappa\, \psi_{st;jk}\; \psi_{s;j}\; \psi_{t;k} \prod_{u \in \mathcal{N}(s)/t} M^*_{us;j} \prod_{u \in \mathcal{N}(t)/s} M^*_{ut;k} \tag{5.36b}$$

The message fixed point condition of equation (5.3) guarantees that the corresponding $\mathbf{T}^* = \{T^*_s, T^*_{st}\}$ satisfy the edgewise marginalization constraints necessary for membership in the constraint set $\mathbb{C}$. That is, elements of $\mathbf{T}^*$ are locally consistent with respect to all the simple two-node trees formed by single edges $(s,t) \in E$, as illustrated in Figure 5.4. From the junction tree perspective [102, 123], we know that local consistency of marginals on edges is equivalent to global consistency for trees. Therefore, the local consistency on edges defining membership in $\mathbb{C}$ implies that the vector $\mathbf{T}^*$ must be consistent on any singly-connected substructure embedded within $G$. As a consequence, the collection of pseudomarginals $\mathbf{T}^*$ specified by the message fixed point satisfies the tree-based consistency condition of part (a), as illustrated in Figure 5.9. Even more generally, a similar line of reasoning establishes that *any* constrained local minimum of the Bethe free energy, whether obtained by TRP/BP or an alternative minimization technique [e.g., 177, 184], can be identified with a pseudomarginal vector $\mathbf{T}^*$ satisfying the condition of Theorem 5.4.2.

## 5.4.5 Sufficient conditions for convergence for two spanning trees

Proposition 5.4.2 can also be used to derive a set of conditions that are sufficient to guarantee the convergence in the case of two spanning trees. To convey the intuition of the proof, suppose that it were possible to interpret the cost function $\mathcal{G}$ as a distance function. Moreover, suppose $\mathbf{U}$ were an arbitrary element of $\mathbb{C} = \cap_i \mathbb{C}^i$, so that we could apply Proposition 5.4.2 for each index $i$. Then equation (5.33) would show that the "distance" between $\theta^n$ and an arbitrary element $\mathbf{U} \in \mathbb{C}$, as measured by $\mathcal{G}$, decreases at each iteration. As with proofs on the convergence of successive projection techniques for Bregman distances [e.g., 31, 45], this property would allow us to establish convergence of the algorithm.

Of course, there are two problems with the use of $\mathcal{G}$ as a type of distance: it is not necessarily non-negative, and it is possible that $\mathcal{G}(\Lambda^i(\mathcal{Q}^i(\theta)); \theta) = 0$ for some $\theta \neq \mathcal{Q}^i(\theta)$. With respect to the first issue, we are able to show in general that an appropriate choice of step size will ensure the non-negativity of $\mathcal{G}(\Lambda^i(\mathcal{Q}^i(\theta)); \theta)$ (see Appendix C.4). The following result then states sufficient conditions (including assuming that the second problem does not arise along TRP trajectories) for convergence in the case of two spanning trees:

**Theorem 5.4.3.** *Consider the application of TRP with two spanning trees $\mathcal{T}^0$ and $\mathcal{T}^1$. Suppose that the sequence of iterates $\{\theta^n\}$ remains bounded, and that:*
*(a) for $i = 0,\ 1$, the condition $\mathcal{G}(\underline{\Lambda}^i(\mathcal{Q}^i(\theta^n)); \theta^n) \to 0$ implies that $[\mathcal{Q}^i(\theta^n) - \theta^n] \to 0$.*
*(b) there exists some integer $K$ such that the condition*

$$\mathcal{G}(\underline{\Lambda}^0(\mathcal{Q}^1(\theta^n)); \theta^n)\mathcal{G}(\underline{\Lambda}^1(\mathcal{Q}^0(\theta^n)); \theta^n) > 0$$

*holds for all $n \geq K$.*
*Then there exist choices of the step sizes $\lambda^n$ such that the sequence $\theta^n$ converges to some $\theta^*$ in the desired constraint set. I.e., $\theta^* = \mathcal{Q}^i(\theta^*)$ for $i = 0, 1$.*

*Proof.* See Appendix C.4, which includes a characterization of the step size choices that ensure convergence. □

This result, though its hypotheses cannot be checked a priori, provides some insight into the factors that cause failures of convergence when applying TRP/BP. In particular, the proof of Theorem 5.4.3 shows that assumption (a) is analogous to the gradient-relatedness condition of standard descent algorithms for nonlinear optimization [20].

## ■ 5.4.6 Implications for continuous processes

The reparameterization approach can be extended and has important implications for continuous processes as well. In particular, by extension to the Gaussian case, we obtain an elementary proof of a generalization to TRP of the result [152, 176] that when BP converges, the means are, in fact, correct. To establish this result, let us consider the Gaussian analog of TRP. For simplicity in notation, we treat the case of scalar Gaussian random variables at each node (though the ideas extend easily to the vector case). In the scalar Gaussian case, the approximate marginal distribution $T_s(x_s)$ at each node $s \in V$ is parameterized by a mean $\mu_s$ and variance $\sigma_s^2$. Similarly, the approximate joint distribution $T_{st}(x_s, x_t)$ can be parameterized by a mean vector $\nu_{st} \triangleq [\nu_{st;s} \;\; \nu_{st;t}]'$, and a covariance matrix. At each iteration, the edge $(s,t)$ is labeled with the edge function $T_{st}/\widetilde{T}_{st;s}\widetilde{T}_{st;t}$, where $\widetilde{T}_{st;s}(x_s) = \int_{-\infty}^{\infty} T_{st}(x_s, x_t)dx_t$ is the marginal distribution over $x_s$ induced by $T_{st}$. This edge function is parameterized by the mean vector $\nu_{st}$, and a quadratic form $A_{st} = [a_{st;s} \;\; a_{st} \;;\; a_{st} \;\; a_{st;t}]$. With this set-up, we have:

**Proposition 5.4.3.** *Consider the Gaussian analog of TRP or BP, and suppose that it converges. Then the computed means are exact, whereas in general the error covariances are incorrect.*

*Proof.* From the original problem specification, we have

$$-\log p(\mathbf{x}) \;\; = \;\; 1/2\ (\mathbf{x} - \widehat{\mu})^T\ P^{-1}\ (\mathbf{x} - \widehat{\mu}) + C \tag{5.37}$$

where $P^{-1}$ is the inverse covariance; $C$ is a constant independent of $\mathbf{x}$; and $\widehat{\mu}$ are the correct means on the graph with cycles.

We begin by noting that the Gaussian analog of Theorem 5.4.1 guarantees that this distribution will remain invariant under the reparameterization updates of TRP (or BP). At any iteration, the distribution is reparameterized in terms of $T_s$ and the edge functions as follows:

$$-\log p(\mathbf{x}) = \frac{1}{2}\sum_{(s,t)\in E}\left\{a_{st;s}(x_s-\nu_{st;s})^2 + 2a_{st}(x_s-\nu_{st;s})(x_t-\nu_{st;t}) + a_{st;t}(x_t-\nu_{st;t})^2\right\}$$
$$+\frac{1}{2}\sum_{s}(x_s-\mu_s)^2/\sigma_s^2 + C \tag{5.38}$$

Note that the pseudomarginal vector $\{T_s, T_{st}\}$ need not be consistent so that, for example, $\widetilde{T}_{st;s}(x_s)$ need not equal $T_s(x_s)$. However, suppose that TRP (or BP) converges so that these quantities are equal, which, in particular, implies that $\mu_s = \nu_{st;s}$ for all $(s,t)$ such that $t \in \mathcal{N}(s)$. That is, the means parameterizing the edge functions must agree with the means at the node marginals. In this case, equations (5.37) and (5.38) are two alternative representations of the same quadratic form, so that we must have $\widehat{\mu}_s = \mu_s$ for each node $s \in V$. Therefore, the means computed by TRP or BP must be exact. In contrast to the means, there is no reason to expect that the error covariances in a graph with cycles need be exact. □

It is worth remarking that there exist highly efficient techniques from numerical linear algebra (e.g., conjugate gradient [56]) for computing the means of a linear-Gaussian problem on a graph. Therefore, although TRP and BP compute the correct means (if they converge), there is little reason to apply them in practice. There remains, however, the interesting problem of computing correct error covariances at each node: we refer the reader to [174] for description of an embedded spanning tree method that efficiently computes both means and error covariances for a linear-Gaussian problem on a graph with cycles.

## ■ 5.4.7 When does TRP/BP yield exact marginals?

It is clear that the TRP/BP algorithm will yield the exact single node marginals of any $p(\mathbf{x})$ defined on a tree-structured graph. In this section, we address of the question of whether there exist particular problems on graphs with cycles for which a TRP/BP solution will be exact. If so, how large is the set of such problems? Theorems 5.4.2 and 5.4.1 provide the insights that are key to our analysis; these theorems place very severe restrictions on cases where TRP/BP fixed points can be exact.

Let $\mathbf{T}^* \in \mathbb{C}$ be a consistent fixed point of the TRP algorithm in the sense of Theorem 5.4.2. Let $P_s$ denote the actual marginals of the given distribution $p(\mathbf{x})$. We begin by defining two distinct notions of exactness:

**Definition 5.4.1 (Exactness).**

(a) The point $\mathbf{T}^*$ is *weakly exact* if all the single node marginals are correct. I.e.,

$$T^*_{s;j} = P_{s;j} \qquad \text{for all} \quad s \in V, \quad j = 0,1 \tag{5.39}$$

(b) The point $\mathbf{T}^*$ is *strongly exact* if all the marginals, both single node and pairwise, are correct. I.e., in addition to equation (5.39), we have

$$T^*_{st;jk} = P_{st;jk} \qquad \text{for all} \quad (s,t) \in E, \quad j,k \;=\; 0,1\ldots m-1 \tag{5.40}$$

The fixed point characterization of Theorem 5.4.2 provides a straightforward technique for constructing TRP/BP fixed points. In particular, we simply specify a distribution in terms of a vector $\mathbf{T}^* \in \mathbb{C}$ that is locally consistent (see equation 5.12) as follows:

$$p(\mathbf{x}; \mathbf{T}^*) \quad \propto \quad \prod_{s \in V} T^*_s(x_s) \prod_{(s,t)\in E} \frac{T^*_{st}(x_s, x_t)}{T^*_s(x_s) T^*_t(x_t)} \tag{5.41}$$

Since $\mathbf{T}^*$ belongs to $\mathbb{C}$, it is consistent on any spanning tree embedded within the graph (i.e., it belongs to the constraint set $\mathbb{C}^i$ corresponding to tree $\mathcal{T}^i$) and therefore is guaranteed to be a fixed point of the TRP updates. The invariance result of Theorem 5.4.1 guarantees that any distribution can be put into the form of equation (5.41) without altering the distribution. As a consequence, the question of exactness reduces to understanding when the $T^*_s$ and $T^*_{st}$ are equivalent to the corresponding marginal distributions of $P_s$ and $P_{st}$ of $p(\mathbf{x}; \mathbf{T}^*)$.

**Example 5.4.1 (Symmetric cases).** By exploiting the fact that TRP/BP updates preserve any symmetries in the problem, it is easy to develop symmetric examples that are weakly exact. For example, for a binary-valued vector $\mathbf{x}$ defined on any graph, let us specify a set of symmetric pseudomarginals as follows:

$$T^*_s \;=\; [0.5 \;\; 0.5]' \tag{5.42a}$$

$$T^*_{st} \;=\; \begin{pmatrix} \mu & 0.5-\mu \\ 0.5-\mu & \mu \end{pmatrix} \tag{5.42b}$$

where $\mu \in [0, 0.5]$ is arbitrary. It is clear that the corresponding vector $\mathbf{T}^*$ for any such problem instance is an element of $\mathbb{C}$, and a fixed point of TRP. Moreover, for such a choice of $\mathbf{T}^*$, symmetry considerations dictate that the actual single-node marginals of $p(\mathbf{x}; \mathbf{T}^*)$, formed as in equation (5.41), will be uniform $[0.5 \;\; 0.5]'$. Therefore, TRP/BP is weakly exact for any such problem instance.

We now investigate the relation between the joint pairwise pseudomarginals $T^*_{st}$, and the actual marginals $P_{st}$. From equation (5.36b), it can be seen that any pseudomarginal $T^*_{st}$ is always related to the original compatibility function $\psi_{st}$ via:

$$T^*_{st;jk} = \kappa \; \psi_{st;jk} \; \rho_{s;j} \rho_{t;k}$$

for some vectors $\rho_s$, $\rho_t$, and normalization constant $\kappa$. For any tree-structured distribution, a relation of this form also holds for the actual marginals.[9] Indeed, a tree-structured distribution is characterized by the property that the dependency between

[9] This is necessarily the case, since TRP/BP is exact for tree-structured problems.

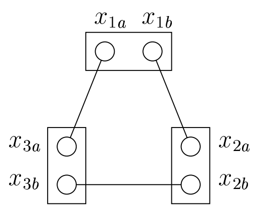



**Figure 5.10.** Degeneracy of compatibility functions for a 4-state process. Each random variable $x_s$ is decomposed into two random components $x_{sa}$ and $x_{sb}$, each of which is a binary random variable. These subcomponents are coupled by compatibility functions indicated in lines. Example courtesy of Tommi Jaakkola.

$x_s$ and $x_t$, for any pair of nodes $(s, t) \in E$, is mediated entirely by the compatibility function $\psi_{st}$. Indeed, if the compatibility function $\psi_{st}$ is removed, then $x_s$ and $x_t$ will be independent in the new distribution.

This intuition motivates us to define a notion of degenerate compatibility functions, for which TRP/BP will be exact for uninteresting reasons. To understand the idea of degeneracy, first consider a distribution $p(\mathbf{x})$ of a binary-valued vector $\mathbf{x}$ for which at least one compatibility function $\psi_{st}$ is rank one. I.e., $\psi_{st}$ be written as the outer product $\psi_{st} = \varphi_s \, \varphi_t'$ for a pair of 2−vectors $\varphi_s$ and $\varphi_t$. These 2−vectors can be absorbed into the single-node functions $\psi_s$ and $\psi_t$, so that edge $(s, t)$ can effectively be removed from the graph. Thus, for example, for a binary process, any distribution on a single cycle with at least one rank one compatibility function is equivalent to a tree-structured distribution.

The picture for $m$-ary processes is a bit more complicated. Here it is possible that all compatibility functions have a rank larger than one, and yet the overall distribution still exhibits degeneracies. Figure 5.10 illustrates a particular example of this degenerate behavior. The graph is a single cycle formed of three nodes, for which the associated random variables $\{\, x_s, \; s = 1, 2, 3 \,\}$ each assume 4 states. Each $x_s$ is decomposed into two random components $x_{sa}$ and $x_{sb}$, each of which is a binary random variable. The connections between the random variables $x_s$ are shown in solid lines. For instance, $x_{2b}$ is directly coupled to $x_{3b}$, but not to $x_{3a}$. None of these compatibility functions are rank one, so that the associated edges cannot be removed without altering the distribution. However, any distribution on this graph still has the property that removing the compatibility function between any pair of variables $x_u$ and $x_v$ leaves them independent in the new distribution.

These illustrative examples motivate the following:

**Definition 5.4.2 (Degeneracy).** A set of compatibility functions (or the associated distribution) is *degenerate* if for at least one $(s, t) \in E$, the compatibility function $\psi_{st}$ (viewed as a $m \times m$ matrix) has rank strictly less than $m$.

The significance of Definition 5.4.2 will become clear in the proof of Proposition 5.4.4, which relies on the fact for a non-degenerate distribution on a tree, two random variables $x_u$ and $x_v$ (for arbitrary $u, v \in V$) are never independent.[10] Therefore, given a non-degenerate distribution defined by a a single cycle, it is never possible to make a pair of random variables $x_u$ and $x_v$ independent by removing only a single edge.

**Proposition 5.4.4 (No exact pairwise marginals on single cycles).**
*Consider a distribution $p(\mathbf{x})$ of a binary-valued vector $\mathbf{x}$ defined by a set of non-degenerate compatibility functions on a single cycle. Let $\mathbf{T}^*$ be TRP/BP fixed point for this problem. Then* none *of the joint pairwise marginals are correct (i.e., $T^*_{st} \neq P_{st}$ for all $(s,t) \in E$).*

*Proof.* A remark on notation before proceeding: throughout this proof, we shall treat the quantities $T^*_{st}(x_s, x_t)$ as functions of $(x_s, x_t)$, rather than matrices. Let $(u, v) \in E$ be an arbitrary edge. By definition, we have:

$$P_{uv}(x_u, x_v) = \sum_{\{\mathbf{x}' \mid (x'_u, x'_v) = (x_u, x_v)\}} p(\mathbf{x}'; \mathbf{T}^*) \tag{5.43a}$$

$$= \kappa \left[ \sum_{(x'_u, x'_v) = (x_u, x_v)} \prod_{s \in V} T^*_s(x'_s) \prod_{(s,t) \in E \backslash (u,v)} \frac{T^*_{st}(x'_s, x'_t)}{T^*_s(x'_s) T^*_t(x'_t)} \right] \frac{T^*_{uv}(x_u, x_v)}{T^*_u(x_u) T^*_v(x_v)} \tag{5.43b}$$

where $\kappa$ is a normalization constant. The quantity within square brackets is the joint marginal distribution $\widehat{T}_{uv}(x_u, x_v)$ of a distribution structured according to the tree $\mathcal{T}$ specified by the subset of edges $E(\mathcal{T}) = E \backslash (u, v)$. For future reference, we refer to this

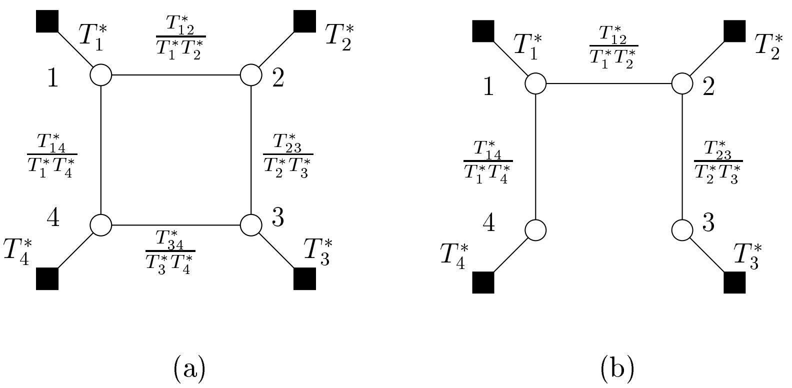



**Figure 5.11.** Relevant graphs for analyzing non-exactness of pairwise marginals on a single cycle. (a) Distribution $p(\mathbf{x}; \mathbf{T}^*)$ on graph $G$. (b) Tree-structured distribution $p(\mathbf{x}; \Pi^{\mathcal{T}}(\mathbf{T}^*))$ formed by removing edge $(u, v) = (3, 4)$.

[10]The same statement does *not* hold for a graph with cycles, as Example 5.4.2 will demonstrate.

tree-structured distribution as $p^{\mathcal{T}}(\mathbf{x}; \mathbf{T}^* x)$. We rewrite equation (5.43b) as:

$$P_{uv}(x_u, x_v) = \kappa\, \widehat{T}_{uv}(x_u, x_v)\, \frac{T^*_{uv}(x_u, x_v)}{T^*_u(x_u) T^*_v(x_v)} \tag{5.44}$$

We now proceed via proof by contradiction. If $P_{uv}(x_u, x_v) = T^*_{uv}(x_u, x_v)$ for all $x_u, x_v$, then equation (5.44) reduces to

$$\widehat{T}_{uv}(x_u, x_v) = T^*_u(x_u) T^*_v(x_v) \tag{5.45}$$

which implies that $x_u$ and $x_v$ are statistically independent under the tree-structured distribution $p^{\mathcal{T}}(\mathbf{x}; \mathbf{T}^*)$. This can occur only if at least one of the potential functions $T^*_{st}$ for $(s, t) \in E \backslash (u, v)$ is degenerate. This degeneracy contradicts the assumptions of the proposition, allowing us to conclude that $P_{uv} \neq T^*_{uv}$ for all $(u, v) \in E$. □

Interestingly, the proof of Proposition 5.4.4 is not valid for graphs with multiple cycles. The final step of the proof is based on the fact that for a tree-structured distribution, the condition of $x_u$ and $x_v$ being independent is equivalent to degeneracy (in the sense of Definition 5.4.2) of the compatibility functions.

This property is not true for graphs with cycles, so that if we try to extend Proposition 5.4.4 to graphs with multiple cycles, the proof breaks down at the final stage. Indeed, for a graph with cycles, it is possible to construct a distribution such that $x_u$ and $x_v$ are independent, even though all the compatibility functions in the graph are non-degenerate (i.e., full rank).

**Example 5.4.2.** In this example, we shall construct a family of problems for which the TRP pseudomarginals are correct for all single nodes, and for all but a single edge. I.e., the TRP solution is weakly exact, and strongly exact with the exception of a single edge.

Consider the 2-cycle graph shown in Figure 5.12(a), and a distribution $p(\mathbf{x}; \mathbf{T}^*)$ of a binary vector $\mathbf{x}$ parameterized in terms of a TRP fixed point $\mathbf{T}^*$. We specify the set of pseudomarginals as follows:

$$T^*_s = [0.5 \;\; 0.5]' \quad \text{for all } s \in V \tag{5.46a}$$

$$T^*_{st} = \begin{pmatrix} \mu & 0.5 - \mu \\ 0.5 - \mu & \mu \end{pmatrix} \quad \text{for all } (s, t) \in E \backslash (3, 4) \tag{5.46b}$$

$$T^*_{34} = \begin{pmatrix} \beta & 0.5 - \beta \\ 0.5 - \beta & \beta \end{pmatrix} \tag{5.46c}$$

The parameter $\mu \in (0, 0.5)$ is arbitrary; we shall specify $\beta \in (0, 0.5)$ as a function of $\mu$.

It can be seen that the vector $\mathbf{T}^*$ is an TRP fixed point, and is therefore consistent on any embedded tree. Consider now the following procedure for obtaining the exact marginals at nodes 3, 4, 5 and 6. We split the graph of Figure 5.12(a) into two subgraphs, as illustrated in panel (b). We marginalize the distribution over nodes 1 and 2, which yields a $2 \times 2$ matrix $Q_{34}$ (a function of $x_3$ and $x_4$) that is passed as a message to the remaining nodes in the lower subgraph.

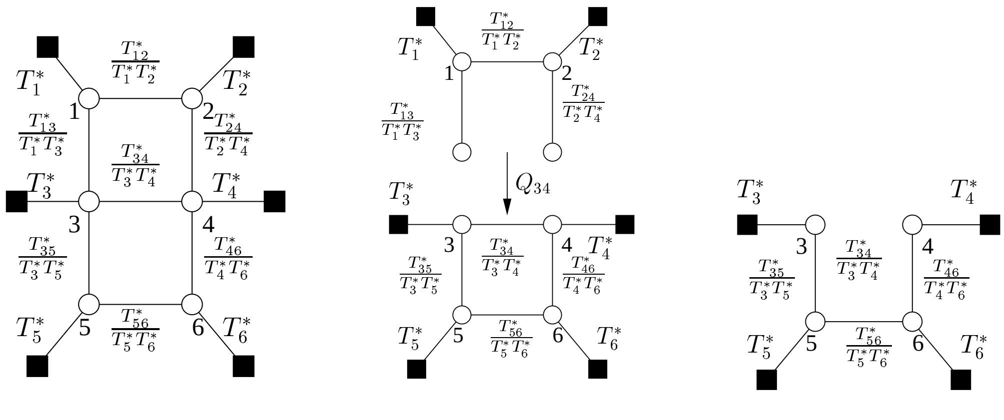
We now choose $\beta$, thereby specifying $T^*_{34}$, in such a way to ensure that:

$$Q_{34;jk} \; \frac{T^*_{34;jk}}{T^*_{3;j} T^*_{4;k}} = \kappa$$

where $\kappa$ is a normalization constant. Since the single node marginals are uniform, this is equivalent to $T^*_{34;jk} = \kappa'/Q_{34;jk}$, where $\kappa'$ is chosen so that the entries of $T^*_{34}$ sum to one.

With this choice of $T^*_{34}$, the quantity $Q_{34}$ effectively cancels out the compatibility function on edge $(3,4)$. To complete the computation of the actual marginals, we simply need to operate over the tree shown in Figure 5.12(c). The compatibility functions on this tree are already in standard form, so that the pseudomarginals $\{ \, T^*_s \mid s = 3,4,5,6 \, \}$ and $\{ \, T^*_{st} \mid (s,t) \in \{(3,5),(4,5),(5,6)\} \, \}$ must be equivalent to the exact marginals $P_s$ and $P_{st}$. By symmetry, a similar argument applies to the upper subgraph. Therefore, all of the single node marginals $T^*_s$ agree with the actual ones, and all of the joint pairwise marginals $T^*_{st}$, *except* $T^*_{34}$, are correct.

As a concrete numerical example, it can be verified that with $\mu = 0.4$, the choice $\beta = 0.196$ yields a graph with the above property. It makes intuitive sense that $\beta < 0.25$, since the compatibility function on edge $(3,4)$ serves to weaken the dependencies that build up between the other two indirect paths between 3 and 4.



(a) (b) (c)

**Figure 5.12.** Example of a graph and compatibility functions for which a TRP/BP solution is nearly exact. (a) Original graph parameterized in terms of a TRP fixed point $\mathbf{T}^*$. (b) Computing exact marginals on the graph by decomposing into two separate subgraphs. The top subgraph sends a message $Q_{34}$ to the bottom graph. We choose the compatibility functions to ensure that $Q_{34}\frac{T^*_{34}}{T^*_3 T^*_4}$ is a constant function. (c) The cancellation leaves us with a tree.

## 5.5 Analysis of the approximation error

An important but very difficult problem is analysis of the errors that arise from approximation techniques such as BP and TRP. To date only a few results, as described briefly in Section 5.1, are available regarding this error. Empirical simulations show that BP gives good approximations for certain graphs (e.g., those with long cycles relative to potential strength); in other cases, however, the approximations can be very poor. In the absence of error analysis, it is impossible to gauge the validity of the approximation. In this section, we provide a contribution to this important problem.

Our analysis of the approximation error is based on two fundamental properties of a fixed point $\theta^*$. First of all, part (a) of Theorem 5.4.2 dictates that for an arbitrary spanning tree $\mathcal{T}^i$, the single node elements $\theta^*_{s;j} = \log T^*_{s;j}$ correspond to a consistent set of marginal distributions on the spanning tree. That is, the quantities $T^*_{s;j}$ have two distinct interpretations:

(a) as the BP or TRP approximations to the exact marginals on the original graph with cycles.

(b) as the single node marginals of the tree-structured distribution $p^{\mathcal{T}}(\mathbf{x}; \mathbf{T}^*)$, as defined in equation (5.35), for any tree $\mathcal{T}$

Secondly, by the invariance stated in Theorem 5.4.1, the distribution $p(\mathbf{x}; \theta^*)$ induced by the fixed point $\theta^*$ is equivalent to the original distribution $p(\mathbf{x}; \theta^0)$. In conjunction, these two properties imply that the exact marginals on the full graph with cycles are related to the approximations $T^*_{s;j}$ by a relatively simply perturbation — namely, removing the functions $T^*_{st}/T^*_s T^*_t$ on edges not in the spanning tree. On this basis, we first derive an exact expression relating expectations under two different distributions, from which we proceed to derive lower and upper bounds on the approximation error.

In the development to follow, we will use the notation and perspective of the TRP algorithm. However, it should be noted that like Theorems 5.4.2 and 5.4.1, our analysis of the approximation error is again algorithm-independent. That is, it applies to any local minimum of the Bethe free energy, whether obtained by TRP/BP or an alternative minimization technique.

### 5.5.1 Exact expression

Our treatment begins at a slightly more general level, before specializing to the case of marginal distributions and the TRP algorithm. Consider a function $f : \mathcal{X}^N \to \mathbb{R}$, and two distributions $p(\mathbf{x}; \widetilde{\theta})$ and $p(\mathbf{x}; \theta)$. Suppose that we wish to express the expectation $\mathbb{E}_{\widetilde{\theta}}[f(\mathbf{x})]$ in terms of an expectation over $p(\mathbf{x}; \theta)$. Using the exponential representation of equation (5.6), it is straightforward to show that

$$\mathbb{E}_{\widetilde{\theta}}[f(\mathbf{x})] \quad = \quad \mathbb{E}_{\theta}\Big[ \; \exp\big\{ \sum_{\alpha} (\widetilde{\theta} - \theta)_{\alpha} \phi_{\alpha}(\mathbf{x}) + \Phi(\theta) - \Phi(\widetilde{\theta}) \big\} \; f(\mathbf{x}) \Big] \tag{5.47}$$

Note that this is a change of measure formula, where the exponentiated quantity can be viewed as the Radon-Nikodym derivative.

We now specialize equation (5.47) to the problem at hand. Let us denote the actual single node marginal $p(x_s = j; \theta^0)$ on the graph with cycles by

$$P_{s;j} \triangleq \mathbb{E}_{\theta^0}[\delta(x_s = j)] = \mathbb{E}_{\theta^*}[\delta(x_s = j)] \tag{5.48}$$

where $\delta(x_s = j)$ is the indicator function for node $x_s$ to take value $j$. To derive equation (5.48), we have used the invariance property (i.e., $p(\mathbf{x}; \theta^0) = p(\mathbf{x}; \theta^*)$) of Theorem 5.4.1. Assume that TRP (or BP) converges to some exponential parameter $\theta^*$, with associated pseudomarginal vector $\mathbf{T}^*$. In this case, Theorem 5.4.2 guarantees that the single-node elements of $\mathbf{T}^*$ can be interpreted as the following expectations:[11]

$$T^*_{s;j} \triangleq \mathbb{E}_{\Pi^i(\theta^*)}[\delta(x_s = j)] \tag{5.49}$$

We now make the assignments $\widetilde{\theta} = \theta^*$; $\theta = \Pi^i(\theta^*)$; and $f(\mathbf{x}) = \delta(x_s = j)$ in equation (5.47) and re-arrange to obtain

$$P_{s;j} - T^*_{s;j} = \mathbb{E}_{\Pi^i(\theta^*)}\Big[\big(\exp\big\{\sum_{\alpha \notin \mathcal{A}^i} \theta^*_\alpha \phi_\alpha(\mathbf{x}) - \Phi(\theta^*)\big\} - 1\big)\, \delta(x_s = j)\Big] \tag{5.50}$$

where we have used the fact that $\Phi(\Pi^i(\theta)) = 0$. Equation (5.50) is an exact expression for the error $(P_{s;j} - T^*_{s;j})$ in terms of an expectation over the tree-structured distribution $p(\mathbf{x}; \Pi^i(\theta^*))$. Note that equation (5.50) holds for all spanning tree indices $i \in \{0, \ldots, L-1\}$.

## 5.5.2 Error bounds

It is important to observe that equation (5.50), though conceptually interesting, is of limited practical use. The problem stems from the presence of the residual term

$$r^i(\mathbf{x}) \triangleq \exp\{\sum_{\alpha \notin \mathcal{A}^i} \theta^*_\alpha \phi_\alpha(\mathbf{x})\}$$

within the expectation on the right-hand side. For most problems, computing the expectation of $r^i(\mathbf{x})$ will not be tractable, since it is a function of all nodes $x_s$ incident with any edge removed to form spanning tree $\mathcal{T}^i$. Indeed, if the computation of equation (5.50) were easy for a particular graph, this would imply that we could compute the *actual* marginals, thereby obviating the need for an approximation technique such as BP/TRP.

This intractability motivates the idea of bounding the approximation error. In order to do so, we make use the bounds derived in Chapter 3, particularly Proposition 3.3.1.

[11] The tree-based consistency condition of Theorem 5.4.2 ensures that $T^*_{s;j} = \mathbb{E}_{\Pi^i(\theta^*)}[\delta(x_s = j)]$ independent of the choice of spanning tree index $i \in \{0, \ldots, L-1\}$.

In order to state these bounds on the approximation error, it is convenient to define a quantity that, for each spanning tree $\mathcal{T}^i$, captures part of the difference between the exact node marginal $P_{s;j}$ (corresponding to $p(\mathbf{x};\theta^*)$) and the approximation $T^*_{s;j}$ (corresponding to the tree-structured distribution $p(\mathbf{x};\Pi^i(\theta^*))$). Recall that the set $\{\phi_\alpha \mid \alpha \in \mathcal{A}\backslash\mathcal{A}^i\}$ corresponds to the set of potentials that were removed from the full graph $G$ in order to form the spanning tree $\mathcal{T}^i$. Accordingly, we define a quantity $\Delta^i_{s;j}$ as a weighted sum of covariance terms involving those potentials removed to form the tree distribution:

$$\Delta^i_{s;j} \triangleq \sum_{\alpha \in \mathcal{A}\backslash\mathcal{A}^i} \theta^*_\alpha \; \mathrm{cov}_{\Pi^i(\theta^*)}\{\delta(x_s = j),\, \phi_\alpha(\mathbf{x})\} \tag{5.51}$$

With this definition, we have the following error bounds on the approximation error:

**Theorem 5.5.1 (Error bounds).** *Let $\theta^*$ be a fixed point of TRP/BP, giving rise to approximate single-node marginals $T^*_{s;j}$, and let $P_{s;j}$ be the true marginal distributions on the graph with cycles. For each spanning tree index $i \in \{0, 1, \ldots, L-1\}$, the error $E_{s;j} \triangleq \log T^*_{s;j} - \log P_{s;j}$ is bounded above and below as follows:*

$$E_{s;j} \;\leq\; D(\Pi^i(\theta^*) \,\|\, \theta^*) - \frac{\Delta^i_{s;j}}{T^*_{s;j}} \tag{5.52a}$$

$$E_{s;j} \;\geq\; \log T^*_{s;j} - \log\Big[1 - (1 - T^*_{s;j})\exp\Big\{ - D(\Pi^i(\theta^*) \,\|\, \theta^*) - \frac{\Delta^i_{s;j}}{1 - T^*_{s;j}}\Big\}\Big] \tag{5.52b}$$

*Proof.* The bounds of this theorem follow by appropriately applying Proposition 3.3.1 from Chapter 3. We first make the identifications $\widetilde{\theta} = \theta^*$ and $\theta = \mathcal{I}^i(\Pi^i(\theta^*))$, and then set $f(\mathbf{x}) = \delta(x_s = j)$, a choice which satisfies the assumptions of Proposition 3.3.1. Equation (5.52a) then follows by applying Proposition 3.3.1, followed by some algebraic manipulation. The lower bound follows via the same argument applied to the function $f(\mathbf{x}) = 1 - \delta(x_s = j)$, which also satisfies the restrictions of Proposition 3.3.1. □

A number of remarks about Theorem 5.5.1 are in order. For practical purposes, the primary consideration is the cost of computing these lower and upper bounds. The terms in the summation defining $\Delta^i_{s;j}$ are tractable. In particular, each of the covariances can be calculated by taking expectations over tree-structured distributions, and their weighted summation is even simpler. On the other hand, within the KL divergence $D(\Pi^i(\theta^*) \,\|\, \theta^*)$ lurks a log partition function $\Phi(\theta^*)$ associated with the graph with cycles. In general, computing this quantity is as costly as performing inference on the original graph. To obtain computable bounds, we require an upper bound on the log partition function. In Chapter 7, we derive a set of such upper bounds, which allow us to compute upper bounds on the results of Theorem 5.5.1.

On the conceptual side, Theorem 5.5.1 highlights three factors that control the accuracy of the TRP/BP approximation. For the sake of concreteness, consider the upper bound of equation (5.52a):

(a) the covariance terms defining $\Delta^i_{s;j}$ reflect the strength of the interaction, as measured under the tree distribution $p(\mathbf{x}; \Pi^i(\theta^*))$, between the delta function $\delta(x_s = j)$ and the clique potential $\phi_\alpha(\mathbf{x})$. When the removed clique potential interacts only weakly with the delta function, then this covariance term will be small and so have little effect.

(b) the parameter $\theta^*_\alpha$ is the amount of weight placed on each clique potential $\phi_\alpha$ that was removed to form the spanning tree.

(c) the KL divergence $D(\Pi^i(\theta^*) \parallel \theta^*)$ measures the discrepancy between the tree-structured distribution $p(\mathbf{x}; \Pi^i(\theta^*))$ and the distribution $p(\mathbf{x}; \theta^*)$ on the graph with cycles. It will be small when the distribution $p(\mathbf{x}; \theta^*)$ is well-approximated by a tree. The presence of this term reflects the empirical finding that BP performs well on graphs that are approximately tree-like (e.g., graphs with fairly long cycles).

A number of extensions to the bounds presented in Theorem 5.5.1 are possible. First of all, it is worthwhile emphasizing that Theorem 5.5.1 provides $L$ bounds on the single-node marginal $P_{s;j}$ — one for each of the $L$ spanning trees used in the algorithm.[12] This allows us to choose the *tightest* of all the spanning tree bounds for a given index $(s; j)$. Point (a) above suggests that one should expect tighter bounds when using a tree formed by removing edges "relatively far" away from node $s$ of interest. Indeed, the covariance between $\delta(x_s = j)$ and the removed clique potential $\phi_\alpha$ captures this notion in a precise fashion.

Secondly, the change of measure expression in equation (5.47), as well as the methods used to prove Theorem 5.5.1, are both valid for more general choices of the function $f : \mathcal{X}^N \to [0, 1]$. Different choices will allow us to derive bounds on the error of other approximate marginals. For instance, making the choice $f(\mathbf{x}) = \delta(x_s = j)\ \delta(x_t = k)$ will lead to bounds on the pairwise marginal $P_{st;jk}$. Thirdly, note that the bounds of Theorem 5.5.1 are first-order, because they account for the interaction between the function $f$ and clique potentials $\phi_\alpha$ only up to first order (i.e., $\text{cov}_\theta\{f,\, \phi_\alpha\}$). On the basis of equation (5.50), it is possible to derive stronger bounds by including higher order terms (as in [e.g., 124]), though with an associated price of increased computation. A thorough analysis of various bounds and the inherent tradeoffs is open for future work.

## ■ 5.5.3 Illustrative examples of bounds

The tightness of the bounds given in Theorem 5.5.1 varies, depending on the graph topology, the choice of clique potentials, and the choice of spanning tree. In this section, we give some simple numerical illustrations. In all cases, we use the results of Chapter 7 to compute an upper bound on the log partition function $\Phi(\theta^*)$, so that the results of Theorem 5.5.1 are actually computable. Observe that bounds of Theorem 5.5.1 can be transformed into lower and upper bounds on the exact marginals. Specifically, a lower

[12] More generally, results of the form of Theorem 5.5.1 hold for any acyclic subgraph embedded within the graph, not just the spanning trees used to implement the algorithm.

bound (respectively upper bound) on the log error $E_{s;1} = \log T^*_{s;1} - \log P_{s;1}$ gives rise to an upper bound (respectively lower bound) on the actual marginal via the relation $P_{s;1} = T^*_{s;1} \exp\{-E_{s;1}\}$.

#### Varying the clique potentials

We first consider the choice of clique potentials. Figure 5.13 illustrates the behavior of TRP and the corresponding error bounds for a binary-valued process on the $3 \times 3$ grid shown in panel (a) for different settings of clique potentials. Shown in panels (b) through (d) are the actual marginals $P_{s;1} = p(x_s = 1; \theta^*)$ compared to the TRP approximations $T^*_{s;1}$ plotted on a node-by-node basis. We have also used the TRP/BP fixed point to compute lower and upper bounds on $P_{s;1}$; these bounds on the actual marginal are plotted in panels (b) through (d) as well.

Panel (b) illustrates the case of weak potentials, so that TRP /BP leads to the very accurate approximation of the exact marginals $P_{s;1}$. The gap between the corresponding lower and upper bounds on the exact marginals is narrow, which assures us that the approximation is excellent. Panel (c), in contrast, displays the more interesting choice of strong attractive clique potentials, for which TRP/BP approximations tend to be skewed towards an extreme value (one in this case). The gap between the upper and lower bounds on the exact marginals is large in comparison to those shown in panel (b). Despite the relative looseness of the bounds, note how the TRP/BP approximate marginals $T^*_{s;1}$ *exceed* the upper bounds[13] for certain nodes (in particular, nodes 5 through 9). Consequently, the error bounds inform us that the BP approximation is very inaccurate for these nodes.

Panel (d) displays the case of strong mixed potentials, where the TRP/BP approximation is again inaccurate. Once more, the TRP/BP approximation lies outside the window of bounds for the actual marginal for certain nodes (e.g., nodes 4,5,7,8).

#### Choice of spanning tree for bounds

As mentioned earlier, bounds of the form in Theorem 5.5.1 hold for any spanning tree (or more generally forest) embedded within the graph $G$. Here we show that the choice of spanning tree can also make a significant difference in the tightness of the bounds. Shown in panels (a) and (b) of Figure 5.14 are the actual marginals $P_{s;1}$ and TRP/BP approximations $T^*_{s;1}$ for a particular binary problem on the $3 \times 3$ grid. Note that the TRP/BP approximation is very good in this case.

As in the previous section, we also used the TRP/BP fixed point to calculate upper and lower bounds on the actual marginal $P_{s;1}$; here we investigated the effect of varying the spanning tree used to compute the bound. Our choice of spanning tree was based on the following heuristic. We first computed the minimal exponential parameter $\gamma^*$ (as in equation (5.25)) corresponding to the overcomplete representation $p(\mathbf{x}; \theta^*)$. We

[13]It is possible for the approximations to exceed the bounds, because the bounds are on the exact marginals $P_{s;1}$.

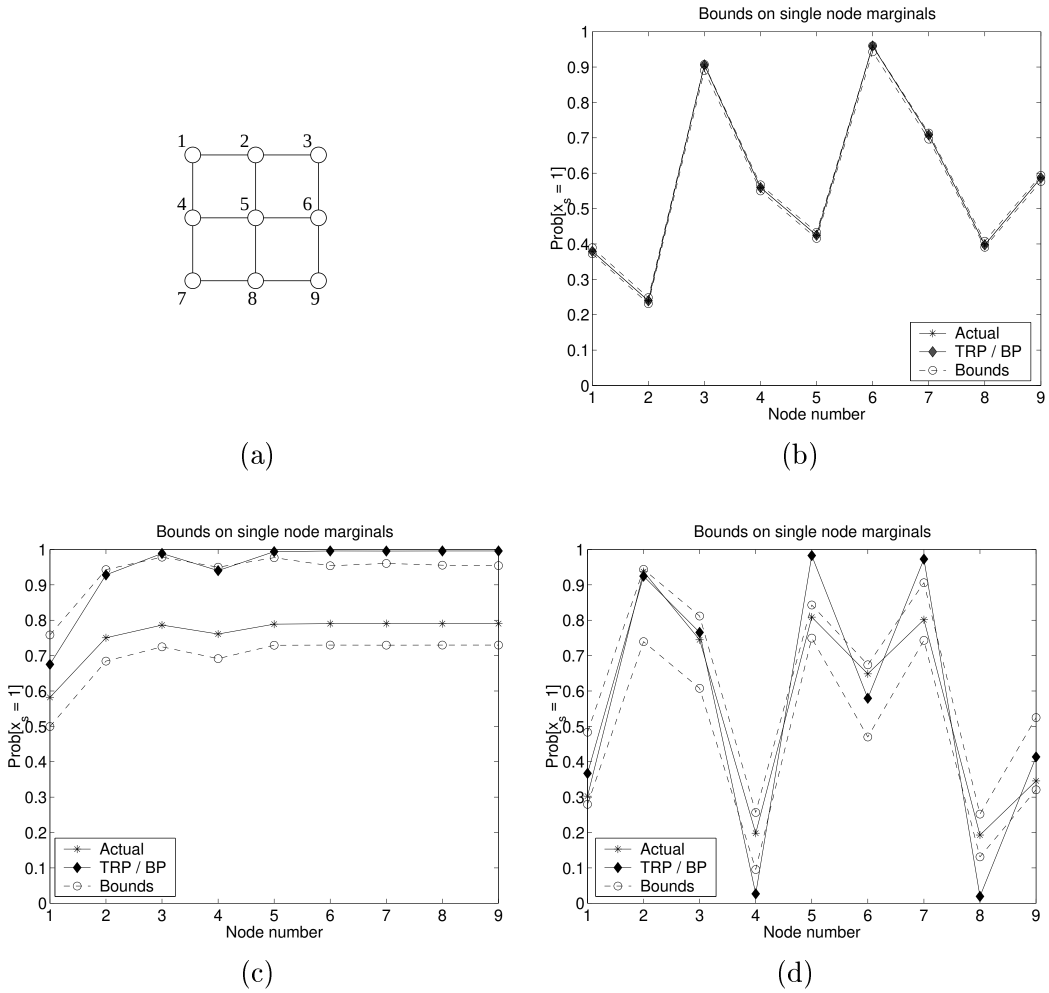
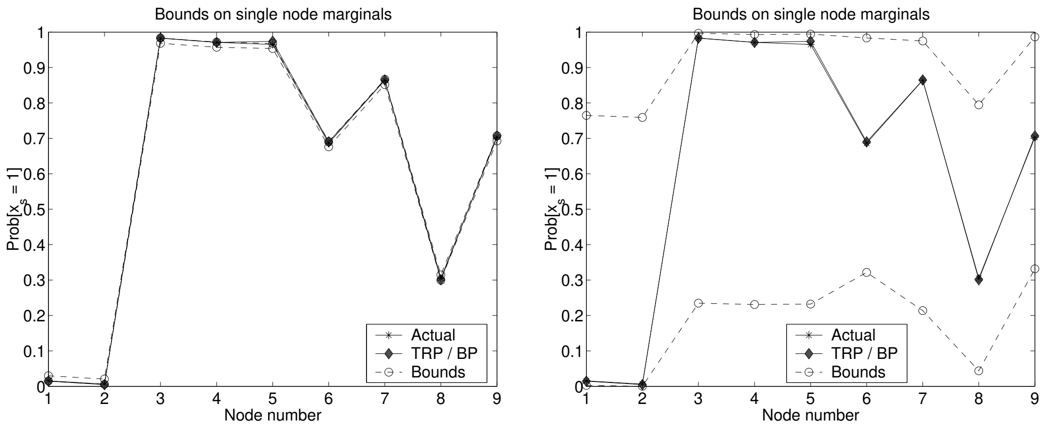



**Figure 5.13.** Behavior of the bounds of Theorem 5.5.1 for the $3 \times 3$ grid shown in panel (a) under various settings of clique potentials. Panels (b) through (d) show the actual marginals $P_{s;1}$ versus the TRP approximations $T^*_{s;1}$, as well as upper and lower bounds on the exact marginals. (b) For weak potentials, TRP gives a good approximation, and the gap between the lower and upper bounds on the exact marginals is very narrow. (c) For strong attractive potentials, the approximation is poor, and the gap becomes relatively large. (d) Similarly, the approximation is also poor for strong mixed potentials. Note how for certain nodes in (c) and (d), the TRP/BP approximation lies above the upper bounds on the actual marginal $P_{s;1}$.




(a) (b)

**Figure 5.14.** The narrowness of the bounds can depend strongly on the choice of spanning tree. Both panels show the exact marginals $P_{s;1}$ compared to the TRP/BP approximations $T^*_{s;1}$ on a node-by-node basis, as well as lower and upper bounds on the actual marginals. (a) The maximum weight spanning tree in the graph yields very narrow bounds. (b) The minimum weight spanning tree yields very poor bounds.

then computed the maximum and minimum weight spanning trees with Kruskal's algorithm [108,117], using $|\gamma^*_{st}|$ as the weight on edge $(s,t)$. Panels (a) and (b) show the lower and upper bounds on the exact marginals obtained from the TRP solution using the maximum and minimum weight spanning trees, respectively. The disparity between the two sets of bounds is striking. Further work should address techniques for choosing spanning trees that yield relatively tight bounds.

#### Uses of error analysis

The preceding examples serve to illustrate the general behavior of the bounds of Theorem 5.5.1 for some small problems, for which the bounds provided useful quantitative information. Computing these quantities requires upper bounding the log partition function $\Phi(\theta^*)$, using techniques from Chapter 7. For this reason, it is possible that for very large problems with strong potentials, the upper and lower bounds (e.g., on the exact marginal $P_{s;1}$) will tend towards to 1 and 0 respectively. In this case, they would no longer be quantitatively useful.

Nonetheless, the error analysis could still be useful in certain contexts. First of all, one direction to explore is the possibility of using our exact error analysis to derive, rather than upper and lower bounds, correction terms to the BP approximation. For instance, the term $\Delta^i_{s;j}$ defined in equation (5.51) is a computable first-order correction term to the BP approximation. Understanding when such corrections are useful is an interesting open problem. Secondly, several researchers [132,183] have proposed extensions to BP; we will analyze these techiques from the reparameterization perspective in

Chapter 6. Since the partition function is fixed regardless of the approximation (when specified in a minimal exponential representation), it plays no role in the relative error between the accuracy of different approximations. Therefore, these bounds could be useful in assessing when a more structured approximation, like generalized belief propagation [183], yields more accurate results.

## ■ 5.6 Discussion

This chapter developed a new conceptual framework for understanding belief propagation and related algorithms, based on their interpretation as performing a sequence of reparameterization updates. Each step of reparameterization entails specifying a new set of compatibility functions for an acyclic subgraph (i.e., tree) of full graph. The key property of these algorithms is that the original distribution is never changed, but simply reparameterized. The ultimate goal is to obtain a new factorization in which the functions on cliques represent approximations to the exact marginals. In particular, these pseudomarginals, which are computed by any reparameterization algorithm, must be consistent on every tree embedded within the graph with cycles. This tree consistency strongly constrains the nature of the approximation. The combination of the invariance result and the tree consistency enabled us to derive an exact expression for the error between the TRP/BP approximations and the exact marginals on an arbitrary graph with cycles. We also developed upper and lower bounds on this approximation error, and provided illustrations of their behavior. In conjunction with the results of Chapter 7, these error bounds are computable, and thus provide potentially useful information on the performance of TRP/BP.

The theoretical results of this chapter followed very naturally from the perspective of tree-based reparameterization. However, it should be noted that most of these results — most importantly, the invariance and tree consistency of the fixed points, as well as the associated error analysis — are, in fact, algorithm-independent. To be precise, the same results apply to any local minimum of the Bethe free energy, regardless of the algorithm [e.g., 177, 184] used to find it. Moreover, this chapter focused exclusively on reparameterization algorithms which involved only singleton and pairwise cliques. However, as we will see in Chapter 6, the ideas and results from this chapter can be extended to more advanced approximation techniques that either operate over larger cliques [183], or make use of more complex approximating structures [132]. We shall extend the same fixed point characterization, invariance and error analysis to these higher-order reparameterization algorithms.