Stochastic Belief Propagation: A Low-Complexity Alternative to the Sum-Product Algorithm

Nima Noorshams, Martin J. Wainwright, *Senior Member, IEEE*.

***Abstract*—The belief propagation (BP) or sum-product algorithm is a widely-used message-passing method for computing marginal distributions in graphical models. At the core of the BP message updates, when applied to a graphical model involving discrete variables with pairwise interactions, lies a matrix-vector product with complexity that is quadratic in the state dimension d, and requires transmission of a $(d-1)$-dimensional vector of real numbers (messages) to its neighbors. Since various applications involve very large state dimensions, such computation and communication complexities can be prohibitively complex. In this paper, we propose a low-complexity variant of BP, referred to as stochastic belief propagation (SBP). As suggested by the name, it is an adaptively randomized version of the BP message updates in which each node passes randomly chosen information to each of its neighbors. The SBP message updates reduce the computational complexity (per iteration) from quadratic to linear in d, without assuming any particular structure of the potentials, and also reduce the communication complexity significantly, requiring only $\log_2 d$ bits transmission per edge. Moreover, we establish a number of theoretical guarantees for the performance of SBP, showing that it converges almost surely to the BP fixed point for any tree-structured graph, and for any graph with cycles satisfying a contractivity condition. In addition, for these graphical models, we provide non-asymptotic upper bounds on the convergence rate, showing that the ℓ_∞ norm of the error vector decays no slower than $\mathcal{O}(1/\sqrt{t})$ with the number of iterations t on trees and the normalized mean-squared error decays as $\mathcal{O}(1/t)$ for general graphs. This analysis, also supported by experimental results, shows that SBP can provably yield reductions in computational and communication complexities for various classes of graphical models.**[1]

Keywords: Graphical models; sum-product algorithm; low-complexity belief propagation; randomized algorithms; stochastic approximation.

I. Introduction

Graphical models provide a general framework for describing statistical interactions among large collections of random variables. A broad range of fields—among them error-control coding, communication theory, statistical signal processing, computer vision, and bioinformatics—involve problems that can be fruitfully tackled using the formalism of graphical models. A computational problem central to such applications is that of *marginalization*, meaning the problem of computing marginal distributions over a subset of random variables. Naively approached, these marginalization problems have exponential complexity, and hence are computationally intractable. Therefore, graphical models are only useful when combined with efficient algorithms. For graphs without cycles, the marginalization problem can be solved exactly and efficiently via an algorithm known as the belief propagation (BP) algorithm or sum-product algorithm. It is a distributed algorithm, in which each node performs a set of local computations, and then relays the results to its graph neighbors in the form of so-called messages. For graphs with cycles, the BP algorithm is no longer an exact method, but nonetheless is widely used and known to be extremely effective in many settings. For a more detailed discussion of the role of the marginalization problem and the use of the BP algorithm, we refer the reader to various overview papers (e.g., [17], [18], [32], [2]).

In many applications of BP, the messages themselves are high-dimensional in nature, either due to discrete random variables with a very large number of possible realizations d (which will be referred to as the number of states), due to factor nodes with high degree, or due to continuous random variables that are discretized. Examples of such problems include disparity estimation in computer vision, tracking problems in sensor networks, and error-control decoding. For such problems, it may be expensive to compute and/or store the messages, and as a consequence, BP may run slowly, and be limited to small-scale instances. Motivated by this challenge, researchers have studied a variety of techniques to reduce the complexity of BP in different applications (e.g., see the papers [9], [27], [19], [14], [15], [6], [26] and references therein). At the core of the BP message-passing is a matrix-vector multiplication, with complexity scaling quadratically in the number of states d. Certain graphical models have special structures that can be exploited so as to reduce this complexity. For instance, in when applied to decode low-density parity-check codes for channel coding (e.g., [10], [17]), the complexity of message-passing, if performed naively, would scale exponentially in the factor degrees. However, a clever use of the fast Fourier transform

N. Noorshams is with the Department of EECS at the University of California, Berkeley. M. J. Wainwright is with the Departments of Statistics and EECS, University of California, Berkeley.

[1]Portions of the results given here were initially reported at the Allerton Conference on Communications, Control, and Computing (September 2011).

over GF(2^r) reduces this complexity to linear in the factor degrees (e.g., see the paper [25] for details). Other problems arising in computer vision involve pairwise factors with a circulant structure for which the fast Fourier transform can also reduce complexity [9]. Similarly, computation can be accelerated by exploiting symmetry in factors [15], or additional factorization properties of the distribution [19].

In the absence of structure to exploit, other researchers have proposed different types of quantization strategies for BP message updates [6], [14], as well as stochastic methods based on particle filtering or non-parametric belief propagation (e.g., [3], [27], [7]) that approximate continuous messages by finite numbers of particles. For certain classes of these methods, it is possible to establish consistency as the number of particles tends to infinity [7], or to establish non-asymptotic results inversely proportional to the square root of the number of particles [13]. As the number of particles diverges, the approximation error becomes negligible, a property that underlies such consistency proofs. Researchers have also proposed stochastic techniques to improve the decoding efficiency of binary error-correcting codes [30], [21]. These techniques, which are based on encoding messages with sequences of Bernoulli random variables, lead to efficient decoding hardware architectures.

In this paper, we focus on the problem of implementing BP in high-dimensional discrete spaces, and propose a novel low-complexity algorithm, which we refer to as *stochastic belief propagation* (SBP). As suggested by its name, it is an adaptively randomized version of the BP algorithm, where each node only passes randomly selected partial information to its neighbors at each round. The SBP algorithm has two features that make it practically appealing. First, it reduces the computational cost of BP by an order of magnitude; in concrete terms, for arbitrary pairwise potentials over d states, it reduces the per iteration computational complexity from quadratic to linear—that is, from $\Theta(d^2)$ to $\Theta(d)$. Second, it significantly reduces the message/communication complexity, requiring transmission of only $\log_2 d$ bits per edge as opposed to $(d-1)$ real numbers in the case of BP.

Even though SBP is based on low-complexity updates, we are able to establish conditions under which it converges (in a stochastic sense) to the exact BP fixed point, and moreover, to establish quantitative bounds on this rate of convergence. These bounds show that SBP can yield provable reductions in the complexity of computing a BP fixed point to a tolerance $\delta > 0$. In more precise terms, we first show that SBP is strongly consistent on any tree-structured graph, meaning that it converges almost surely to the unique BP fixed point; in addition, we provide non-asymptotic upper bounds on the ℓ_∞ norm (maximum value) of the error vector as a function of iteration number (Theorem 1). For general graphs with cycles, we show that when the ordinary BP message updates satisfy a type of contraction condition, then the SBP message updates are strongly consistent, and converge in normalized mean-squared error at the rate $\mathcal{O}(1/t)$ to the unique BP fixed point, where t is the number of iterations. We also show that the typical performance is sharply concentrated around its mean (Theorem 2). These theoretical results are supported by simulation studies, showing the convergence of the algorithm on various graphs, and the associated reduction in computational complexity that is possible.

The remainder of the paper is organized as follows. We begin in Section II with background on graphical models as well as the BP algorithm. In Section III, we provide a precise description of the SBP, before turning in Section III-B to statements of our main theoretical results, as well as discussion of some of their consequences. Section IV is devoted to the proofs of our results, with more technical aspects of the proofs deferred to the Appendices. In Section V, we demonstrate the correspondence between our theoretical predictions and the algorithm's practical behavior.

II. Background

In this section, we provide some background on graphical models as well as the belief propagation algorithm.

A. Graphical Models

Consider a random vector $X := \{X_1, X_2, \ldots, X_n\}$, where for each $v = 1, 2, \ldots, n$, the variable X_v takes values in some discrete space $\mathcal{X} := \{1, 2, \ldots, d\}$ with cardinality d. An undirected graphical model, also known as a Markov random field, defines a family of joint probability distributions over this random vector by associating the index set $\{1, 2, \ldots, n\}$ with the vertex set $\mathcal{V}$ of an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In addition to the vertex set, the graph consists of a collection of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, where a pair $(v, u) \in \mathcal{E}$ if and only if nodes u and v are connected by an edge.[2] The structure of the graph describes the statistical dependencies among the different random variables—in particular, via the cliques[3] of the graph. For each clique I of the graph, let $\psi_I : \mathcal{X}^{|I|} \to (0, \infty)$ be a function of the sub-vector $X_I := \{X_v,\ v \in I\}$ of the random variables indexed by the clique, and then consider the set of all distributions over X that factorize as

$$\mathbb{P}(x_1, \ldots, x_n) \ \propto \prod_{I \in \mathcal{C}} \psi_I(x_I), \tag{1}$$

where $\mathcal{C}$ is the set of all cliques in the graph.

As a concrete example, consider the two-dimensional grid shown in Figure 1(a). Since its cliques consist of the set of all vertices $\mathcal{V}$ together with the set of all edges $\mathcal{E}$, the general factorization (1) takes the special form

$$\mathbb{P}(x_1, \ldots, x_n) \ \propto \prod_{v \in \mathcal{V}} \psi_v(x_v) \prod_{(v,u) \in \mathcal{E}} \psi_{vu}(x_v, x_u), \tag{2}$$

[2]It should be noted that by (v, u) we mean an unordered tuple of vertices.

[3]A clique I of a graph is a subset of vertices that are all joined by edges, and so form a fully connected subgraph.

where $\psi_v : \mathcal{X} \to (0, \infty)$ is the node potential function for node v, and $\psi_{vu} : \mathcal{X} \times \mathcal{X} \to (0, \infty)$ is the edge potential function for the edge (v, u). A factorization of this form (2) is known as a *pairwise Markov random field*. It is important to note that there is no loss of generality in assuming a pairwise factorization of this form; indeed, any graphical model with discrete random variables can be converted into a pairwise form by suitably augmenting the state space (e.g., see Yedidia et al. [33] or Wainwright and Jordan [32], Appendix E.3). Moreover, the BP message updates can be easily translated from the original graph to the pairwise graph, and vice versa. Accordingly, for the remainder of this paper, we focus on the case of a pairwise MRF.

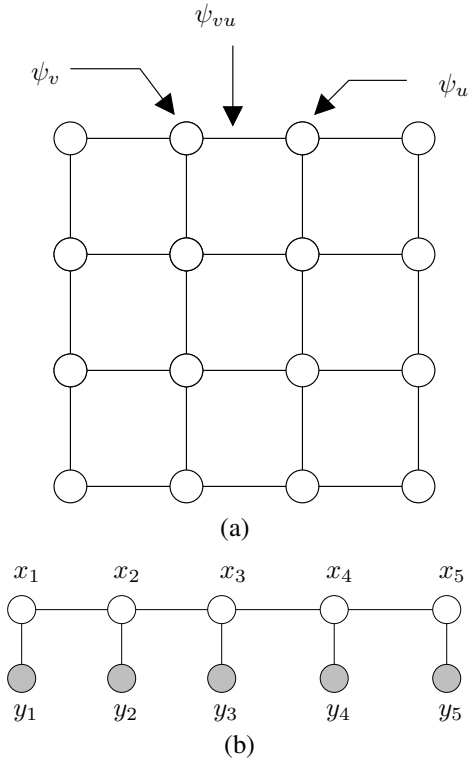


Fig. 1. Examples of pairwise Markov random fields. (a) A two-dimensional grid: the potential functions ψ_u and ψ_v are associated with nodes u and v, respectively, whereas the potential function ψ_{vu} is associated with edge (v, u). (b) Hidden Markov model including both hidden variables $(x_1, \ldots, x_5)$, represented as white nodes, and observed variables $(y_1, \ldots, y_5)$, represented as shaded nodes.

In various application contexts, the random vector $(X_1, \ldots, X_n)$ is an unobserved or "hidden" quantity, and the goal is to draw inferences on the basis of a collection of observations $(Y_1, \ldots, Y_n)$. The link between the observed and hidden variables is specified in terms of a conditional probability distribution, which in many cases can be written in the product form $\mathbb{P}(y \mid x) = \prod_{v=1}^{n} \mathbb{P}(y_v \mid x_v)$. For instance, in error-control coding using a low-density parity check code, the vector X takes values in a linear subspace of $GF(2)^n$, corresponding to valid codewords, and the observation vector Y is obtained from some form of memoryless channel (e.g., binary symmetric, additive white Gaussian noise, etc.). In image denoising applications, the vector X represents a rasterized form of the image, and the observation Y corresponds to a corrupted form of the image.

In terms of drawing conclusions about the hidden variables based on the observations, the central object is the posterior distribution $\mathbb{P}(x \mid y)$. From the definition of the conditional probability and the form of the prior and likelihoods, this posterior can also be factorized in pairwise form

$$\begin{aligned}\mathbb{P}(x \mid y) \;&\propto\; \mathbb{P}(x_1, \ldots, x_n) \prod_{v=1}^{n} \mathbb{P}(y_v \mid x_v) \\ &= \prod_{v \in \mathcal{V}} \widetilde{\psi}_v(x_v) \prod_{(v,u) \in \mathcal{E}} \psi_{vu}(x_v, x_u),\end{aligned}$$

where $\widetilde{\psi}_v(x_v) := \psi_v(x_v)\mathbb{P}(y_v \mid x_v)$ is the new node compatibility function. (Since the observation y_v is fixed, there is no need to track its functional dependence.) Thus, the problem of computing marginals for a posterior distribution can be cast[4] as an instance of computing marginals for a pairwise Markov random field (2).

Our focus in this paper is on the *marginalization problem*, meaning the computation of the single-node marginal distributions

$$\mathbb{P}(x_v) \;:=\; \sum_{\{x' \mid x'_v = x_v\}} \mathbb{P}\,(x'_1, \ldots, x'_n) \quad \text{for each } v \in \mathcal{V}, \tag{3}$$

and more generally, higher-order marginal distributions on edges and cliques. Note that to calculate this summation, brute force is not tractable and requires nd^{n-1} computations. For any graph without cycles—known as a tree—this computation can be carried far more efficiently in only $\mathcal{O}(nd^2)$ operations using an algorithm known as the belief propagation, to which we now turn.

B. Belief Propagation Algorithm

Belief propagation, is an iterative algorithm consisting of a set of local message-passing rounds, for computing either exact or approximate marginal distributions. For tree-structured (cycle-free) graphs, it is known that BP message-based marginals converge to the exact marginals in a finite number of iterations. However, the same message-passing updates can also be applied to more general graphs, and are known to be effective for computing approximate marginals in numerous applications. Here we provide a very brief treatment, referring the reader to various standard sources [17], [2], [33], [32] for further background.

[4]For illustrative purposes, we have assumed here that the distribution $\mathbb{P}(y \mid x)$ has a product form, but a somewhat more involved reduction also applies to a general observation model.

In order to define the message-passing updates, we require some further notation. For each node $u \in \mathcal{V}$, let $\mathcal{N}(u) := \{v \mid (v,u) \in \mathcal{E}\}$ denote its set of neighbors, and let $\vec{\mathcal{E}}(u) := \{(v \leftarrow u) \mid v \in \mathcal{N}(u)\}$ denote the set of all directed edges emanating from u. Finally, we define $\vec{\mathcal{E}} := \cup_{u \in \mathcal{V}} \vec{\mathcal{E}}(u)$, the set of *all directed edges* in the graph; note that $\vec{\mathcal{E}}$ has cardinality $2|\mathcal{E}|$. In the BP algorithm, one message $m_{vu} \in \mathbb{R}^d$ is assigned to every directed edge $(v \leftarrow u) \in \vec{\mathcal{E}}$. By concatenating all of these d-dimensional vectors, one for each of the $2|\mathcal{E}|$ members of $\vec{\mathcal{E}}$, we obtain a D-dimensional vector of messages $m = \{m_{vu}\}_{(v \leftarrow u) \in \vec{\mathcal{E}}}$, where $D := 2|\mathcal{E}|d$.

At each round $t = 0, 1, 2, \ldots$, every node $u \in \mathcal{V}$ calculates a message $m_{vu}^{t+1} \in \mathbb{R}^d$ to be sent to its neighbor $v \in \mathcal{N}(u)$. In mathematical terms, this operation can be represented as an update of the form $m_{vu}^{t+1} = F_{vu}(m^t)$ where $F_{vu} : \mathbb{R}^D \to \mathbb{R}^d$ is the local update function of the directed edge $(v \leftarrow u)$. In more detail, for each $x_v \in \mathcal{X}$, we have[5]

$$\begin{aligned} m_{vu}^{t+1}(x_v) \;&=\; [F_{vu}(m^t)](x_v) \\ &=\; \kappa \sum_{x_u \in \mathcal{X}} \Big(\psi_{vu}(x_v, x_u) \psi_u(x_u) \prod_{w \in \mathcal{N}(u) \setminus \{v\}} m_{uw}^t(x_u) \Big), \end{aligned} \tag{4}$$

where κ is a normalization constant chosen to ensure that $\sum_{x_v} m_{vu}^{t+1}(x_v) = 1$. Figure 2(a) provides a graphical representation of the flow of information in this local update.

By concatenating the local updates (4), we obtain a global update function $F : \mathbb{R}^D \to \mathbb{R}^D$ of the form

$$F(m) \;=\; \{F_{vu}(m)\}_{(v \leftarrow u) \in \vec{\mathcal{E}}}. \tag{5}$$

Typically, the goal of message-passing is to obtain a *fixed point*, meaning a vector $m^* \in \mathbb{R}^D$ such that $F(m^*) = m^*$ and (4) can be seen as an iterative way of solving this fixed-point equation. For any tree-structured graph, it is known that the update (5) has a unique fixed point. For a general graph (with some mild conditions on the potentials; see Yedidia et al. [33] for details), it is known that the global update (5) has at least one fixed point, but it is no longer unique in general. However, there are various types of contraction conditions that can be used to guarantee uniqueness on a general graph (e.g., [29], [12], [20], [23]).

Given a fixed point m^*, node v computes its marginal (approximation) τ_v^* by combining the local potential function ψ_v with a product of all incoming messages as

$$\tau_v^*(x_v) \;=\; \kappa\, \psi_v(x_v) \prod_{u \in \mathcal{N}(v)} m_{vu}^*(x_v), \tag{6}$$

where κ is a normalization constant chosen so that $\sum_{x_v \in \mathcal{X}} \tau_v^*(x_v) = 1$. See Figure 2(b) for an illustration of this computation. For any tree-structured graph, the quantity $\tau_v^*(x_v)$ is equal to the single-node marginal $\mathbb{P}(x_v)$, as previously defined (3). For a graph with cycles, the vector τ_v^* represents an approximation to the single-node marginal, and is known to be a useful approximation for many classes of graphical models.

[5]It is worth mentioning that m_{vu}^{t+1} is only a function of the messages m_{uw}^t for $w \in \mathcal{N}(u)\setminus\{v\}$. Therefore, we have $F_{vu} : \mathbb{R}^{(\rho_u - 1)d} \to \mathbb{R}^d$, where ρ_u is the degree of the node u. Since it is clear from the context and for the purpose of reducing the notation overhead, we say $m_{vu}^{t+1} = F_{vu}(m^t)$ instead of $m_{vu}^{t+1} = F_{vu}(\{m_{uw}^t\}_{w \in \mathcal{N}(u)\setminus\{v\}})$.

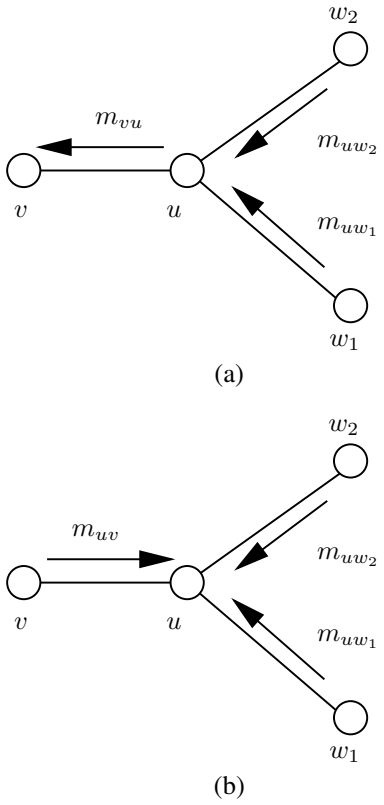


Fig. 2. Graphical representation of message-passing algorithms. (a) Node u transmits the message $m_{vu} = F_{vu}(m)$, derived from (4), to its neighbor v. (b) Upon receiving all the messages, node u updates its marginal estimate.

III. Algorithm and Main Results

We now turn to a description of the SBP algorithm (Section III-A), as well as the statement of our main theoretical guarantees on its behavior (Section III-B).

A. Stochastic Belief Propagation

When applied to a pairwise graphical model with random variables taking d states, the number of summations and multiplications required by the original BP algorithm is $\Theta(d^2)$ per iteration and per edge as can be seen by inspection of the message update equation (4). This quadratic complexity—which is incurred on a per iteration, per edge basis—is prohibitive in many applications, where the state dimension may be on the order of thousands. As discussed earlier in Section I, although certain graphical

models have particular structures that can be exploited to reduce the complexity of the updates, not all problems have such special structures, so that a general-purpose approach is of interest. In addition to computational cost, a standard BP message update can also be expensive in terms of communication cost, since each update requires transmitting $(d-1)$ real numbers along each edge. For applications that involve power limitations, such as sensor networks, reducing this communication cost is also of interest.

Stochastic belief propagation is an adaptively randomized form of the usual BP message updates that yields savings in both computational and communication cost. It is motivated by a simple observation—namely, that the message-passing update along the directed edge $(v \leftarrow u)$ can be formulated as an expectation over suitably normalized columns of a compatibility matrix (see (7)). Here the probability distribution in question depends on the incoming messages, and changes from iteration to iteration. This perspective leads naturally to an *adaptively randomized variant* of BP: instead of computing and transmitting the full expectation at each round—which incurs $\Theta(d^2)$ computational cost and requires sending $\Theta(d)$ real numbers—the SBP algorithm simply picks a single normalized column with the appropriate (message-dependent) probability, and performs a randomized update. As we show, each such operation can be performed in $\Theta(d)$ time and requires transmitting only $\log_2 d$ bits, so that the SBP message updates are less costly by an order of magnitude.

With this intuition in hand, we are now ready for a precise description of the SBP algorithm. Let us view the edge potential function ψ_{vu} as a matrix of numbers $\psi_{vu}(i,j)$, for $i,j = 1,\ldots,d$. For the directed edge $(v \leftarrow u)$, define the collection of column vectors[6]

$$\Gamma_{vu}(:,j) \;:=\; \frac{\psi_{vu}(:,j)}{\sum_{i=1}^{d} \psi_{vu}(i,j)}, \tag{7}$$

and marginal weights $\beta_{vu}(j) := \big(\sum_{i=1}^{d} \psi_{vu}(i,j)\big)\, \psi_u(j)$, for $j = 1,2,\ldots,d$. We assume that the column vectors $\Gamma_{vu}(:,j)$ and normalization constants $\beta_{vu}(j)$ have been pre-computed and stored, which can be done in an off-line manner and requires $\Theta(d^2)$ operations. In addition, the algorithm makes use of a positive sequence of step sizes $\{\lambda^t\}_{t=0}^{\infty}$. In terms of these quantities, the SBP algorithm consists of the steps shown in Figure 3.

The per iteration per edge computational complexity of the SBP algorithm lies in calculating the probability mass function p_{vu}, defined in (9); generating a random index J_{vu} according to the mass function (9), and performing the weighted update (10). Denoting the maximum degree of the graph by $\rho_{\max}$, we require at most $(\rho_{\max}-1)d$ multiplications to compute M_{vu}. Moreover, an additional $2d$ operations are needed to compute the probability mass function p_{vu}. On the other hand, generating a random index J_{vu}, can be done with less than d operations by picking a number U uniformly at random from $[0,1]$ and setting[7] $J_{vu} := \inf\big\{j : \sum_{\ell=1}^{j} p_{vu}(\ell) > U\big\}$. Finally the update (10) needs $3d+3$ operations. Adding up these contributions, we find that the SBP algorithm requires at most $(\rho_{\max}+5)d+3$ multiplications and/or summations per iteration per edge to update the messages. As can be seen from (4), the regular BP complexity is $\Theta\big(d^2\big)$. Therefore, for graphs with bounded degree (of most interest in practical applications), the SBP message updates have reduced the per iteration computational complexity by a factor of d. In addition to computational efficiency, SBP provides us with a significant gain in message/communication complexity over BP. This can be observed from the fact that the normalized compatibility matrix Γ_{vu} is only a function of edge potentials ψ_{vu}, hence known to the node v. Therefore, node u has to transmit only the random column index J_{vu} to node v, which can be done with $\log_2 d$ bits. Thus, we obtain a significant reduction in communication complexity relative to standard BP, which requires transmitting a $(d-1)$-dimensional vector of real numbers per edge at every round. Here we summarize the features of our algorithm that make it appealing for practical purposes.

- *Computational complexity*: SBP reduces the per iteration complexity by an order of magnitude from $\Theta(d^2)$ to $\Theta(d)$.
- *Communication complexity*: SBP requires transmitting only $\log_2 d$ bits per edge in contrast to transmitting a $(d-1)$-dimensional vector of real numbers in the case of BP.

The remainder of the paper is devoted to understanding when, and if so, how quickly the SBP message updates converge to a BP fixed point. Let us provide some intuition as to why such a behavior might be expected. Recall that the update (10) is random, depending on the choice of index J chosen in step II(b). Suppose that we take expectations of the update (10) only over the distribution (9), in effect conditioning on all past randomness in the algorithm. (We make this idea precise via the notion of σ-fields in our analysis.) Doing so yields that the expectation of the update (10) is given by

$$\mathbb{E}\big[m_{vu}^{t+1} \mid m_{vu}^{t}\big] \;=\; (1-\lambda^t)\, m_{vu}^{t} \;+\; \lambda^t \sum_{j=1}^{d} \Gamma_{vu}(:,j)\, p_{vu}^{t}(j).$$

Recalling the definitions (7) and (9) of the matrix Γ_{vu} and mass function p_{vu}, respectively, and performing some

[6]The columns of the compatibility matrix Γ_{vu} are normalized to sum to one: i.e., $\sum_{i=1}^{d} \Gamma_{vu}(i,j) = 1$ for all $j = 1,2,\ldots,d$.

[7]It is known that for any distribution function $G(\cdot)$, the random variable $G^{-1}(U)$ has the distribution $G(\cdot)$.

Stochastic Belief Propagation Algorithm:

(I) Initialize the message vector $m^0 \in \mathbb{R}_+^D$.

(II) For iterations $t = 0, 1, 2, 3, \ldots$, and for each directed edge $(v \leftarrow u) \in \vec{\mathcal{E}}$:

(a) Compute the product of incoming messages:

$$M_{vu}^t(j) \quad = \prod_{w \in \mathcal{N}(u) \setminus \{v\}} m_{uw}^t(j) \quad \text{for } j \in \{1, \ldots, d\}. \tag{8}$$

(b) Pick a random index $J_{vu}^{t+1} \in \{1, 2, \ldots, d\}$ according to the probability distribution

$$p_{vu}^t(j) \;\propto\; M_{vu}^t(j)\, \beta_{vu}(j) \quad \text{for } j \in \{1, \ldots, d\}. \tag{9}$$

(c) For a given step size $\lambda^t \in (0, 1)$, update the message $m_{vu}^{t+1} \in \mathbb{R}_+^d$ via

$$m_{vu}^{t+1} \;=\; (1 - \lambda^t)\, m_{vu}^t \;+\; \lambda^t\, \Gamma_{vu}(:, J_{vu}^{t+1}). \tag{10}$$

Fig. 3: Specification of stochastic belief propagation.

algebra, we see that, in an average sense, the SBP message update is equivalent to (a damped version of the) usual BP message update. The technical difficulties lie in showing that despite the fluctuations around this average behavior, the SBP updates still converge to the BP fixed point when the stepsize or damping parameter λ^t is suitably chosen. We now turn to precisely this task.

B. Main Theoretical Results

Thus far, we have proposed a stochastic variant of the usual belief propagation (BP) algorithm. In contrast to the usual deterministic updates, this algorithm generates a random sequence $\{m^t\}_{t=0}^{\infty}$ of message vectors. This randomness raises two natural questions:

- Is the SBP algorithm *strongly consistent*? More precisely, assuming that the ordinary BP algorithm has a unique fixed point m^*, under what conditions do we have $m^t \to m^*$ almost surely as $t \to \infty$?
- When convergence occurs, *how fast* does it take place? The computational complexity per iteration is significantly reduced, but what are the trade-offs incurred by the number of iterations required?

The goal of this section is to provide some precise answers to these questions, ones which show that under certain conditions, there are provable gains to be achieved by the SBP algorithm. We begin with the case of trees, for which the ordinary BP message updates are known to have a unique fixed point for any choice of potential functions. For any tree-structured problem, the upcoming Theorem 1 guarantees that the SBP message updates are strongly consistent, and moreover that in terms of the elementwise ℓ_∞ norm they converge in expectation at least as quickly as $\mathcal{O}(1/\sqrt{t})$, where t is the number of iterations. We then turn to the case of general graphs. Although the BP fixed point need not be unique in general, a number of contractivity conditions that guarantee uniqueness and convergence of ordinary BP have been developed (e.g., [29], [12], [20], [23]). Working under such conditions, we show in Theorem 2 that the SBP algorithm is strongly consistent, and we show that the normalized mean-squared error decays at least as quickly as $\mathcal{O}(1/t)$. In addition, we provide high probability bounds on the error at each iteration, showing that the typical performance is highly concentrated around its average. Finally, in Section III-B3, we provide a new set of sufficient conditions for contractivity in terms of node/edge potentials and the graph structure. As we discuss, our theoretical analysis shows not only that SBP is provably correct, but also that in various regimes, substantial gains in overall computational complexity can be obtained relative to the ordinary BP.

1) Guarantees for Tree-Structured Graphs: We begin with the case of a tree-structured graph, meaning a graph $\mathcal{G}$ that contains no cycles. As a special case, the hidden Markov chain shown in Figure 1(b) is an instance of such a tree-structured graph. Recall that for some integer $r \geq 1$, a square matrix A is said to be nilpotent of degree r if $A^r = 0$. (We refer the reader to Horn and Johnson [11] for further background on nilpotent matrices and their properties.) Also recall the definition of the diameter of a graph $\mathcal{G}$, denoted by $\mathrm{diam}(\mathcal{G})$, as the length (number of edges) of the longest path between any pair of nodes in the graph. For a tree, this diameter can be at most $n-1$, a bound achieved by the chain graph. In stating Theorem 1, we make use of the following definition: for vectors $x, y \in \mathbb{R}^D$, we write $x \preceq y$ if and only if $x(i) \leq y(i)$ for all $i = 1, 2, \ldots, D$. Moreover, for an arbitrary $x \in \mathbb{R}^D$, let $|x|$ denote the vector obtained from taking the absolute value of its elements. With this notation in hand, we are now ready to state our first result.

Theorem 1 (Tree-structured graphs)**.** *For any tree-structured Markov random field, the sequence of messages $\{m^t\}_{t=0}^{\infty}$ generated by the SBP algorithm with step size $\lambda^t = 1/(t+1)$,*

has the following properties:

(a) The message sequence $\{m^t\}_{t=0}^{\infty}$ converges almost surely to the unique BP fixed point m^* as $t \to \infty$.

(b) There exist a nilpotent matrix $A \in \mathbb{R}^{D\times D}$ of degree at most $r = \mathrm{diam}(\mathcal{G})$ such that the D-dimensional error vector $m^t - m^*$ satisfies the elementwise inequality

$$\mathbb{E}\big[|m^t - m^*|\big] \preceq 4\,(I-2A)^{-1}\,\frac{\vec{1}}{\sqrt{t}}, \qquad (11)$$

for all iterations $t = 1, 2, \ldots$.

Remarks: The proof of this result is given in Section IV-A. Part (a) shows that the SBP algorithm is guaranteed to converge almost surely to the unique BP fixed point, regardless of the choice of node/edge potentials and the initial message vector. Part (b) refines this claim by providing a quantitative upper bound on the rate of convergence: in expectation, the ℓ_∞ norm of the error vector is guaranteed to decay at the rate $\mathcal{O}(1/\sqrt{t})$. As noted by a helpful reviewer, the upper bound in part (b) is likely to be conservative at times, since the inverse matrix $(I-2A)^{-1}$ may have elements that grow exponentially in the graph diameter r. As shown by our experimental results, the theory is overly conservative in this way, as SBP still behaves well on trees with large diameters (such as chains). Indeed, in the following section, we provide less conservative results for general graphs under a certain contractivity condition.

2) Guarantees for General Graphs: Our next theorem addresses the case of general graphs. In contrast to the case of tree-structured graphs, depending on the choice of potential functions, the BP message updates may have multiple fixed points, and need not converge in general. A sufficient condition for both uniqueness and convergence of the ordinary BP message updates, which we assume in our analysis of SBP, is that the update function F, defined in (5), is *contractive*, meaning that there exists some $0 < \mu < 2$ such that

$$\|F(m) - F(m')\|_2 \leq \big(1 - \frac{\mu}{2}\big)\,\|m - m'\|_2. \qquad (12)$$

Past work has established contractivity conditions of this form when the BP updates are formulated in terms of log messages [29], [12], [20], [23]. In Section III-B3, we use related techniques to establish sufficient conditions for contractivity for the BP message update F that involves the messages (as opposed to log messages).

Recalling the normalized compatibility matrix with columns $\Gamma_{vu}(:,j) := \psi_{vu}(:,j)\psi_u(j)/\beta_{vu}(j)$, we define its minimum and maximum values per row as follows:[8]

$$\begin{aligned}\underline{B}^0_{vu}(i) &:= \min_{j\in\mathcal{X}} \Gamma_{vu}(i,j) > 0, \quad \text{and}\\ \overline{B}^0_{vu}(i) &:= \max_{j\in\mathcal{X}} \Gamma_{vu}(i,j) < 1. \end{aligned} \qquad (13)$$

[8]As will be discussed later, we can obtain a sequence of more refined (tighter) lower $\{\underline{B}^\ell_{vu}(i)\}_{\ell=0}^{\infty}$ and upper $\{\overline{B}^\ell_{vu}(i)\}_{\ell=0}^{\infty}$ bounds by confining the space of feasible messages.

The pre-factor in our bounds involves the constant

$$K(\psi) := 4\,\frac{\sum_{(v\leftarrow u)\in\vec{\mathcal{E}}}\big(\max_{i\in\mathcal{X}}\overline{B}^0_{vu}(i)\big)}{\sum_{(v\leftarrow u)\in\vec{\mathcal{E}}}\big(\min_{i\in\mathcal{X}}\underline{B}^0_{vu}(i)\big)}. \qquad (14)$$

With this notation, we have the following result:

Theorem 2 (General graphs)**.** *Suppose that the BP update function $F : \mathbb{R}^D \to \mathbb{R}^D$ satisfies the contraction condition* (12)*.*

(a) Then BP has a unique fixed point m^*, and the SBP message sequence $\{m^t\}_{t=0}^{\infty}$, generated with the step size $\lambda^t = \mathcal{O}(1/t)$, converges almost surely to m^* as $t \to \infty$.

(b) With the step size $\lambda^t = \alpha/(\mu\cdot(t+2))$ for some fixed $1 < \alpha < 2$, we have

$$\begin{aligned}\frac{\mathbb{E}\big[\|m^t - m^*\|_2^2\big]}{\|m^*\|_2^2} &\leq \frac{3^\alpha\,K(\psi)\,\alpha^2}{2^\alpha\,\mu^2(\alpha-1)}\left(\frac{1}{t}\right)\\ &\quad + \frac{\|m^0 - m^*\|_2^2}{\|m^*\|_2^2}\left(\frac{2}{t}\right)^{\alpha} \end{aligned} \qquad (15)$$

for all iterations $t = 1, 2, \ldots$.

(c) With the step size $\lambda^t = 1/(\mu\cdot(t+1))$, we have

$$\frac{\mathbb{E}\big[\|m^t - m^*\|_2^2\big]}{\|m^*\|_2^2} \leq \frac{K(\psi)}{\mu^2}\left(\frac{1+\log t}{t}\right); \qquad (16)$$

also for every $0 < \epsilon < 1$ and $t \geq 2$, we have

$$\frac{\|m^t - m^*\|_2^2}{\|m^*\|_2^2} \leq \frac{K(\psi)}{\mu^2}\left(1 + \frac{8}{\sqrt{\epsilon}}\right)\left(\frac{1+\log t}{t}\right) \qquad (17)$$

with probability at least $1-\epsilon$.

Remarks: The proof of Theorem 2 is given in Section IV-B. Here we discuss some of the various guarantees that it provides. First, part (a) of the theorem shows that the SBP algorithm is strongly consistent, in that it converges almost surely to the unique BP fixed point. This claim is analogous to the almost sure convergence established in Theorem 1(a) for trees. Second, the bound (15) in Theorem 2(b) provides a non-asymptotic bound on the normalized mean-squared error $\mathbb{E}[\|m^t - m^*\|_2^2]/\|m^*\|_2^2$. For the specified choice of step-size ($1 < \alpha < 2$), the first component of the bound (15) is dominant, hence the expected error (in squared ℓ_2-norm) is of the order[9] $1/t$. Therefore, after $t = \Theta(1/\delta)$ iterations, the SBP algorithm returns a solution with MSE at most $\mathcal{O}(\delta)$. Finally, part (c) provides bounds, both in expectation and with high probability, for a slightly different step size choice. On one hand, the bound in expectation (16) is of the order $\mathcal{O}((\log t)/t)$, and so includes an additional logarithmic factor not present in the bounds from part (b). However, as shown in the high probability bound (17), the

[9]At least superficially, this rate might appear faster than the $1/\sqrt{t}$ rate established for trees in Theorem 1(b); however, the reader should be careful to note that Theorem 1 involves the elementwise ℓ_∞-norm, which is not squared, as opposed to the squared ℓ_2-norm studied in Theorem 2.

squared error is also guaranteed to satisfy a sample-wise version of the same bound with high probability. This theoretical claim is consistent with our later experimental results, showing that the error exhibits tight concentration around its expected behavior.

Let us now compare the guarantees of SBP to those of BP. Under the contraction condition of Theorem 2, the ordinary BP message updates are guaranteed to converge geometrically quickly, meaning that $\Theta(\log(1/\delta))$ iterations are sufficient to obtain δ-accurate solution. In contrast, under the same conditions, the SBP algorithm requires $\Theta(1/\delta)$ iterations to return a solution with MSE at most δ, so that its iteration complexity is larger. However, as noted earlier, the BP message updates require $\Theta(d^2)$ operations for each edge and iteration, whereas the SBP message updates require only $\Theta(d)$ operations. Putting the pieces together, we conclude that:

- on one hand, ordinary BP requires $\Theta\big(|\mathcal{E}|\, d^2\, \log(1/\delta)\big)$ operations to compute the fixed point to accuracy δ;
- in comparison, SBP requires $\Theta\big(|\mathcal{E}|\, d\, (1/\delta)\big)$ operations to compute the fixed point to expected accuracy δ.

Consequently, we see that as long the desired tolerance is not too small—in particular, if $\delta \geq 1/d$—then SBP leads to computational savings. In many practical applications, the state dimension is on the order of 10^3 to 10^5, so that the precision δ can be of the order 10^{-3} to 10^{-5} before the complexity of SBP becomes of comparable order to that of BP. Given that most graphical models represent approximations to reality, it is likely that larger tolerances δ are often of interest.

3) Sufficient Conditions for Contractivity: Theorem 2 is based on the assumption that the update function is contractive, meaning that its Lipschitz constant L is less than one. In past work, various authors have developed contractivity conditions, based on analyzing the log messages, that guarantee uniqueness and convergence of ordinary BP (e.g., [29], [12], [20], [23]). Our theorem requires contractivity on the messages (as opposed to log messages), which requires a related but slightly different argument. In this section, we show how to control L and thereby provide sufficient conditions for Theorem 2 to be applicable.

Our contractivity result applies when the messages under consideration belong to a set of the form

$$\mathcal{S} := \Big\{ m \in \mathbb{R}^D \Big| \sum_{i \in \mathcal{X}} m_{vu}(i) = 1, \underline{B}_{vu}(i) \leq m_{vu}(i) \leq \overline{B}_{vu}(i) \quad \forall (v \leftarrow u) \in \vec{\mathcal{E}},\ \forall i \in \mathcal{X} \Big\}, \tag{18}$$

for some choice of the upper and lower bounds—namely, $\overline{B}_{vu}(i)$ and $\underline{B}_{vu}(i)$ respectively. For instance, for all iterations $t = 0, 1, \ldots$, the messages always belong to a set of this form[10] with $\underline{B}_{vu}(i) = \underline{B}^0_{vu}(i)$ and $\overline{B}_{vu}(i) = \overline{B}^0_{vu}(i)$, as previously defined (13). Since the bounds $(\underline{B}^0_{vu}(i), \overline{B}^0_{vu}(i))$ do not involve the node potentials, one suspects that they might be tightened at subsequent iterations, and indeed, there is a progressive refinement of upper and lower bounds of this form. Assuming that the messages belong to a set $\mathcal{S}$ at an initial iteration, then for any subsequent iterations, we are guaranteed the inclusion

$$m \in F(\mathcal{S}) := \big\{ F(m') \in \mathbb{R}^D \ \mid \ m' \in \mathcal{S} \big\}, \tag{19}$$

which then leads to the refined upper and lower bounds

$$\underline{B}^1_{vu}(i) := \inf_{m \in \mathcal{S}} \left\{ \sum_{j=1}^d \Gamma_{vu}(i,j)\, \frac{\beta_{vu}(j)\, M_{vu}(j)}{\sum_{\ell=1}^d \beta_{vu}(\ell)\, M_{vu}(\ell)} \right\},$$

and

$$\overline{B}^1_{vu}(i) := \sup_{m \in \mathcal{S}} \left\{ \sum_{j=1}^d \Gamma_{vu}(i,j)\, \frac{\beta_{vu}(j)\, M_{vu}(j)}{\sum_{\ell=1}^d \beta_{vu}(\ell)\, M_{vu}(\ell)} \right\},$$

where we recall the quantity $M_{vu}(j) = \prod_{w \in \mathcal{N}(u)\backslash\{v\}} m_{uw}(j)$ previously defined (8). While such refinements are possible, in order to streamline our presentation, we focus primarily on the zero'th order bounds $\underline{B}_{vu}(i) = \underline{B}^0_{vu}(i)$ and $\overline{B}_{vu}(i) = \overline{B}^0_{vu}(i)$.

Given a set $\mathcal{S}$ of the form (18), we associate with the directed edges $(v \leftarrow u)$ and $(u \leftarrow w)$ (where $w \in \mathcal{N}(u)\backslash\{v\}$) the non-negative numbers

$$\Phi_1(v,u) := \sum_{w \in \mathcal{N}(u)\backslash\{v\}} \big(\phi_{vu,uw}\, (\phi_{vu,uw} + \chi_{vu,uw})\big)^{\frac{1}{2}}, \tag{20}$$

and

$$\Phi_2(u,w) := \sum_{v \in \mathcal{N}(u)\backslash\{w\}} \big(\phi_{vu,uw}\, (\phi_{vu,uw} + \chi_{vu,uw})\big)^{\frac{1}{2}}, \tag{21}$$

where

$$\phi_{vu,uw} := \max_{j \in \mathcal{X}} \sup_{m \in \mathcal{S}} \left\{ \frac{\beta_{vu}(j)\, M_{vu}(j)}{\sum_{k=1}^d \beta_{vu}(k)\, M_{vu}(k)}\, \frac{1}{m_{uw}(j)} \right\}, \tag{22}$$

and

$$\chi_{vu,uw} := \max_{j \in \mathcal{X}} \sup_{m \in \mathcal{S}} \left\{ \frac{\beta_{vu}(i)\, M_{vu}(i)}{\big(\sum_{k=1}^d \beta_{vu}(k)\, M_{vu}(k)\big)^2} \sum_{j=1}^d \frac{\beta_{vu}(j)\, M_{vu}(j)}{m_{uw(j)}} \right\}. \tag{23}$$

Recall the normalized compatibility matrix $\Gamma_{vu} \in \mathbb{R}^{d \times d}$ on the directed edge $(v \leftarrow u)$, as previously defined

[10] It turns out that the BP update function on the directed edge $(v \leftarrow u)$ is a convex combination of the normalized columns $\Gamma_{vu}(:,j)$ for $j = 1, \ldots, d$. Therefore, we have $\underline{B}^0_{vu}(i) \leq m_{vu}(i) \leq \overline{B}^0_{vu}(i)$, for all $i = 1, \ldots, d$.

in (7). Since Γ_{vu} has positive entries, the Perron-Frobenius theorem [11] guarantees that the maximal eigenvalue is equal to one, and is associated with a pair of left and right eigenvectors (unique up to scaling) with positive entries. Since Γ_{vu} is column-stochastic, any multiple of the all-one vector $\vec{1}$ can be chosen as the left eigenvector. Letting $z_{vu} \in \mathbb{R}^d$ denote the right eigenvector with positive entries, we are guaranteed that $\vec{1}^T z_{vu} > 0$, and hence we may define the matrix $\Gamma_{vu} - z_{vu}\vec{1}^T/(\vec{1}^T z_{vu})$. By construction, this matrix has all of its eigenvalues strictly less than 1 in absolute value (Lemma 8.2.7, [11]).

Proposition 1. *The global update function $F : \mathbb{R}^D \to \mathbb{R}^D$ defined in* (5) *is Lipschitz with constant at most*

$$L := 2 \max_{(v \leftarrow u) \in \vec{\mathcal{E}}} \|\Gamma_{vu} - \frac{z_{vu}\vec{1}^T}{\vec{1}^T z_{vu}}\|_2 \max_{(v \leftarrow u) \in \vec{\mathcal{E}}} \Phi_1(v,u) \max_{(u \leftarrow w) \in \vec{\mathcal{E}}} \Phi_2(u,w), \tag{24}$$

where $\|\cdot\|_2$ denotes the maximum singular value of a matrix.

In order to provide some intuition for Proposition 1, let us consider a simple but illuminating example.

Example 1 (Potts model)**.** The *Potts model* [9], [28], [16] is often used for denoising, segmentation, and stereo computation in image processing and computer vision. It is a pairwise Markov random field that is based on edge potentials of the form

$$\psi_{vu}(i,j) = \begin{cases} 1 & \text{if } i = j, \text{ and} \\ \gamma & \text{if } i \neq j, \end{cases}$$

for all edges $(v,u) \in \mathcal{E}$ and $i,j \in \{1,2,\ldots,d\}$. The parameter $\gamma \in (0,1]$ can be tuned to enforce different degrees of smoothness: at one extreme, setting $\gamma = 1$ enforces no smoothness, whereas a choice close to zero enforces a very strong type of smoothness. (To be clear, the special structure of the Potts model can be exploited to compute the BP message updates quickly; our motivation in considering it here is only to provide a simple illustration of our contractivity condition.)

For the Potts model, we have $\beta_{vu}(j) = \psi_u(j)\,(1 + (d-1)\gamma)$, and hence Γ_{vu} is a symmetric matrix with

$$\Gamma_{vu}(i,j) = \begin{cases} \frac{1}{1+(d-1)\gamma} & \text{if } i = j \\ \frac{\gamma}{1+(d-1)\gamma} & \text{if } i \neq j. \end{cases}$$

Some straightforward algebra shows that the second largest singular value of Γ_{vu} is given by $(1-\gamma)/(1+(d-1)\gamma)$, whence

$$\max_{(v \leftarrow u) \in \vec{\mathcal{E}}} \|\Gamma_{vu} - \frac{z_{vu}\vec{1}^T}{\vec{1}^T z_{vu}}\|_2 = \frac{1-\gamma}{1+(d-1)\gamma}.$$

The next step is to find upper bounds on the terms $\Phi_1(v,u)$ and $\Phi_2(u,w)$, in particular by upper bounding the quantities $\phi_{vu,uw}$ and $\chi_{vu,uw}$, as defined in equations (22) and (23) respectively. In Appendix A, we show that the Lipschitz constant of F_{vu} is upper bounded as

$$L \leq 4\,(1-\gamma)(1+(d-1)\gamma) \max_{u \in \mathcal{V}} \left\{ \frac{(\rho_u - 1)^2}{\gamma^{2\rho_u}} \max_{j \in \mathcal{X}} \left\{ \frac{\psi_u(j)}{\sum_{\ell=1}^d \psi_u(\ell)} \right\}^2 \right\},$$

where ρ_u is the degree of node u. Therefore, a sufficient condition for contractivity in the case of the Potts model is

$$\max_{u \in \mathcal{V}} \left\{ \frac{(\rho_u - 1)}{\gamma^{\rho_u}} \max_{j \in \mathcal{X}} \left\{ \frac{\psi_u(j)}{\sum_{\ell=1}^d \psi_u(\ell)} \right\} \right\} < \left(\frac{1}{4\,(1-\gamma)\,(1+(d-1)\gamma)} \right)^{\frac{1}{2}}. \tag{25}$$

To gain intuition, consider the special case in which the node potentials are uniform, so that $\psi_u(j)/(\sum_{\ell=1}^d \psi_u(\ell)) = 1/d$ for $j = 1,2,\ldots,d$. In this case, for any graph with bounded node degrees, the bound (25) guarantees contraction for all γ in an interval $[\epsilon, 1]$. For non-uniform node potentials, the inequality (25) is weaker, but it can be improved via the refined sets (19) discussed previously.

IV. Proofs

We now turn to the proofs of our two main results, namely Theorems 1 and 2, as well as the auxiliary result, Proposition 1, on contractivity of the BP message updates. For our purposes, it is convenient to note that the ordinary BP update can be written as an expectation of the form

$$F_{vu}(m^t) = \mathbb{E}_{J_{vu}^{t+1} \sim p_{vu}^t}\big[\Gamma_{vu}(:, J_{vu}^{t+1})\big], \tag{26}$$

for all $t = 0,1,\ldots$. Here the index J_{vu}^{t+1} is chosen randomly according to the probability mass function (9).

A. Proof of Theorem 1

We begin by stating a lemma that plays a central role in the proof of Theorem 1.

Lemma 1. *For any tree-structured Markov random field, there exists a nilpotent matrix $A \in \mathbb{R}^{D \times D}$ of degree at most $r = \operatorname{diam}(\mathcal{G})$ such that*

$$|F(m) - F(m')| \preceq A\,|m - m'|, \tag{27}$$

for all $m, m' \in \mathcal{S}$.

The proof of this lemma is somewhat technical, so that we defer it to Appendix B. In interpreting this result, the reader should recall that for vectors $x, y \in \mathbb{R}^D$, the notation $x \preceq y$ denotes inequality in an elementwise sense—i.e., $x(i) \leq y(i)$ for $i = 1,\ldots,D$.

An immediate corollary of this lemma is the existence and uniqueness of the BP fixed point. Since we may iterate inequality (27), we find that

$$|F^{(\ell)}(m) - F^{(\ell)}(m')| \preceq A^{\ell} |m - m'|,$$

for all iterations $\ell = 1, 2, \ldots$, and arbitrary messages m, m', where $F^{(\ell)}$ denotes the composition of F with itself ℓ times. The nilpotence of A ensures that $A^r = 0$, and hence $F^{(r)}(m) = F^{(r)}(m')$ for all messages m, and m'. Let $m^* = F^{(r)}(m)$ denote the common value. The claim is that m^* is the unique fixed point of the BP update function F. This can be shown as follows: from Lemma 1 we have

$$\begin{aligned} |F(m^*) - m^*| &= |F^{(r+1)}(m) - F^{(r)}(m)| \\ &\preceq A\,|F^{(r)}(m) - F^{(r-1)}(m)|. \end{aligned}$$

Iterating the last inequality for the total of r times, we obtain

$$|F(m^*) - m^*| \preceq A^r\,|F(m) - m| = 0,$$

and hence $F(m^*) = m^*$. On the other hand, the uniqueness of the BP fixed point is a direct consequence of the facts that for any fixed point m^* we have $F^{(r)}(m^*) = m^*$, and for all arbitrary messages m, m' we have $F^{(r)}(m) = F^{(r)}(m')$. Accordingly, we see that Lemma 1 provides an alternative proof of the well-known fact that BP converges to a unique fixed point on trees after at most $r = \mathrm{diam}(\mathcal{G})$ iterations.

We now show how Lemma 1 can be used to establish the two claims of Theorem 1.

1) Part (a): Almost Sure Consistency: We begin with the almost sure consistency claim of part (a). By combining all the local updates, we form the global update rule

$$m^{t+1} = (1 - \lambda^t)\, m^t + \lambda^t\, \nu^{t+1} \tag{28}$$

for iterations $t = 0, 1, 2, \ldots$, where

$$\nu^{t+1} := \{\Gamma_{vu}(:, J_{vu}^{t+1})\}_{(v \leftarrow u) \in \vec{\mathcal{E}}}$$

is the D-dimensional vector obtained from stacking up all the normalized columns $\Gamma_{vu}(:, J_{vu}^{t+1})$. Defining the vector $Y^{t+1} := \nu^{t+1} - F(m^t) \in \mathbb{R}^D$, we can rewrite the update equation (28) as

$$m^{t+1} = (1 - \lambda^t)\, m^t + \lambda^t\, F(m^t) + \lambda^t\, Y^{t+1} \tag{29}$$

for $t = 0, 1, 2, \ldots$. With our step size choice $\lambda^t = 1/(t+1)$, unwrapping the recursion (29) yields the representation

$$m^t = \frac{1}{t} \sum_{\ell=0}^{t-1} F(m^{\ell}) + \frac{1}{t} \sum_{\ell=1}^{t} Y^{\ell}.$$

Subtracting the unique fixed point m^* from both sides then leads to

$$\begin{aligned} m^t - m^* &= \frac{1}{t} \sum_{\ell=1}^{t-1} (F(m^{\ell}) - F(m^*)) \\ &\quad + \underbrace{\frac{1}{t} \sum_{\ell=1}^{t} Y^{\ell} + \frac{1}{t}\,(F(m^0) - F(m^*))}_{Z^t}, \end{aligned} \tag{30}$$

where we have introduced the convenient shorthand Z^t. We may apply the triangle inequality to each element of this vector equation; doing so and using Lemma 1 to upper bound the terms $|F(m^{\ell}) - F(m^*)|$, we obtain the element-wise inequality

$$|m^t - m^*| \preceq \frac{1}{t} \sum_{\ell=1}^{t-1} A\,|m^{\ell} - m^*| + |Z^t| \quad \text{for } t = 1, 2, \ldots.$$

Since A^r is the all-zero matrix, unwrapping the last inequality $r = \mathrm{diam}(\mathcal{G})$ times yields the element-wise upper bound

$$|m^t - m^*| \preceq G_0^t + AG_1^t + A^2 G_2^t + \cdots + A^{r-1} G_{r-1}^t, \tag{31}$$

where the terms G_{ℓ}^t are defined via the recursion $G_{\ell}^t := \frac{1}{t} \sum_{j=1}^{t-1} G_{\ell-1}^j$ for $\ell = 1, \ldots, r-1$, with initial conditions $G_0^t := |Z^t|$.

It remains to control the sequences $\{G_{\ell}^t\}_{t=1}^{\infty}$ for $\ell = 0, 1, \ldots, r-1$. In order to do so, we first establish a martingale difference property for the variables Y^t defined prior to (29). For each $t = 0, 1, 2, \ldots$, define the σ-field $\mathcal{F}^t := \sigma(m^0, m^1, \ldots, m^t)$, as generated by the randomness in the messages up to time t. Based on the representation (26), we see that $\mathbb{E}\big[Y^{t+1} | \mathcal{F}^t\big] = \vec{0}$, showing that $\{Y^{t+1}\}_{t=0}^{\infty}$ forms martingale difference sequence with respect to the filtration $\{\mathcal{F}^t\}_{t=0}^{\infty}$. From the definition, it can be seen that the entries of Y^{t+1} are bounded; more precisely, we have $|Y^{t+1}(i)| \leq 1$ for all iterations $t = 0, 1, 2, \ldots$, and all states $i = 1, 2, \ldots D$. Consequently, the sequence $\{Y^{\ell}\}_{\ell=1}^{\infty}$ is a bounded martingale difference sequence.

We begin with the term G_0^t. Since Y^{ℓ} is a bounded martingale difference, standard convergence results [8] guarantee that $|\sum_{\ell=1}^{t} Y^{\ell}|/t \to \vec{0}$ almost surely. Moreover, we have the bound $|F(m^0) - F(m^*)|/t \preceq \vec{1}/t$. Recalling the definition of Z^t from (30), we conclude that $G_0^t = |Z^t|$ converges to the all-zero vector almost surely as $t \to \infty$. In order to extend our argument to the terms G_{ℓ}^t for $\ell = 1, \ldots, r-1$, we make use of the following fact: for any sequence of real numbers $\{x^t\}_{t=0}^{\infty}$ such that $x^t \to 0$, we also have $(\sum_{\ell=0}^{t-1} x^{\ell})/t \to 0$ (e.g., see Royden [24]). Consequently, for any realization ω such that the deterministic sequence $\{G_0^t(\omega)\}_{t=0}^{\infty}$ converges to zero,

we are also guaranteed that the sequence $\{G_1^t(\omega)\}_{t=0}^{\infty}$, with elements $G_1^t(\omega) = (\sum_{j=1}^{t-1} G_0^j(\omega))/t$, converges to zero. Since we have shown that $G_0^t \stackrel{\text{a.s.}}{\to} 0$, we conclude that $G_1^t \stackrel{\text{a.s.}}{\to} 0$ as well. This argument can be iterated, thereby establishing almost sure convergence for all of the terms G_ℓ^t. Putting the pieces together, we conclude that the vector $|m^t - m^*|$ converges almost surely to the all-zero vector as $t \to \infty$, thereby completing the proof of part (a).

2) Part (b): Bounds on Expected Absolute Error: We now turn to part (b) of Theorem 1, which provides upper bounds on the expected absolute error. We establish this claim by exploiting some martingale concentration inequalities [5]. From part (a), we know that $\{Y^t\}_{t=1}^{\infty}$ is a bounded martingale difference sequence, in particular with $|Y^t(i)| \leq 1$. Applying the Azuma-Hoeffding inequality [5] yields the tail bound

$$\mathbb{P}\Big(\frac{1}{t}\,|\sum_{\ell=1}^{t} Y^\ell(i)| \,>\, \gamma\Big) \;\leq\; 2\exp\Big(-\frac{t\,\gamma^2}{2}\Big),$$

for all $\gamma > 0$, and $i = 1, 2, \ldots, D$. By integrating this tail bound, we can upper bound the mean: in particular, we have

$$\begin{aligned}\mathbb{E}\Big[\frac{1}{t}\,|\sum_{\ell=1}^{t} Y^\ell(i)|\Big] \;&=\; \int_0^\infty \mathbb{P}\Big(\frac{1}{t}\,|\sum_{\ell=1}^{t} Y^\ell(i)| \,>\, \gamma\Big)\,d\gamma \\ &\leq\; \sqrt{\frac{2\pi}{t}},\end{aligned}$$

and hence

$$\mathbb{E}\big[G_0^t\big] \;=\; \mathbb{E}\big[|Z^t|\big] \;\preceq\; \sqrt{\frac{2\pi}{t}}\,\vec{1} + \frac{\vec{1}}{t} \;\preceq\; \frac{4}{\sqrt{t}}\,\vec{1}. \qquad (32)$$

Turning to the term G_1^t, we have

$$\mathbb{E}[G_1^t] \;= \frac{1}{t}\sum_{\ell=1}^{t-1}\mathbb{E}[G_0^\ell] \;\stackrel{\text{(i)}}{\preceq}\; \frac{1}{t}\sum_{\ell=1}^{t-1}\frac{4}{\sqrt{\ell}}\,\vec{1} \;\stackrel{\text{(ii)}}{\preceq}\; \frac{2\cdot 4}{\sqrt{t}}\,\vec{1},$$

where step (i) uses the inequality (32), and step (ii) is based on the elementary upper bound $\sum_{\ell=1}^{t-1} 1/\sqrt{\ell} \leq 1 + \int_1^{t-1} 1/\sqrt{x}\,dx < 2\sqrt{t}$. By repeating this same argument in a recursive manner, we conclude that $\mathbb{E}\big[G_\ell^t\big] \preceq (2^\ell \cdot 4/\sqrt{t})\,\vec{1}$ for $\ell = 2, 3, \ldots, r-1$. Taking the expectation on both sides of the the inequality (31) and substituting these upper bounds, we obtain

$$\mathbb{E}\big[|m^t - m^*|\big] \;\preceq\; 4\left(\sum_{\ell=0}^{r-1} 2^\ell A^\ell\right)\frac{\vec{1}}{\sqrt{t}} \;=\; 4\,(I-2A)^{-1}\,\frac{\vec{1}}{\sqrt{t}},$$

where we have used the fact that $A^r = 0$.

B. Proof of Theorem 2

We now turn to the proof of Theorem 2. Note that since the update function is contractive, the existence and uniqueness of the BP fixed point is an immediate consequence of the Banach fixed-point theorem [1].

1) Part (a): Almost Sure Consistency: We establish part (a) by applying the Robbins-Monro theorem, a classical result from stochastic approximation theory (e.g., [22], [4]). In order to do so, we begin by writing the update (10) in the form

$$m_{vu}^{t+1} = m_{vu}^t - \lambda^t \underbrace{\big\{m_{vu}^t - \Gamma_{vu}(:, J_{vu}^{t+1})\big\}}_{H_{vu}(m_{vu}^t, J_{vu}^{t+1})},$$

where for any realization $\bar{J}_{vu} \in \{1, 2, \ldots, d\}$, the mapping $m_{vu} \mapsto H_{vu}(m_{vu}, \bar{J}_{vu})$ should be understood as a function from $\mathbb{R}^d$ to $\mathbb{R}^d$. By concatenating together all of these mappings, one for each directed edge $(v \leftarrow u)$, we obtain a family of mappings $H(\cdot, \bar{J})$ from $\mathbb{R}^D$ to $\mathbb{R}^D$, one for each realization $\bar{J} \in \{1, 2, \ldots, d\}^{2|\vec{\mathcal{E}}|}$ of column indices.

With this notation, we can write the message update of the SBP algorithm in the compact form

$$m^{t+1} \;=\; m^t \;-\; \lambda^t\, H(m^t, J^{t+1}), \qquad (33)$$

valid for for $t = 1, 2, \ldots$, and suitable for application of the Robbins-Monro theorem. (See Appendix C for further details.)

In order to apply this result, we need to verify its hypotheses. First of all, it is easy to see that we have a bound of the form

$$\mathbb{E}\big[\|H(m, J)\|_2^2\big] \;\leq\; c(1 + \|m\|_2^2),$$

for some constant c. Moreover, the conditional distribution of the vector J^{t+1}, given the past, depends only on m^t; more precisely we have

$$\mathbb{P}\big(J^{t+1}|J^t, J^{t-1}, \ldots, m^t, m^{t-1}, \ldots\big) \;=\; \mathbb{P}\big(J^{t+1}|m^t\big).$$

Lastly, defining the averaged function $h(m) := \mathbb{E}\big[H(m, J)|m\big] \;=\; m - F(m)$, the final requirement is to verify that the fixed point m^* satisfies the stability condition

$$\inf_{m \in \mathcal{S}\setminus\{m^*\}} \langle m - m^*,\, h(m)\rangle \;>\; 0, \qquad (34)$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product, and $\mathcal{S}$ denotes the compact set in which the messages lie. Using the Cauchy-Schwartz inequality and the fact that F is Lipschitz with constant $L = 1 - \mu/2$, we obtain

$$\begin{aligned}&\langle m - m^*,\, h(m) - h(m^*)\rangle \\ &\quad= \;\|m - m^*\|_2^2 \;-\; \langle m - m^*,\, F(m) - F(m^*)\rangle \\ &\quad\geq\; \frac{\mu}{2}\,\|m - m^*\|_2^2 \;>\; 0, \qquad (35)\end{aligned}$$

where the strict inequality holds for all $m \neq m^*$. Since m^* is a fixed point, we must have $h(m^*) = m^* - F(m^*) = 0$, which concludes the proof.

2) Part (b): Non-Asymptotic Bounds on Normalized Mean-Squared Error: Let $e^t := (m^t - m^*)/\|m^*\|_2$ denote the re-normalized error vector. In order to upper bound $\mathbb{E}\big[\|e^t\|_2^2\big]$ for all $t = 1, 2, \ldots$, we first control the quantity $\|e^{t+1}\|_2^2 - \|e^t\|_2^2$, corresponding to the increment in the squared error. Doing some simple algebra yields

$$\begin{aligned}\|e^{t+1}\|_2^2 - \|e^t\|_2^2 &\\ &= \frac{1}{\|m^*\|_2^2}\big(\|m^{t+1} - m^*\|_2^2 - \|m^t - m^*\|_2^2\big)\\ &= \frac{1}{\|m^*\|_2^2}\,\langle m^{t+1} - m^t,\, m^{t+1} + m^t - 2m^*\rangle.\end{aligned}$$

Recalling the update equation (33), we obtain

$$\begin{aligned}\|e^{t+1}\|_2^2 - \|e^t\|_2^2 \;&= \frac{1}{\|m^*\|_2^2}\,\langle -\lambda^t H(m^t, J^{t+1}),\\ &\quad -\lambda^t H(m^t, J^{t+1}) + 2(m^t - m^*)\rangle\\ &= \frac{(\lambda^t)^2}{\|m^*\|_2^2}\,\|H(m^t, J^{t+1})\|_2^2\\ &\quad - \frac{2\lambda^t}{\|m^*\|_2^2}\,\langle H(m^t, J^{t+1}),\, m^t - m^*\rangle.\end{aligned} \tag{36}$$

Now taking the expectation on both sides of (36) yields

$$\begin{aligned}\mathbb{E}[\|e^{t+1}\|_2^2] - \mathbb{E}[\|e^t\|_2^2] \;&= \frac{(\lambda^t)^2}{\|m^*\|_2^2}\,\mathbb{E}\big[\|H(m^t, J^{t+1})\|_2^2\big]\\ &\quad - \frac{2\lambda^t}{\|m^*\|_2^2}\,\mathbb{E}\big[\mathbb{E}\big[\langle H(m^t, J^{t+1}),\, m^t - m^*\rangle | \mathcal{F}^t\big]\big]\\ &= \frac{(\lambda^t)^2}{\|m^*\|_2^2}\,\mathbb{E}\big[\|H(m^t, J^{t+1})\|_2^2\big]\\ &\quad - \frac{2\lambda^t}{\|m^*\|_2^2}\,\mathbb{E}\big[\langle h(m^t) - h(m^*),\, m^t - m^*\rangle\big],\end{aligned} \tag{37}$$

where we used the facts that $\mathbb{E}[H(m^t, J^{t+1})|\mathcal{F}^t] = h(m^t)$ and $h(m^*) = 0$. We continue by upper bounding the term $G_1 = \|H(m^t, J^{t+1})\|_2^2/\|m^*\|_2^2$ and lower bounding the term $G_2 = \langle h(m^t) - h(m^*),\, m^t - m^*\rangle/\|m^*\|_2^2$.

Lower bound on G_2: Recalling (35) from our proof of part (a), we see that

$$G_2 \;\geq\; \frac{\mu}{2}\,\|e^t\|_2^2. \tag{38}$$

Upper bound on G_1: From the definition of the update function, we have

$$\begin{aligned}\|H(m^t, J^{t+1})\|_2^2 \;&= \sum_{(v\leftarrow u)\in\vec{\mathcal{E}}} \|m_{vu}^t - \Gamma_{vu}(:, J_{vu}^t)\|_2^2\\ &\leq 2\sum_{(v\leftarrow u)\in\vec{\mathcal{E}}} \big(\|m_{vu}^t\|_2^2 \,+\, \|\Gamma_{vu}(:, J_{vu}^t)\|_2^2\big).\end{aligned}$$

Recalling the bounds (13) and using the fact that vectors m_{vu}^t and $\Gamma_{vu}(:, J_{vu}^t)$ sum to one, we obtain

$$\begin{aligned}&\|H(m^t, J^{t+1})\|_2^2\\ &\leq\; 2\sum_{(v\leftarrow u)\in\vec{\mathcal{E}}} \big(\max_{i\in\mathcal{X}} \overline{B}_{vu}^0(i)\big)\big(\|m_{vu}^t\|_1 \,+\, \|\Gamma_{vu}(:, J_{vu}^t)\|_1\big)\\ &=\; 4\sum_{(v\leftarrow u)\in\vec{\mathcal{E}}} \big(\max_{i\in\mathcal{X}} \overline{B}_{vu}^0(i)\big).\end{aligned}$$

On the other hand, we also have

$$\begin{aligned}\|m^*\|_2^2 \;&\geq \sum_{(v\leftarrow u)\in\vec{\mathcal{E}}} \big(\min_{i\in\mathcal{X}} \underline{B}_{vu}^0(i)\big)\|m_{vu}^*\|_1\\ &= \sum_{(v\leftarrow u)\in\vec{\mathcal{E}}} \big(\min_{i\in\mathcal{X}} \underline{B}_{vu}^0(i)\big).\end{aligned}$$

Combining the pieces, we conclude that the term G_1 is upper bounded as

$$G_1 \;\leq\; K(\psi) \;:=\; 4\,\frac{\sum_{(v\leftarrow u)\in\vec{\mathcal{E}}}\big(\max_{i\in\mathcal{X}} \overline{B}_{vu}^0(i)\big)}{\sum_{(v\leftarrow u)\in\vec{\mathcal{E}}}\big(\min_{i\in\mathcal{X}} \underline{B}_{vu}^0(i)\big)}. \tag{39}$$

Since both G_1 and G_2 are non-negative, the bounds (39) and (38) also hold in expectation. Combining these bounds with the representation (37), we obtain the upper bound $\mathbb{E}[\|e^{t+1}\|_2^2] - \mathbb{E}[\|e^t\|_2^2] \leq\; K(\psi)\,(\lambda^t)^2 - \lambda^t\mu\,\mathbb{E}[\|e^t\|_2^2]$, or equivalently

$$\mathbb{E}[\|e^{t+1}\|_2^2] \;\leq\; K(\psi)\,(\lambda^t)^2 \;+\; (1 - \lambda^t\mu)\,\mathbb{E}[\|e^t\|_2^2].$$

Setting $\lambda^t = \alpha/(\mu(t+2))$ and unwrapping this recursion yields

$$\begin{aligned}\mathbb{E}[\|e^{t+1}\|_2^2] \;&\leq \frac{K(\psi)\,\alpha^2}{\mu^2}\sum_{i=2}^{t+2}\left(\frac{1}{i^2}\prod_{\ell=i+1}^{t+2}\Big(1 - \frac{\alpha}{\ell}\Big)\right)\\ &\quad + \prod_{\ell=2}^{t+2}\Big(1 - \frac{\alpha}{\ell}\Big)\,\mathbb{E}[\|e^0\|_2^2],\end{aligned} \tag{40}$$

where we have adopted the convention that the inside product is equal to one for $i = t+2$. The following lemma, proved in Appendix D, provides a useful upper bound on the products arising in this expression:

Lemma 2. *For all $i \in \{1, 2, \ldots, t+1\}$, we have*

$$\prod_{\ell=i+1}^{t+2}\Big(1 - \frac{\alpha}{\ell}\Big) \;\leq\; \left(\frac{i+1}{t+3}\right)^{\alpha}.$$

Substituting this upper bound into the inequality (40) yields

$$\begin{aligned}&\mathbb{E}[\|e^{t+1}\|_2^2]\\ &\leq\; \frac{K(\psi)\,\alpha^2}{\mu^2(t+3)^\alpha}\sum_{i=2}^{t+2}\frac{(i+1)^\alpha}{i^2} \;+\; \left(\frac{2}{t+3}\right)^{\alpha}\mathbb{E}[\|e^0\|_2^2]\\ &\leq\; \frac{K(\psi)\,\alpha^2}{\mu^2(t+3)^\alpha}\big(\frac{3}{2}\big)^\alpha\sum_{i=2}^{t+2}\frac{1}{i^{2-\alpha}} \;+\; \left(\frac{2}{t+3}\right)^{\alpha}\mathbb{E}[\|e^0\|_2^2].\end{aligned}$$

It remains to upper bound the term $\sum_{i=2}^{t+2} 1/i^{2-\alpha}$. Since the function $1/x^{2-\alpha}$ is decreasing in x for $\alpha < 2$, we have the integral upper bound $\sum_{i=2}^{t+2} 1/i^{2-\alpha} \leq \int_1^{t+2} 1/x^{2-\alpha}\, dx$, which yields

$$\mathbb{E}[\|e^{t+1}\|_2^2] \leq \begin{cases} \left(\frac{3}{2}\right)^{\alpha} \frac{K(\psi)\alpha^2}{\mu^2(1-\alpha)} \frac{1}{(t+3)^{\alpha}} + \left(\frac{2}{t+3}\right)^{\alpha} \mathbb{E}[\|e^0\|_2^2] & 0 < \alpha < 1, \\ \frac{3}{2}\frac{K(\psi)}{\mu^2}\frac{\log(t+2)}{t+3} + \frac{2}{t+3}\mathbb{E}[\|e^0\|_2^2] & \alpha = 1, \\ \left(\frac{3}{2}\right)^{\alpha} \frac{K(\psi)\alpha^2}{\mu^2(\alpha-1)} \frac{(t+2)^{\alpha-1}}{(t+3)^{\alpha}} + \left(\frac{2}{t+3}\right)^{\alpha} \mathbb{E}[\|e^0\|_2^2] & 1 < \alpha < 2. \end{cases}$$

If we now focus on the range of $\alpha \in (1, 2)$, which yields the fastest convergence rate, some simple algebra yields the form of the claim given in the theorem statement.

3) High Probability Bounds: Recall the algebra at the beginning of Section IV-B2. Adding and subtracting the conditional mean of the second term of (36) yields

$$\|e^{t+1}\|_2^2 - \|e^t\|_2^2 = \frac{(\lambda^t)^2}{\|m^*\|_2^2} \|H(m^t, J^{t+1})\|_2^2 - \frac{2\lambda^t}{\|m^*\|_2^2} \langle h(m^t),\, m^t - m^* \rangle + 2\lambda^t \langle Y^{t+1},\, e^t \rangle,$$

where we have denoted the term

$$Y^{t+1} := \frac{h(m^t) - H(m^t, J^{t+1})}{\|m^*\|_2}.$$

Recalling the bounds on $G_1 = \|H(m^t, J^{t+1})\|_2^2 \,/\, \|m^*\|_2^2$ and $G_2 = \langle h(m^t),\, m^t - m^* \rangle \,/\, \|m^*\|_2^2$ from part (b), we have

$$\|e^{t+1}\|_2^2 - \|e^t\|_2^2 \leq K(\psi)(\lambda^t)^2 - \mu\lambda^t \|e^t\|_2^2 + 2\lambda^t \langle Y^{t+1},\, e^t \rangle,$$

or equivalently

$$\|e^{t+1}\|_2^2 \leq K(\psi)(\lambda^t)^2 + (1 - \mu\lambda^t)\|e^t\|_2^2 + 2\lambda^t \langle Y^{t+1},\, e^t \rangle.$$

Substituting the step size choice $\lambda^t = 1/(\mu(t+1))$ and then unwrapping this recursion yields

$$\begin{aligned} \|e^{t+1}\|_2^2 &\leq \frac{K(\psi)}{\mu^2(t+1)} \sum_{\tau=1}^{t+1} \frac{1}{\tau} + \frac{2}{\mu\,(t+1)} \sum_{\tau=0}^{t} \langle Y^{\tau+1},\, e^{\tau} \rangle \\ &\leq \frac{K(\psi)}{\mu^2} \frac{1 + \log(t+1)}{t+1} + \frac{2}{\mu\,(t+1)} \sum_{\tau=0}^{t} \langle Y^{\tau+1},\, e^{\tau} \rangle. \end{aligned} \tag{41}$$

Note that by construction, the sequence $\{Y^{\tau}\}_{\tau=1}^{\infty}$ is a martingale difference sequence with respect to the filtration $\mathcal{F}^{\tau} = \sigma(m^0, m^1, \ldots, m^{\tau})$ that is $\mathbb{E}\big[Y^{\tau+1} \mid \mathcal{F}^{\tau}\big] = \vec{0}$ and accordingly $\mathbb{E}\big[\langle Y^{\tau+1},\, e^{\tau} \rangle\big] = 0$ for $\tau = 0, 1, 2, \ldots$. We continue by controlling the stochastic term $(\sum_{\tau=0}^{t} \langle Y^{\tau+1},\, e^{\tau} \rangle)/(t+1)$—namely its variance,

$$\begin{aligned} &\operatorname{var}\left(\frac{1}{t+1} \sum_{\tau=0}^{t} \langle Y^{\tau+1},\, e^{\tau} \rangle \right) \\ &= \frac{1}{(t+1)^2}\, \mathbb{E}\left[\Big(\sum_{\tau=0}^{t} \langle Y^{\tau+1},\, e^{\tau} \rangle \Big)^2 \right] \\ &= \underbrace{\frac{1}{(t+1)^2} \sum_{\tau=0}^{t} \mathbb{E}\big[\langle Y^{\tau+1},\, e^{\tau} \rangle^2 \big]}_{T_1} \\ &\quad + \underbrace{\frac{2}{(t+1)^2} \sum_{0 \leq \tau_2 < \tau_1 \leq t} \mathbb{E}\big[\langle Y^{\tau_1+1},\, e^{\tau_1} \rangle \langle Y^{\tau_2+1},\, e^{\tau_2} \rangle \big]}_{T_2}. \end{aligned}$$

Since we have

$$\begin{aligned} \mathbb{E}\big[\langle Y^{\tau_1+1}, e^{\tau_1} \rangle \langle Y^{\tau_2+1},\, e^{\tau_2} \rangle \big] &= \mathbb{E}\big[\mathbb{E}\big[\langle Y^{\tau_1+1},\, e^{\tau_1} \rangle \langle Y^{\tau_2+1},\, e^{\tau_2} \rangle \mid \mathcal{F}^{\tau_1} \big] \big] \\ &= \mathbb{E}\big[\langle Y^{\tau_2+1},\, e^{\tau_2} \rangle\, \mathbb{E}\big[\langle Y^{\tau_1+1},\, e^{\tau_1} \rangle \mid \mathcal{F}^{\tau_1} \big] \big] \;=\; 0, \end{aligned}$$

for all $\tau_1 > \tau_2$, the cross product term T_2 vanishes. On the other hand, the martingale difference sequence is bounded. This can be shown as follows: from part (b) we know $\|H(m^{\tau}, J^{\tau+1})\|_2 / \|m^*\|_2 \leq \sqrt{K(\psi)}$; also using the fact that $\|\cdot\|_2$ is convex, Jensen's inequality yields $\|h(m^{\tau})\|_2 / \|m^*\|_2 \leq \sqrt{K(\psi)}$; therefore, we have

$$\begin{aligned} \|Y^{\tau+1}\|_2 &\leq \frac{\|H(m^{\tau}, J^{\tau+1})\|_2}{\|m^*\|_2} + \frac{\|h(m^{\tau})\|_2}{\|m^*\|_2} \\ &\leq 2\sqrt{K(\psi)}. \end{aligned}$$

Moving on to the first term T_1, we exploit the Cauchy Schwartz inequality in conjunction with the fact that the martingale difference sequence is bounded to obtain

$$\begin{aligned} \mathbb{E}\big[\langle Y^{\tau+1},\, e^{\tau} \rangle^2 \big] &\leq \mathbb{E}\big[\|Y^{\tau+1}\|_2^2\, \|e^{\tau}\|_2^2 \big] \\ &\leq 4\, K(\psi)\, \mathbb{E}\big[\|e^{\tau}\|_2^2 \big]. \end{aligned}$$

Taking the expectation on both sides of the inequality (41) yields $\mathbb{E}\big[\|e^{\tau}\|_2^2\big] \leq (K(\psi)/\mu^2)\,(1 + \log\tau)/\tau$; and hence we have

$$\mathbb{E}\big[\langle Y^{\tau+1},\, e^{\tau} \rangle^2 \big] \;\leq\; \frac{4\, K(\psi)^2}{\mu^2} \frac{1 + \log\tau}{\tau},$$

for all $\tau \geq 1$. Moreover, since

$$\begin{aligned} \frac{\|m^0\|_2}{\|m^*\|_2} &\leq \left(\frac{\sum_{(v \leftarrow u) \in \vec{\mathcal{E}}} \big(\max_{i \in \mathcal{X}} \overline{B}_{vu}^0(i) \big)}{\sum_{(v \leftarrow u) \in \vec{\mathcal{E}}} \big(\min_{i \in \mathcal{X}} \underline{B}_{vu}^0(i) \big)} \right)^{\frac{1}{2}} \\ &= \sqrt{\frac{K(\psi)}{4}}, \end{aligned}$$

the initial term $\mathbb{E}\big[\langle Y^1,\, e^0 \rangle^2\big] \leq 4\, K(\psi)\, \mathbb{E}\big[\|e^0\|_2^2\big]$ is upper bounded by $4K(\psi)^2$. Finally, putting all the pieces together,

we obtain

$$\begin{aligned}\operatorname{var}\left(\frac{1}{t+1}\sum_{\tau=0}^{t}\langle Y^{\tau+1},\, e^{\tau}\rangle\right) &\leq \frac{4\,K(\psi)^2}{\mu^2\,(t+1)^2}\sum_{\tau=1}^{t}\frac{1+\log\tau}{\tau} \;+\; \frac{4\,K(\psi)^2}{(t+1)^2} \\ &\overset{\text{(i)}}{\leq} \frac{4\,K(\psi)^2}{\mu^2}\,\frac{(1+\log(t+1))^2+4}{(t+1)^2},\end{aligned}$$

where inequality (i) follows from elementary inequality

$$\sum_{\tau=1}^{t}(1+\log\tau)/\tau \leq (1+\log t)^2,$$

and the fact that $\mu < 2$. Consequently, we may apply Chebyshev's inequality to control the stochastic deviation $\sum_{\tau=1}^{t+1}\langle Y^{\tau+1},\, e^{\tau}\rangle/(t+1)$. More specifically, for any $\gamma > 0$, a quantity to be specified shortly, we have

$$\begin{aligned}\mathbb{P}\Big(\big|\frac{2}{\mu\,(t+1)}\sum_{\tau=0}^{t}\langle Y^{\tau+1},\, e^{\tau}\rangle\big| \;>\; \gamma\Big) \\ \leq\; \frac{16\,K(\psi)^2}{\mu^4\,\gamma^2}\,\frac{(1+\log(t+1))^2+4}{(t+1)^2}.\end{aligned} \quad (42)$$

We now combine our earlier bound (41) with the tail bound (42), making the specific choice

$$\gamma \;=\; \frac{4\,K(\psi)}{\mu^2\,\sqrt{\epsilon}}\,\frac{\sqrt{(1+\log(t+1))^2+4}}{t+1},$$

for a fixed $0 < \epsilon < 1$, thereby concluding that

$$\begin{aligned}\|e^{t+1}\|_2^2 \;\leq\; &\frac{K(\psi)}{\mu^2}\,\frac{1+\log(t+1)}{t+1} \\ &+\; \frac{4\,K(\psi)}{\mu^2\,\sqrt{\epsilon}}\,\frac{\sqrt{(1+\log(t+1))^2+4}}{t+1},\end{aligned}$$

with probability at least $1-\epsilon$. Simplifying the last bound, we obtain

$$\|e^{t+1}\|_2^2 \;\leq\; \frac{K(\psi)}{\mu^2}\left(1+\frac{8}{\sqrt{\epsilon}}\right)\frac{1+\log(t+1)}{t+1},$$

for all $t \geq 1$, with probability at least $1-\epsilon$.

C. Proof of Proposition 1

Recall the definition (9) of the probability mass function $\{p_{vu}(j)\}_{j\in\mathcal{X}}$ used in the update of directed edge $(v \leftarrow u)$. This probability depends on the current value of the message, so we can view it as being generated by a function $q_{vu} : \mathbb{R}^D \to \mathbb{R}^d$ that performs the mapping $m \mapsto \{p_{vu}(j)\}_{j\in\mathcal{X}}$. In terms of this function, we can rewrite the BP message update equation (4) on the directed edge $(v \leftarrow u)$ as $F_{vu}(m) = \Gamma_{vu}\, q_{vu}(m)$, where the renormalized compatibility matrix Γ_{vu} was defined previously (7). We now define the $D \times D$ block diagonal matrix $\Gamma := \operatorname{blkdiag}\{\Gamma_{vu}\}_{(v\leftarrow u)\in\vec{\mathcal{E}}}$, as well as the function $q : \mathbb{R}^D \to \mathbb{R}^D$ obtained by concatenating all of the functions q_{vu}, one for each directed edge. In terms of these quantities, we rewrite the global BP message update in the compact form $F(m) = \Gamma\, q(m)$.

With these preliminaries in place, we now bound the Lipschitz constant of the mapping $F : \mathbb{R}^D \to \mathbb{R}^D$. Given an arbitrary pair of messages $m, m' \in \mathcal{S}$, we have

$$\begin{aligned}\|F(m)-F(m')\|_2^2 \;&=\; \|\Gamma\,\big(q(m)-q(m')\big)\|_2^2 \\ &=\; \sum_{(v\leftarrow u)\in\vec{\mathcal{E}}}\|\Gamma_{vu}\big(q_{vu}(m)-q_{vu}(m')\big)\|_2^2.\end{aligned} \quad (43)$$

By the Perron-Frobenius theorem [11], we know that Γ_{vu} has a unique maximal eigenvalue of 1, achieved for the left eigenvector $\vec{1} \in \mathbb{R}^d$, where $\vec{1}$ denotes the vector of all ones. Since the d-dimensional vectors $q_{vu}(m)$ and $q_{vu}(m')$ are both probability distributions, we have $\langle\vec{1},\, q_{vu}(m) - q_{vu}(m')\rangle = 0$. Therefore, we conclude that

$$\begin{aligned}&\Gamma_{vu}\big(q_{vu}(m)-q_{vu}(m')\big) \\ &\qquad=\; \Big(\Gamma_{vu}-\frac{z_{vu}\vec{1}^T}{\vec{1}^T z_{vu}}\Big)\big(q_{vu}(m)-q_{vu}(m')\big),\end{aligned}$$

where z_{vu} denotes the right eigenvector of Γ_{vu} corresponding to the eigenvalue one. Combining this equality with the representation (43), we find that

$$\begin{aligned}&\|F(m)-F(m')\|_2^2 \\ &\quad=\sum_{(v\leftarrow u)\in\vec{\mathcal{E}}}\|\big(\Gamma_{vu}-\frac{z_{vu}\vec{1}^T}{\vec{1}^T z_{vu}}\big)\big(q_{vu}(m)-q_{vu}(m')\big)\|_2^2 \\ &\quad\leq \max_{(v\leftarrow u)\in\vec{\mathcal{E}}}\|\Gamma_{vu}-\frac{z_{vu}\vec{1}^T}{\vec{1}^T z_{vu}}\|_2^2\;\;\|q(m)-q(m')\|_2^2.\end{aligned} \quad (44)$$

It remains to upper bound the Lipschitz constant of the mapping $q : \mathbb{R}^D \to \mathbb{R}^D$ previously defined.

Lemma 3. *For all $m \neq m'$, we have*

$$\frac{\|q(m)-q(m')\|_2}{\|m-m'\|_2} \;\leq\; 2\;\max_{(v\leftarrow u)\in\vec{\mathcal{E}}}\Phi_1(v,u)\;\max_{(u\leftarrow w)\in\vec{\mathcal{E}}}\Phi_2(u,w), \quad (45)$$

where the quantities $\Phi_1(v,u)$, and $\Phi_2(u,w)$ were previously defined in (20) *and* (21) .

As the proof of Lemma 3 is somewhat technical, we defer it to Appendix E. Combining the upper bound (45) with the earlier bound (44) completes the proof of the proposition.

V. Experimental Results

In this section, we present a variety of experimental results that confirm the theoretical predictions, and show that SBP is a practical algorithm. We provide results both for simulated graphical models, and real-world applications to image denoising and disparity computation.

A. Simulations on Synthetic Problems

We start by performing some simulations for the Potts model, in which the edge potentials are specified by a parameter $\gamma \in (0, 1]$, as discussed in Example 1. The node potentials are generated randomly, on the basis of fixed parameters $\mu \geq \sigma > 0$ satisfying $\mu + \sigma < 1$, as follows: for each $v \in \mathcal{V}$ and label $i \neq 1$, we generate an independent random variable $Z_{v;i}$ uniformly distributed on the interval $(-1, +1)$, and then set

$$\psi_v(i) = \begin{cases} 1 & i = 1, \\ \mu + \sigma Z_{v;i} & i \geq 2. \end{cases}$$

For a fixed graph topology and collection of node/edge potentials, we first run BP to compute the fixed point m^*.[11] We then run the SBP algorithm to find the sequence of messages $\{m^t\}_{t=0}^{\infty}$ and compute the normalized squared error $\|m^t - m^*\|_2^2 / \|m^*\|_2^2$. In cases where the normalized mean-squared error is reported, we computed it by averaging over 20 different runs of the algorithm. (Note that the runs are different, since the SBP algorithm is randomized.)

In our first set of experiments, we examine the consistency of the SBP on a chain-structured graph, as illustrated in Figure 1(b), representing a particular instance of a tree. We implemented the SBP algorithm with step size $\lambda^t = 2/(t+1)$, and performed simulations for a chain with $n = 100$ nodes, state dimension $d = 64$, node potential parameters $(\mu, \sigma) = (0.1, 0.1)$, and for two different choices of edge potential $\gamma \in \{0.02, 0.05\}$. The resulting traces of the normalized squared error versus iteration number are plotted in Figure 4; each panel contains 10 different sample paths. These plots confirm the prediction of strong consistency given in Theorem 1(a)—in particular, the error in each sample path converges to zero. We also observe that the typical performance is highly concentrated around its average, as can be observed from the small amount of variance in the sample paths.

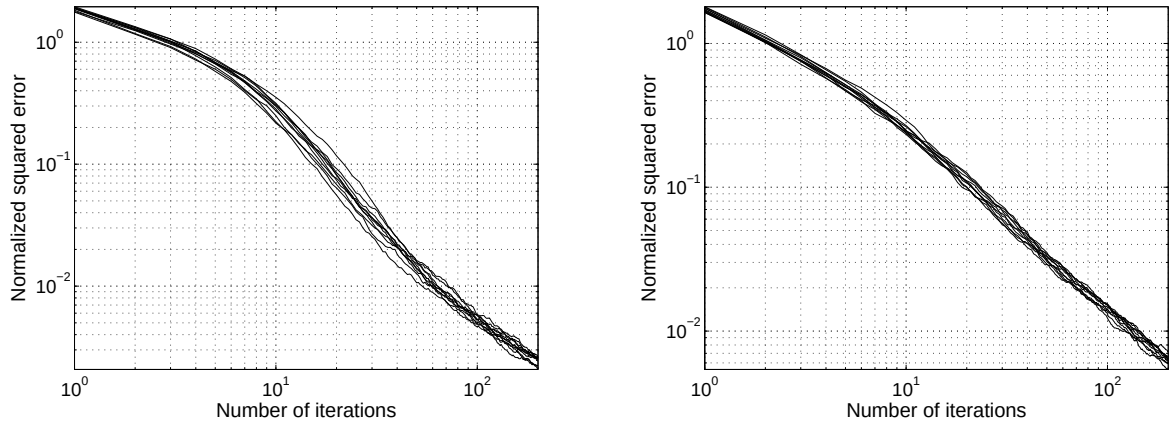
Our next set of simulations are designed to study the effect of increasing of the state dimension d on convergence rates. We performed simulations both for the chain with $n = 100$ nodes, as well as a two-dimensional square grid with $n = 100$ nodes. In all cases, we implemented the SBP algorithm with step sizes $\lambda^t = 2/(t+1)$, and generated the node/edge potentials with parameters $(\mu, \sigma) = (0.1, 0.1)$ and $\gamma = 0.1$ respectively. In Figure 5, we plot the normalized mean-squared error (estimated by averaging over 20 trials) versus the number of iterations for the chain in panel (a), and the grid in panel (b). Each panel contains four different curves, each corresponding to a choice of state dimension $d \in \{128, 256, 512, 1024\}$. For the given step size, Theorem 2 guarantees that the convergence rate should be upper bounded by $1/t^{\alpha}$ ($\alpha \leq 1$) with the number of iterations t. In the log-log domain plot, this convergence rate manifests itself as a straight line with slope $-\alpha$. For the chain simulations shown in panel (a), all four curves exhibit exactly this behavior, with the only difference with increasing dimension being a vertical shift (no change in slope). For the grid simulations in panel (b), problems with smaller state dimension exhibit somewhat faster convergence rate than predicted by theory, whereas the larger problems ($d \in \{512, 1024\}$) exhibit linear convergence on the log-log scale.

As discussed previously, the SBP message updates are less expensive by a factor of d. The top two rows of Table V-A show the per iteration running time of both BP and SBP algorithms, for different state dimensions as indicated. As predicted by theory, the SBP running time per iteration is significantly lower than BP, scaling linearly in d in contrast to the quadratic scaling of BP. To be fair in our comparison, we also measured the total computation time required for either BP or SBP to converge to the fixed point up to a δ-tolerance, with $\delta = 0.01$. This comparison allows for the fact that BP may take many fewer iterations than SBP to converge to an approximate fixed point. Nonetheless, as shown in the bottom two rows of Table V-A, in all cases except one (chain graph with dimension $d = 128$), we still see significant speed-ups from SBP in this overall running time. This gain becomes especially pronounced for larger dimensions, where these types of savings are more important.

B. Applications in Image Processing and Computer Vision

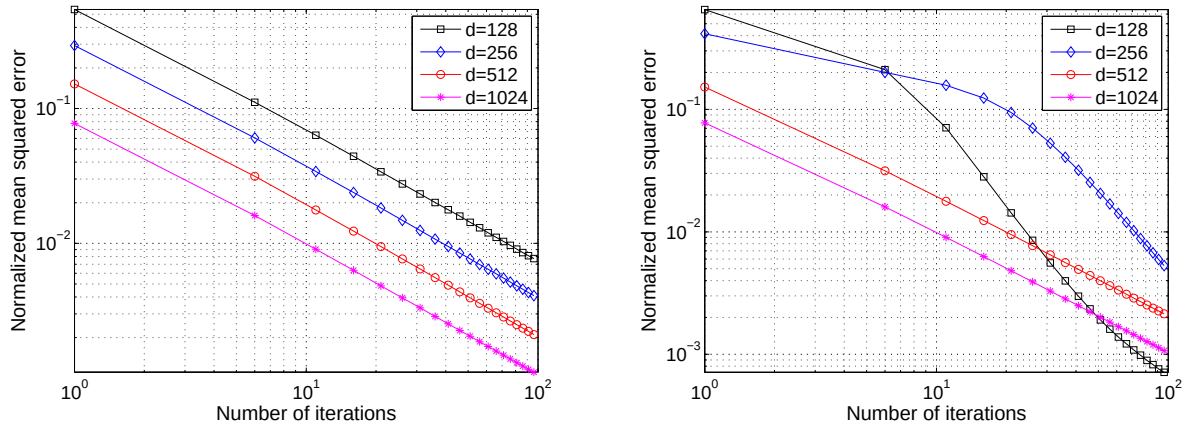
In our next set of experiments, we study the SBP on some larger scale graphs and more challenging problem instances, with applications to image processing and computer vision. Message-passing algorithms can be used for image denoising, in particular, on a two dimensional square grid where every node corresponds to a pixel. Running the BP algorithm on the graph, one can obtain (approximations to) the most likely value of every pixel based on the noisy observations. In this experiment, we consider a 200×200 image with $d = 256$ gray-scale levels, as showin in Figure 6(a). We then contaminate every pixel with an independent Gaussian random variable with standard deviation $\sigma = 0.1$, as shown in Figure 6(b). Enforcing the Potts model with smoothness parameter $\gamma = 0.05$ as the edge potential, we run BP and SBP for the total of $t = 5$ and $t = 100$ iterations, respectively, to obtain the refined images (see panels (c) and (d), respectively, in Figure 6). Figure 7 illustrates the mean-squared error versus the running time for both BP and SBP denoising. As one can observe, despite smaller jumps in the error reduction, the per-iteration running time of SBP is substantially lower than BP. Overall, SBP has done a marginally better job than BP in a substantially shorter

[11] We stop the BP iterations when $\|m^{t+1} - m^t\|_2$ becomes less than 10^{-4}.



(a) (b)

Fig. 4. The panels illustrate the normalized squared error $\|m^t - m^*\|_2^2/\|m^*\|_2^2$ versus the number of iterations t for a chain of size $n = 100$ and state dimension $d = 64$. Each plot contains 10 different sample paths. Panel (a) corresponds to the coupling parameter $\gamma = 0.02$ whereas panel (b) corresponds to $\gamma = 0.05$. In all cases, the SBP algorithm was implemented with step size $\lambda^t = 2/(t+1)$, and the node potentials were generated with parameters $(\mu, \sigma) = (0.1, 0.1)$.



(a) (b)

Fig. 5. Effect of increasing state dimension on convergence rates. Plots of the normalized mean-squared error $\mathbb{E}\big[\|m^t - m^*\|_2^2\big]/\|m^*\|_2^2$ versus the number of iterations for two different graphs: (a) chain with $n = 100$ nodes, and (b) two-dimensional square grid with $n = 100$ nodes. In both panels, each curve corresponds different state dimension $d \in \{128, 256, 512, 1024\}$. All simulations were performed with step sizes $\lambda^t = 2/(t+1)$, and the node/edge parameters were generated with parameters $(\mu, \sigma) = (0.1, 0.1)$ and $\gamma = 0.1$ respectively.

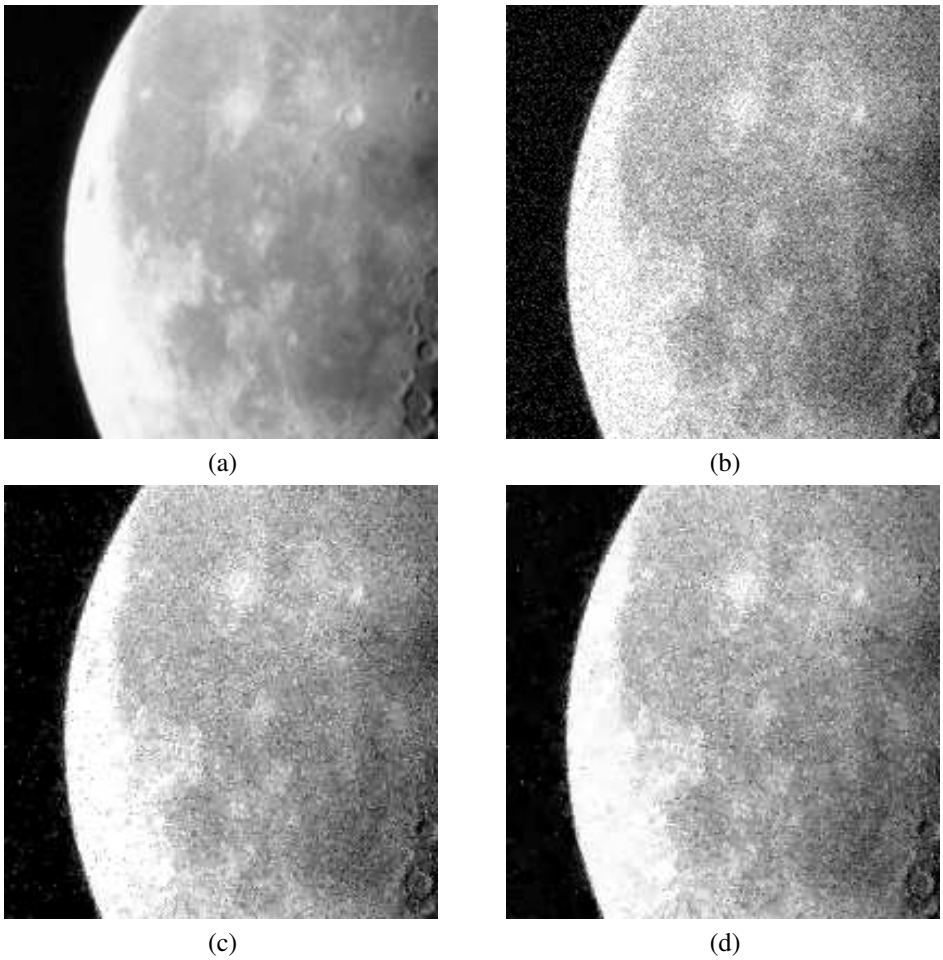
amount of time in this instance.[12]

Finally, in our last experiment, we apply SBP to a computer vision problem. Graphical models and message-passing algorithms are popular in application to the stereo vision problem [28], [16], in which the goal is to estimate objects depth based on the pixel dissimilarities in two (left and right view) images. Adopting the original model in Sun et al. [28], we again use a form of the Potts model in order to enforce a smoothness prior, and also use the form of the observation potentials given in the Sun et al. paper. We then run BP and SBP (with step size $3/(t+2)$) for a total of $t = 10$ and $t = 50$ iterations respectively in order to estimate the pixel dissimilarities. The results for the test image "map" are presented in Figure 8. Here, the maximum pixel dissimilarity is $d = 32$, which makes stereo vision a relatively low-dimensional problem. In this particular application, the SBP is faster by about a factor of $3 - 4$ times per iteration; however, the need to run more iterations makes it comparable to BP. This is to be expected since the state dimension $d = 32$ is relatively small, and the relative advantage of SBP becomes more significant for

[12]Note that the purpose of this experiment is not to analyze the potential of SBP (or for that matter BP) in image denoising, but to rather observe their relative performances and computational complexities.

		$d = 128$	$d = 256$	$d = 512$	$d = 1024$
Chain	BP (per iteration)	0.0700	0.2844	2.83	18.0774
	SBP (per iteration)	0.0036	0.0068	0.0145	0.0280
	BP (total)	0.14	0.57	5.66	36.15
	SBP (total)	0.26	0.27	0.29	0.28
Grid	BP (per iteration)	0.1300	0.5231	5.3125	32.5050
	SBP (per iteration)	0.0095	0.0172	0.0325	0.0620
	BP (total)	0.65	3.66	10.63	65.01
	SBP (total)	0.21	1.31	0.65	0.62

TABLE I. Comparison of BP and SBP computational cost for two different graphs each with $n = 100$ nodes. For each graph type, the top two rows show per iteration running time (in seconds) of the BP and SBP algorithms for different state dimensions. The bottom two rows show total running time (in seconds) to compute the message fixed point to $\delta = 0.01$ accuracy.



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

Fig. 6. Image denoising application, (a) original image, (b) noisy image, (c) refined image obtained from BP after $t = 5$ iterations, and (d) refined image obtained from SBP after $t = 100$ iterations. The image is 200×200 with $d = 256$ gray-scale levels. The SBP step size, the Potts model parameter, and noise standard deviation are set to $\lambda^t = 1/(t + 1)$, $\gamma = 0.05$, and $\sigma = 0.1$, respectively.

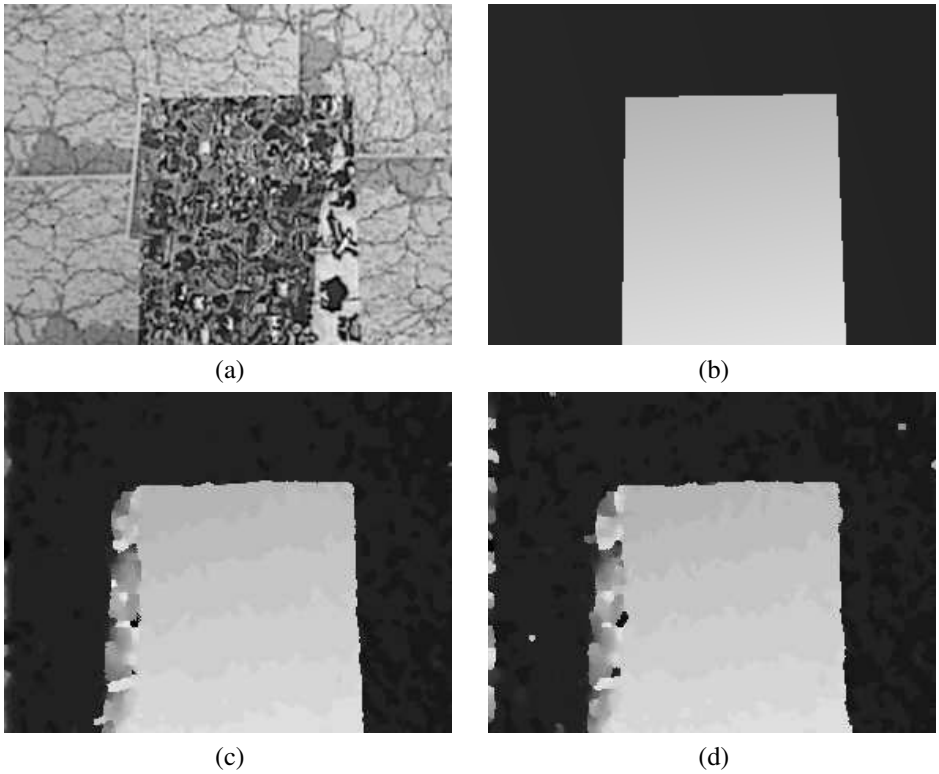


Fig. 8. Stereo vision, depth recognition, application, (a) reference image, (b) ground truth, (c) BP estimate after $t = 10$ iterations, and (d) SBP estimate after $t = 50$ iterations. The algorithms are applied to the standard "map" image with maximum pixel dissimilarity $d = 32$. The SBP step size is set to $\lambda^t = 3/(t+2)$.

larger state dimensions d.

VI. Discussion

In this paper, we have developed and analyzed a new and low-complexity alternative to BP message-passing. The SBP algorithm has per iteration computational complexity that scales linearly in the state dimension d, as opposed to the quadratic dependence of BP, and a communication cost of $\log_2 d$ bits per edge and iteration, as opposed to $d - 1$ real numbers for standard BP message updates. Stochastic belief propagation is also easy to implement, requiring only random number generation and the usual distributed updates of a message-passing algorithm. Our main contribution was to prove a number of theoretical guarantees for the SBP message updates, including convergence for any tree-structured problem, as well as for general graphs for which the ordinary BP message update satisfies a suitable contraction condition. In addition, we provided non-asymptotic upper bounds on the SBP error, both in expectation and in high probability.

The results described here suggest a number of directions for future research. First, the ideas exploited here have natural generalizations to problems involving continuous random variables and also other algorithms that operate over the sum-product semi-ring, including the generalized belief propagation algorithm [33] as well as reweighted sum-product algorithms [31]. More generally, the BP algorithm can be seen as optimizing the dual of the Bethe free energy function [33], and it would be interesting to see if SBP can be interpreted as a stochastic version of this Bethe free energy minimization. It is also natural to consider whether similar ideas can be applied to analyze stochastic forms of message-passing over other semi-rings, such as the max-product algebra that underlies the computation of maximum a posteriori (MAP) configurations in graphical models. In this paper, we have developed SBP for applications to Markov random fields with pairwise interactions. In principle, any undirected graphical model with discrete variables can be reduced to this form [33], [32]; however, in certain applications, such as decoding of LDPC codes over non-binary state spaces, this could be cumbersome. For such cases, it would be useful to derive a variant of SBP that applies directly to factor graphs with higher-order interactions. Moreover, the results derived in this paper are based on the assumption that the co-domain of the potential functions do not include zero. We suspect that these condition might be relaxed, and similar results could be obtained. Finally, our analysis for general graphs has been done under a contractivity condition, but it is likely that this requirement could be loosened. Indeed, the SBP algorithm

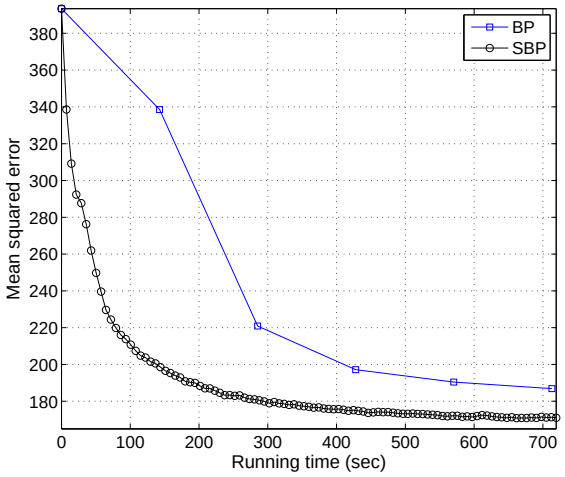


Fig. 7. Mean-squared error versus the running time (in seconds) for both BP and SBP image denoising. The simulations are performed with the step size $\lambda^t = 1/(t+1)$, and the Potts model parameter $\gamma = 0.05$ on a 200×200 image with $d = 256$ gray-scale levels. The noise is assumed to be additive, independent Gaussian random variables with standard deviation $\sigma = 0.1$.

works well for many problems where this condition need not be satisfied.

Acknowledgements

Both authors were partially supported by MURI grant N00014-11-1-0688 to MJW. Both authors would like to thank Alekh Agarwal for helpful discussions on stochastic approximation and optimization at the initial phases of this research, the anonymous reviewers for their helpful feedback, as well as Associate Editor Pascal Vontobel for his careful reading and detailed suggestions that helped to improve the paper.

APPENDIX

A. Details of Example 1

In this appendix, we verify the sufficient condition for contractivity (25). Recall the definition (13) of the zero'th order bounds. By construction, we have the relations

$$\underline{B}_{vu}(i) = \underline{B}^0_{vu}(i) = \frac{\gamma}{1+(d-1)\gamma}, \quad \text{and}$$
$$\overline{B}_{vu}(i) = \overline{B}^0_{vu}(i) = \frac{1}{1+(d-1)\gamma}$$

for all $i \in \mathcal{X}$ and $(v \leftarrow u) \in \vec{\mathcal{E}}$. Substituting these bounds into the definitions (22) and (23) and doing some simple algebra yields the upper bounds

$$\phi_{vu,uw} \leq \max_{j \in \mathcal{X}} \left\{ \frac{\beta_{vu}(j) \prod_{s \in \mathcal{N}(u) \setminus \{v,w\}} \overline{B}_{su}(j)}{\sum_{\ell=1}^d \beta_{vu}(\ell) \prod_{s \in \mathcal{N}(u) \setminus v} \underline{B}_{su}(\ell)} \right\}$$
$$= \frac{1+(d-1)\gamma}{\gamma^{\rho_u - 1}} \max_{j \in \mathcal{X}} \left\{ \frac{\psi_u(j)}{\sum_{\ell=1}^d \psi_u(\ell)} \right\},$$

and

$$\chi_{vu,uw}$$
$$\leq \max_{j \in \mathcal{X}} \left\{ \frac{\beta_{vu}(j) \prod_{s \in \mathcal{N}(u) \setminus v} \overline{B}_{su}(j)}{\sum_{\ell=1}^d \beta_{vu}(\ell) \prod_{s \in \mathcal{N}(u) \setminus v} \underline{B}_{su}(\ell)} \right\} \max_{j \in \mathcal{X}} \left\{ \frac{1}{\underline{B}_{uw}(j)} \right\}$$
$$= \frac{1+(d-1)\gamma}{\gamma^{\rho_u}} \max_{j \in \mathcal{X}} \left\{ \frac{\psi_u(j)}{\sum_{\ell=1}^d \psi_u(\ell)} \right\},$$

where we have denoted the degree of the node u by ρ_u. Substituting these inequalities into expression (24) and noting that $\gamma \leq 1$, we find that the global update function has Lipschitz constant at most

$$L \leq 4\,(1-\gamma)(1+(d-1)\gamma)$$
$$\max_{u \in \mathcal{V}} \left\{ \frac{(\rho_u - 1)^2}{\gamma^{2\rho_u}} \max_{j \in \mathcal{X}} \left\{ \frac{\psi_u(j)}{\sum_\ell \psi_u(\ell)} \right\}^2 \right\},$$

as claimed.

B. Proof of Lemma 1

By construction, for each directed edge $(v \leftarrow u)$, the message vector m_{vu} belongs to the probability simplex—that is, $\sum_{i \in \mathcal{X}} m_{vu}(i) = 1$, and $m_{vu} \succeq \vec{0}$. From equation (26), the vector m_{vu} is a convex combination of the columns of the matrix Γ_{vu}. Recalling bounds (13), we conclude that the message vector must belong to the set $\mathcal{S}$, as defined in (18), in particular with $\underline{B}_{vu}(i) = \underline{B}^0_{vu}(i)$ and $\overline{B}_{vu}(i) = \overline{B}^0_{vu}(i)$. Note that the set $\mathcal{S}$ is compact, and any member of it has strictly positive elements under our assumptions.

For directed edges $(v \leftarrow u)$ and $(s \leftarrow w)$, let $\frac{\partial F_{vu}}{\partial m_{sw}} \in \mathbb{R}^{d \times d}$ denote the Jacobian matrix obtained from taking the partial derivative of the update function F_{vu} with respect to the message vector m_{sw}. By inspection, the function F_{vu} is continuously differentiable; consequently, the function $\frac{\partial F_{vu}(i;m)}{\partial m_{sw}(j)}$ is continuous, and hence must achieve its supremum over the compact set $\mathcal{S}$. We may use these Jacobian matrices to define a matrix $A_{vu,sw} \in \mathbb{R}^{d \times d}$ with entries

$$A_{vu,sw}(i,j) := \max_{m \in \mathcal{S}} \left| \frac{\partial F_{vu}(i;m)}{\partial m_{sw}(j)} \right|, \quad \text{for } i,j = 1,\ldots,d.$$

We then use these matrices to define a larger matrix $A \in \mathbb{R}^{D \times D}$, consisting of $2|\mathcal{E}| \times 2|\mathcal{E}|$ sub-blocks each of size $d \times d$, with the sub-blocks indexed by pairs of directed edges $(v \leftarrow u) \in \vec{\mathcal{E}}$. In particular, the matrix $A_{vu,sw}$ occupies the sub-block indexed by the edge pair $(v \leftarrow u)$ and $(s \leftarrow w)$. Note that by the structure of the

update function F, the matrix $A_{vu,sw}$ can be non-zero only if $s = u$ and $w \in \mathcal{N}(u) \backslash \{v\}$.

Now let $\nabla F \in \mathbb{R}^{D \times D}$ denote the Jacobian matrix of the update function F. By the integral form of the mean value theorem, we have the representation

$$F(m) - F(m') \;=\; \left[\int_0^1 \nabla F(m' + \tau(m - m'))\, d\tau \right] (m - m').$$

Applying triangle inequality separately to each component of this D-dimensional vector and then using the definition of A, we obtain the elementwise upper bound

$$|F(m) - F(m')| \;\preceq\; A\, |m - m'|.$$

It remains to show that A is nilpotent: more precisely, we show that A^r is the all-zero matrix, where $r = \operatorname{diam}(\mathcal{G})$ denotes the diameter of the graph $\mathcal{G}$. In order to do so, we first let $B \in \mathbb{R}^{2|\mathcal{E}| \times 2|\mathcal{E}|}$ be the "block indicator" matrix—that is, its entries are given by

$$B(v \leftarrow u, s \leftarrow w) \;=\; \begin{cases} 1 & \text{if } A_{vu,sw} \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Based on this definition, it is straightforward to verify that if $B^r = 0$ for some positive integer r, then we also have $A^r = 0$. Consequently, it suffices to show that $B^r = 0$ for $r = \operatorname{diam}(\mathcal{G})$.

Fix a pair of directed edges $(v \leftarrow u)$ and $(s \leftarrow w)$, and some integer $\ell \geq 1$. We first claim that the matrix entry $B^\ell(v \leftarrow u, s \leftarrow w)$ is non-zero only if there exists a backtrackless *directed path* of length $\ell + 1$ from w to v that includes both s and u, meaning that there exist nodes $s_1, s_2, \ldots, s_{\ell-2}$ such that

$$w \in \mathcal{N}(s) \backslash s_1, \;\; s_1 \in \mathcal{N}(s_2) \backslash s_3, \ldots, \;\text{ and }\; s_{\ell-2} \in \mathcal{N}(u) \backslash v.$$

We prove this claim via induction. The base case $\ell = 1$ is true by construction. Now supposing that the claim holds at order ℓ, we show that it must hold at order $\ell + 1$. By definition of matrix multiplication, we have

$$\begin{aligned} &B^{\ell+1}(v \leftarrow u, s \leftarrow w) \\ &\qquad = \sum_{(x \leftarrow y) \in \vec{\mathcal{E}}} B^\ell(v \leftarrow u, x \leftarrow y)\, B(x \leftarrow y, s \leftarrow w). \end{aligned}$$

In order for this entry to be non-zero, there must exist a directed edge $(x \leftarrow y)$ that forms a $(\ell + 1)$-directed path to $(v \leftarrow u)$, and moreover, we must have $s = y$, and $w \in \mathcal{N}(x) \setminus y$. These conditions are equivalent of having a backtrackless directed path of length $\ell + 2$ from w to v, with s and u as intermediate nodes, thereby completing the proof of our intermediate claim.

Finally, we observe that in a tree-structured graph, there can be no directed path of length greater than $r = \operatorname{diam}(\mathcal{G})$. Consequently, our intermediate claim implies that $B^r = 0$ for any tree-structured graph, which completes the proof.

C. A version of Robbins-Monro theorem

Here we state a version of the Robbins-Monro theorem suitable for our proof of Theorem 2. Denoting the expected vector field function by $h(m) := \mathbb{E}\big[H(m, J)|m\big]$, suppose there exists a vector m^* such that

$$\inf_{m \in \mathcal{S} \backslash \{m^*\}} \langle m - m^*,\, h(m) \rangle \;>\; 0.$$

Now suppose that

- the vector field function $H(m, \cdot)$ has a bounded second moment—that is $\mathbb{E}\big[\|H(m, J)\|_2^2\big] \;\leq\; c(1 + \|m\|_2^2)$ for some constant c,
- the conditional distribution of the random vector J^{t+1} knowing the past depends only on m^t—that is $\mathbb{P}\big(J^{t+1} | J^t, J^{t-1}, \cdots, m^t, m^{t-1}, \cdots\big) \;=\; \mathbb{P}\big(J^{t+1} | m^t\big)$, and finally,
- the step size sequence $\{\lambda^t\}_{t=0}^\infty$ satisfies the conditions $\sum_{t=0}^\infty \lambda^t = \infty$, and $\sum_{t=0}^\infty (\lambda^t)^2 < \infty$.

Then the sequence $\{m^t\}_{t=0}^\infty$, generated by (33), is guaranteed to converge to m^*

D. Proof of Lemma 2

Noting that it is equivalent to bound the logarithm, we have

$$\log \prod_{\ell=i+1}^{t+2} \left(1 - \frac{\alpha}{\ell}\right) \;=\; \sum_{\ell=i+1}^{t+2} \log \left(1 - \frac{\alpha}{\ell}\right) \;\leq\; -\alpha \sum_{\ell=i+1}^{t+2} \frac{1}{\ell}, \tag{46}$$

where we used the fact that $\log(1 - x) \leq -x$ for $x \in (0, 1)$. Since the function $1/x$ is decreasing, we have

$$\sum_{\ell=i+1}^{t+2} \frac{1}{\ell} \;\geq\; \int_{i+1}^{t+3} \frac{1}{x}\, dx \;=\; \log(t + 3) \;-\; \log(i + 1). \tag{47}$$

Substituting inequality (47) into (46) yields $\log \prod_{\ell=i+1}^{t+2} \big(1 - \frac{\alpha}{\ell}\big) \;\leq\; \alpha \,\big(\log(i + 1) \;-\; \log(t + 3)\big)$, from which the claim stated in the lemma follows.

E. Proof of Lemma 3

Let $\nabla q(m) \in \mathbb{R}^{D \times D}$ denote the Jacobian matrix of the function $q : \mathbb{R}^D \to \mathbb{R}^D$ evaluated at m. Since q is differentiable, we can apply the integral form of the mean value theorem to write

$$q(m) - q(m') = \big[\int_0^1 \nabla q(m' + \tau(m - m'))\, d\tau \big]\, (m - m').$$

From this representation, we obtain the upper bound

$$\begin{aligned} &\|q(m) - q(m')\|_2 \\ &\qquad \leq \left[\int_0^1 |\!|\!| \nabla q(m' + \lambda(m - m'))|\!|\!|_2\, d\lambda \right] \|(m - m')\|_2 \\ &\qquad \leq \sup_{m \in \mathcal{S}} |\!|\!| \nabla q(m) |\!|\!|_2\, \|m - m'\|_2, \end{aligned}$$

showing that it suffices to control the quantity $\sup_{m\in\mathcal{S}} \|\!|\nabla q(m)|\!\|_2$.

Let $\frac{\partial q_{vu}(m)}{\partial m_{sw}}$ be the $d \times d$ matrix of partial derivatives of the function $q_{vu} : \mathbb{R}^D \to \mathbb{R}^d$ obtained from taking the partial derivatives with respect to the message vector $m_{sw} \in \mathbb{R}^d$. We then define a $2|\mathcal{E}| \times 2|\mathcal{E}|$-dimensional matrix A with the entries

$$A(v \leftarrow u, s \leftarrow w) := \begin{cases} \sup_{m\in\mathcal{S}} \|\!|\frac{\partial q_{vu}(m)}{\partial m_{sw}}|\!\|_2 & \text{if } s = u, \text{ and } w \in \mathcal{N}(u)\backslash\{v\} \\ 0 & \text{otherwise.} \end{cases} \tag{48}$$

Our next step is to show that $\sup_{m\in\mathcal{S}} \|\!|\nabla q(m)|\!\|_2 \leq \|\!|A|\!\|_2$. Let $y = \{y_{vu}\}_{(v\leftarrow u)\in\vec{\mathcal{E}}}$ be an arbitrary D-dimensional vector, where each sub-vector y_{vu} is an element of $\mathbb{R}^d$. By exploiting the structure of $\nabla q(m)$ and y, we have

$$\begin{aligned} \|\nabla q(m)\, y\|_2^2 &= \sum_{(v\leftarrow u)\in\vec{\mathcal{E}}} \| \sum_{w\in\mathcal{N}(u)\backslash\{v\}} \frac{\partial q_{vu}(m)}{\partial m_{uw}}\, y_{uw}\|_2^2 \\ &\overset{\text{(i)}}{\leq} \sum_{(v\leftarrow u)\in\vec{\mathcal{E}}} \Big(\sum_{w\in\mathcal{N}(u)\backslash\{v\}} \|\frac{\partial q_{vu}(m)}{\partial m_{uw}}\, y_{uw}\|_2 \Big)^2 \\ &\overset{\text{(ii)}}{\leq} \sum_{(v\leftarrow u)\in\vec{\mathcal{E}}} \Big(\sum_{w\in\mathcal{N}(u)\backslash\{v\}} \|\!|\frac{\partial q_{vu}(m)}{\partial m_{uw}}|\!\|_2 \|y_{uw}\|_2 \Big)^2 \\ &\overset{\text{(iii)}}{\leq} \sum_{(v\leftarrow u)\in\vec{\mathcal{E}}} \Big(\sum_{w\in\mathcal{N}(u)\backslash\{v\}} A(v \leftarrow u, u \leftarrow w) \|y_{uw}\|_2 \Big)^2, \end{aligned}$$

where the bound (i) follows by triangle inequality; the bound (ii) follows from definition of the operator norm; and the final inequality (iii) follows by definition of A.

Defining the vector $z \in \mathbb{R}^{2|\mathcal{E}|}$ with the entries $z_{uw} = \|y_{uw}\|_2$, we have established the upper bound $\|\nabla q(m)\, y\|_2^2 \leq \|Az\|_2^2$, and hence that

$$\|\nabla q(m)\, y\|_2^2 \leq \|\!|A|\!\|_2^2\, \|z\|_2^2 = \|\!|A|\!\|_2^2\, \|y\|_2^2,$$

where the final equality uses the fact that $\|y\|_2^2 = \|z\|_2^2$ by construction. Since both the message m and vector y were arbitrary, we have shown that $\sup_{m\in\mathcal{S}} \|\!|\nabla q(m)|\!\|_2 \leq \|\!|A|\!\|_2$, as claimed.

Our final step is to control the quantities $\sup_{m\in\mathcal{S}} \|\!|\frac{\partial q_{vu}(m)}{\partial m_{sw}}|\!\|_2$ that define the entries of A. In this argument, we make repeated use of the elementary matrix inequality [11]

$$\|\!|B|\!\|_2^2 \leq \underbrace{\Big(\max_{i=1,\ldots,n} \sum_{j=1}^n |B_{ij}| \Big)}_{\|\!|B|\!\|_\infty} \underbrace{\Big(\max_{j=1,\ldots,n} \sum_{i=1}^n |B_{ij}| \Big)}_{\|\!|B|\!\|_1}, \tag{49}$$

valid for any $n \times n$ matrix.

Recall the definition of the probability distribution (9) that defines the function $q_{vu} : \mathbb{R}^D \to \mathbb{R}^d$, as well as our shorthand notation $M_{vu}(k) = \prod_{w\in\mathcal{N}(u)\backslash\{v\}} m_{uw}(k)$. Taking the derivatives and performing some algebra yields

$$\begin{aligned} \frac{\partial q_{vu}(i\,;\,m)}{\partial m_{uw}(j)} &= \sum_{k=1}^d \frac{\partial q_{vu}(i\,;\,m)}{\partial M_{vu}(k)} \frac{\partial M_{vu}(k)}{\partial m_{uw}(j)} \\ &= \frac{\partial q_{vu}(i\,;\,m)}{\partial M_{vu}(j)} \frac{M_{vu}(j)}{m_{uw}(j)} \\ &= \frac{-\beta_{vu}(i)\, M_{vu}(i)\, \beta_{vu}(j)}{\big(\sum_{k=1}^d \beta_{vu}(k) M_{vu}(k)\big)^2} \frac{M_{vu}(j)}{m_{uw}(j)}, \end{aligned}$$

for $i \neq j$, and $w \in \mathcal{N}(u)\backslash\{v\}$. For $i = j$, we have

$$\begin{aligned} \frac{\partial q_{vu}(i\,;\,m)}{\partial m_{uw}(i)} &= \frac{\partial q_{vu}(i\,;\,m)}{\partial M_{vu}(i)} \frac{M_{vu}(i)}{m_{uw}(i)} \\ &= \Big[\frac{\beta_{vu}(i)}{\sum_{k=1}^d \beta_{vu}(k) M_{vu}(k)} - \frac{\beta_{vu}(i)^2 M_{vu}(i)}{\big(\sum_{k=1}^d \beta_{vu}(k) M_{vu}(k)\big)^2} \Big] \frac{M_{vu}(i)}{m_{uw}(i)}. \end{aligned}$$

Putting together the pieces leads to the upper bounds

$$\|\!|\frac{\partial q_{vu}(m)}{\partial m_{uw}}|\!\|_1 \leq 2 \max_{j\in\mathcal{X}} \Big\{ \frac{\beta_{vu}(j)\, M_{vu}(j)}{\sum_{k=1}^d \beta_{vu}(k)\, M_{vu}(k)} \frac{1}{m_{uw}(j)} \Big\},$$

and

$$\begin{aligned} &\|\!|\frac{\partial q_{vu}(m)}{\partial m_{uw}}|\!\|_\infty \\ &\leq \max_{i\in\mathcal{X}} \Big\{ \frac{\beta_{vu}(i)\, M_{vu}(i)}{\sum_{k=1}^d \beta_{vu}(k)\, M_{vu}(k)} \frac{1}{m_{uw}(i)} \\ &\quad + \frac{\beta_{vu}(i)\, M_{vu}(i)}{\big(\sum_{k=1}^d \beta_{vu}(k)\, M_{vu}(k)\big)^2} \sum_{j=1}^d \frac{\beta_{vu}(j)\, M_{vu}(j)}{m_{uw(j)}} \Big\}. \end{aligned}$$

Recalling the definitions (22) and (23) of $\phi_{vu,uw}$ and $\chi_{vu,uw}$ respectively, we find that

$$\|\!|\frac{\partial q_{vu}(m)}{\partial m_{uw}}|\!\|_1 \leq 2\, \phi_{vu,uw},$$

and

$$\|\!|\frac{\partial q_{vu}(m)}{\partial m_{uw}}|\!\|_\infty \leq \phi_{vu,uw} + \chi_{vu,uw}.$$

Thus, by applying inequality (49) with $B = \frac{\partial q_{vu}(m)}{\partial m_{uw}}$, we conclude that

$$\|\!|\frac{\partial q_{vu}(m)}{\partial m_{uw}}|\!\|_2^2 \leq 2\, \phi_{vu,uw}\, (\phi_{vu,uw} + \chi_{vu,uw}).$$

Since this bound holds for any message $m \in \mathcal{S}$, we conclude that each of the matrix entries $A(v \leftarrow u, u \leftarrow w)$ satisfies the same inequality. Again applying the basic matrix

inequality (49), this time with $B = A$, we conclude that $\|\!|A|\!\|_2$ is upper bounded by

$$2 \max_{(v \leftarrow u) \in \vec{\mathcal{E}}} \sum_{w \in \mathcal{N}(u) \setminus \{v\}} \left(\phi_{vu,uw} \left(\phi_{vu,uw} + \chi_{vu,uw}\right)\right)^{\frac{1}{2}}$$
$$\max_{(u \leftarrow w) \in \vec{\mathcal{E}}} \sum_{v \in \mathcal{N}(u) \setminus w} \left(\phi_{vu,uw} \left(\phi_{vu,uw} + \chi_{vu,uw}\right)\right)^{\frac{1}{2}},$$

which concludes the proof.

REFERENCES

[1] R. P. Agarwal, M. Meehan, and D. O'Regan. *Fixed Point Theory and Applications*. Cambridge University Press, 2004.

[2] S. M. Aji and R. J. McEliece. The generalized distributive law. *IEEE Transactions on Information Theory*, 46(2):325–343, March 2000.

[3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transaction on Signal Processing*, 50(2):174–188, 2002.

[4] A. Benveniste, M. Metivier, and P. Priouret. *Adaptive Algorithms and Stochastic Approximations*. Springer-Verlag, New York, NY, 1990.

[5] F. Chung and L. Lu. Concentration inequalities and martingale inequalities: A survey. *Internet Mathematics*, 3(1):79–127, 2006.

[6] J. Coughlan and H. Shen. Dynamic quantization for belief propagation in sparse spaces. *Computer Vision and Image Understanding*, 106(1):47–58, 2007.

[7] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.

[8] R. Durrett. *Probability: Theory and Examples*. Duxbury Press, New York, NY, 1995.

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41–54, 2006.

[10] R. G. Gallager. *Low-Density Parity-Check Codes*. PhD thesis, Cambridge, MA, 1963.

[11] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1985.

[12] A. T. Ihler, J. W. Fisher, and A. S. Willsky. Loopy belief propagation: convergence and effects of message errors. *Journal of Machine Learning Research*, 6:905–936, May 2005.

[13] A. T. Ihler and D. McAllester. Particle belief propagation. In *Proceedings Conference on Artificial Intelligence and Statistics, Clearwater, Florida, USA*, pages 256–263, 2009.

[14] M. Isard, J. MacCormick, and K. Achan. Continuously-adaptive discretization for message-passing algorithms. In *Proceedings Advances in Neural Information Processing Systems, Vancouver, Canada*, pages 737–744, 2009.

[15] K. Kersting, B. Ahmadi, and S. Natarajan. Counting belief propagation. In *Proceedings Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, Canada*, 2009.

[16] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Proceedings 18th International Conference on Pattern Recognition, Hong Kong*, pages 15–18, 2006.

[17] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transaction on Information Theory*, 47(2):498–519, 2001.

[18] H.-A. Loeliger. An introduction to factor graphs. *IEEE Signal Processing Magazine*, 21:28–41, 2004.

[19] J. J. McAuley and T. S. Caetano. Faster algorithms for max-product message passing. *Journal of Machine Learning Research*, 12:1349–1388, 2011.

[20] J. M. Mooij and H. J. Kappen. Sufficient conditions for convergence of the sum-product algorithm. *IEEE Transactions on Information Theory*, 53(12):4422–4437, December 2007.

[21] A. C. Rapley, C. Winstead, V. C. Gaudet, and C. Schlegel. Stochastic iterative decoding on factor graphs. In *Proceedings 3rd International Symposium on Turbo Codes and Related Topics, Brest, France*, pages 507–510, 2003.

[22] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[23] T. G. Roosta, M. J. Wainwright, and S. S. Sastry. Convergence analysis of reweighted sum-product algorithms. *IEEE Transactions on Signal Processing*, 56(9):4293–4305, September 2008.

[24] H. L. Royden. *Real Analysis*. Prentice-Hall, New Jersey, 1988.

[25] H. Song and J. R. Cruz. Reduced-complexity decoding of q-ary LDPC codes for magnetic recording. *IEEE Transaction on Magnetics*, 39(2):1081–1087, 2003.

[26] L. Song, A. Gretton, D. Bickson, Y. Low, and C. Guestrin. Kernel belief propagation. In *Proceedings Artificial Intelligence and Statistics, Ft. Lauderdale, Florida, USA*, 2011.

[27] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, volume 1, pages 605–612, 2003.

[28] J. Sun, H. Y. Shum, and N. N. Zheng. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800, 2003.

[29] S. Tatikonda and M. I. Jordan. Loopy belief propagation and Gibbs measures. In *Proceedings 18th confernce on Uncertainty in Artificial Intelligence, Alberta, Canada*, volume 18, pages 493–500, August 2002.

[30] S. S. Tehrani, W. J. Gross, and S. Mannor. Stochastic decoding of LDPC codes. *IEEE Communications Letters*, 10(10):716–718, 2006.

[31] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transaction on Information Theory*, 51(7):2313–2335, July 2005.

[32] M. J. Wainwright and M. I. Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc, Hanover, MA 02339, USA, 2008.

[33] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transaction on Information Theory*, 51(7):2282–2312, July 2005.