Belief Propagation for Continuous State Spaces: Stochastic Message-Passing with Quantitative Guarantees

Nima Noorshams   NNOORSHAMS@CAL.BERKELEY.EDU
Martin J. Wainwright   WAINWRIG@EECS.BERKELEY.EDU
Department of Electrical Engineering & Computer Sciences
University of California Berkeley
Berkeley, CA 94720-1770, USA

Abstract

The sum-product or belief propagation (BP) algorithm is a widely used message-passing technique for computing approximate marginals in graphical models. We introduce a new technique, called stochastic orthogonal series message-passing (SOSMP), for computing the BP fixed point in models with continuous random variables. It is based on a deterministic approximation of the messages via orthogonal series basis expansion, and a stochastic estimation of the basis coefficients via Monte Carlo techniques and damped updates. We prove that the SOSMP iterates converge to a δ-neighborhood of the unique BP fixed point for any tree-structured graph, and for any graphs with cycles in which the BP updates satisfy a contractivity condition. In addition, we demonstrate how to choose the number of basis coefficients as a function of the desired approximation accuracy δ and smoothness of the compatibility functions. We illustrate our theory with both simulated examples and in application to optical flow estimation.

Keywords: graphical models, sum-product for continuous state spaces, low-complexity belief propagation, stochastic approximation, Monte Carlo methods, orthogonal basis expansion

1. Introduction

Graphical models provide a parsimonious yet flexible framework for describing probabilistic dependencies among large numbers of random variables. They have proven useful in a variety of application domains, including computational biology, computer vision and image processing, data compression, and natural language processing, among others. In all of these applications, a central computational challenge is the *marginalization problem*, by which we mean the problem of computing marginal distributions over some subset of the variables. Naively approached, such marginalization problems become intractable for all but toy problems, since they entail performing summation or integration over high-dimensional spaces. The sum-product algorithm, also known as belief propagation (BP), is a form of dynamic programming that can be used to compute exact marginals much more efficiently for graphical models without cycles, known as trees. It is an iterative algorithm in which nodes in the graph perform a local summation/integration operation, and then relay results to their neighbors in the form of messages. Although it is guaranteed to be exact on trees, it is also commonly applied to graphs with cycles, in which context it is often known as loopy BP. For more details on graphical models and BP, we refer the readers to the papers by Kschischang

et al. (2001), Wainwright and Jordan (2008), Aji and McEliece (2000), Loeliger (2004) and Yedidia et al. (2005).

In many applications of graphical models, we encounter random variables that take on continuous values (as opposed to discrete). For instance, in computer vision, the problem of optical flow calculation is most readily formulated in terms of estimating a vector field in $\mathbb{R}^2$. Other applications involving continuous random variables include tracking problems in sensor networks, vehicle localization, image geotagging, and protein folding in computational biology. With certain exceptions (such as multivariate Gaussian problems), the marginalization problem is very challenging for continuous random variables: in particular, the messages correspond to functions, so that they are expensive to compute and transmit, in which case BP may be limited to small-scale problems. Motivated by this challenge, researchers have proposed different techniques to reduce complexity of BP in different applications (Arulampalam et al., 2002; Sudderth et al., 2010; Isard, 2003; Doucet et al., 2001; Ihler and McAllester, 2009; Coughlan and Shen, 2007; Isard et al., 2009; Song et al., 2011; Noorshams and Wainwright, 2012, 2013). For instance, various types of quantization schemes (Coughlan and Shen, 2007; Isard et al., 2009) have been used to reduce the effective state space and consequently the complexity. In another line of work, researchers have proposed stochastic methods inspired by particle filtering (Arulampalam et al., 2002; Sudderth et al., 2010; Isard, 2003; Doucet et al., 2001; Ihler and McAllester, 2009). These techniques are typically based on approximating the messages as weighted particles (Doucet et al., 2001; Ihler and McAllester, 2009), or mixture of Gaussians (Sudderth et al., 2010; Isard, 2003). Other researchers (Song et al., 2011) have proposed the use of kernel methods to simultaneously estimate parameters and compute approximate marginals in a simultaneous manner.

In this paper, we present a low-complexity (efficient) alternative to belief propagation with continuous variables. Our method, which we refer to as stochastic orthogonal series message-passing (SOSMP), is applicable to general pairwise Markov random fields, and is equipped with various theoretical guarantees. As suggested by its name, the algorithm is based on combining two ingredients: orthogonal series approximation of the messages, and the use of stochastic updates for efficiency. In this way, the SOSMP updates lead to a randomized algorithm with substantial reductions in communication and computational complexity. Our main contributions are to analyze the convergence properties of the SOSMP algorithm, and to provide rigorous bounds on the overall error as a function of the associated computational complexity. In particular, for tree-structured graphs, we establish almost sure convergence, and provide an explicit inverse polynomial convergence rate (Theorem 2). For loopy graphical models on which the usual BP updates are contractive, we also establish similar convergence rates (Theorem 3). Our general theory provides quantitative upper bounds on the number of iterations required to compute a δ-accurate approximation to the BP message fixed point, as we illustrate in the case of kernel-based potential functions (Theorem 4).

The reminder of the paper is organized as follows. We begin in Section 2, with the necessary background on the graphical models as well as the BP algorithm. Section 3 is devoted to a precise description of the SOSMP algorithm. In Section 4, we state our main theoretical results and develop some of their corollaries. In order to demonstrate the algorithm's effectiveness and confirm theoretical predictions, we provide some experimental results, on both synthetic and real data, in Section 5. In Section 6, we provide the proofs of our main results, with some of the technical aspects deferred to the appendices.

2. Background

We begin by providing some background on graphical models and the BP algorithm.

2.1 Undirected Graphical Models

Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consisting of a collection of nodes $\mathcal{V} = \{1, 2, \ldots, n\}$, along with a collection of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. An edge is an undirected pair (u, v), and self-edges are forbidden (meaning that $(u, u) \notin \mathcal{E}$ for all $u \in \mathcal{V}$). For each $u \in \mathcal{V}$, let X_u be a random variable taking values in a space $\mathcal{X}_u$. An undirected graphical model, also known as a Markov random field, defines a family of joint probability distributions over the random vector $X := \{X_u, u \in \mathcal{V}\}$, in which each distribution must factorize in terms of local potential functions associated with the cliques of the graph. In this paper, we focus on the case of pairwise Markov random fields, in which case the factorization is specified in terms of functions associated with the nodes and edges of the graph.

More precisely, we consider probability densities p that are absolutely continuous with respect to a given measure μ, typically the Lebesgue measure for the continuous random variables considered here. We say that p *respects the graph structure* if it can be factorized in the form

$$p(x_1, x_2, \ldots, x_n) \propto \prod_{u \in \mathcal{V}} \psi_u(x_u) \prod_{(u,v) \in \mathcal{E}} \psi_{uv}(x_u, x_v). \tag{1}$$

Here $\psi_u : \mathcal{X}_u \to (0, \infty)$ is the node potential function, whereas $\psi_{uv} : \mathcal{X}_u \times \mathcal{X}_v \to (0, \infty)$ denotes the edge potential function. A factorization of this form (1) is also known as *pairwise Markov random field*; see Figure 1 for a few examples that are widely used in practice.

In many applications, a central computational challenge is the computation of the marginal distribution

$$p(x_u) := \underbrace{\int_X \cdots \int_X}_{(n-1) \text{ times}} p(x_1, x_2, \ldots, x_n) \prod_{v \in \mathcal{V} \setminus \{u\}} \mu(dx_v) \tag{2}$$

at each node $u \in \mathcal{V}$. Naively approached, this problem suffers from the curse of dimensionality, since it requires computing a multi-dimensional integral over an $(n-1)$-dimensional space. For Markov random fields defined on trees (graphs without cycles), part of this exponential explosion can be circumvented by the use of the BP or sum-product algorithm, to which we turn in the following section.

Before proceeding, let us make a few comments about the relevance of the marginals in applied problems. In a typical application, one also makes independent noisy observations y_u of each hidden random variable X_u. By Bayes' rule, the posterior distribution of X given the observations $y = (y_1, \ldots, y_n)$ then takes the form

$$p_{X|Y}(x_1, \ldots, x_n \mid y_1, \ldots, y_n) \propto \prod_{u \in \mathcal{V}} \widetilde{\psi}_u(x_u; y_u) \prod_{(u,v) \in \mathcal{E}} \psi_{uv}(x_u, x_v), \tag{3}$$

where we have introduced the convenient shorthand for the modified node-wise potential functions $\widetilde{\psi}_u(x_u; y_u) := p(y_u \mid x_u) \psi_u(x_u)$. Since the observation vector y is fixed and known, any computational problem for the posterior distribution (3) can be reduced to an equivalent problem for a pairwise Markov random field of the form (1), using the given definition of the modified potential
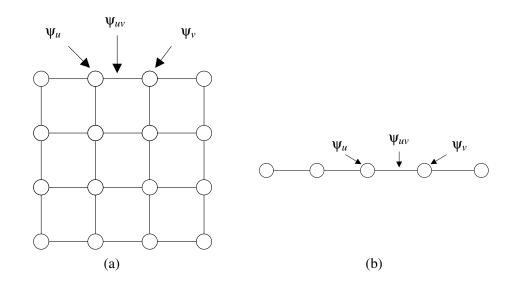
Figure 1: Examples of pairwise Markov random fields. (a) Two-dimensional grid. (b) Markov chain model. Potential functions ψ_u and ψ_v are associated with nodes u and v respectively, whereas potential function ψ_{uv} is associated with edge (u,v).

functions. In addition, although our theory allows for distinct state spaces X_u at each node $u \in \mathcal{V}$, throughout the remainder of the paper, we suppress this possible dependence so as to simplify exposition.

2.2 Belief Propagation

The BP algorithm, also known as the sum-product algorithm, is an iterative method based on message-passing updates for computing either exact or approximate marginal distributions. For trees (graphs without cycles), it is guaranteed to converge after a finite number of iterations and yields the exact marginal distributions, whereas for graphs with cycles, it yields only approximations to the marginal distributions. Nonetheless, this "loopy" form of BP is widely used in practice. Here we provide a very brief treatment sufficient for setting up the main results and analysis of this paper, referring the reader to various standard sources (Kschischang et al., 2001; Wainwright and Jordan, 2008) for further background.

In order to define the message-passing updates, we require some further notation. For each node $v \in \mathcal{V}$, let $\mathcal{N}(v) := \{u \in \mathcal{V} \mid (u,v) \in \mathcal{E}\}$ be its set of neighbors, and we use $\vec{\mathcal{E}}(v) := \{(v \to u) \mid u \in \mathcal{N}(v)\}$ to denote the set of all directed edges emanating from v. We use $\vec{\mathcal{E}} := \cup_{v \in \mathcal{V}} \vec{\mathcal{E}}(v)$ to denote the set of all directed edges in the graph. Let $\mathcal{M}$ denote the set of all probability densities (with respect to the base measure μ) defined on the space X—that is

$$\mathcal{M} = \left\{ m : X \to [0, \infty) \,\middle|\, \int_X m(x)\mu(dx) = 1 \right\}.$$

The messages passed by the BP algorithm are density functions, taking values in the space $\mathcal{M}$. More precisely, we assign one message $m_{v \to u} \in \mathcal{M}$ to every directed edge $(v \to u) \in \vec{\mathcal{E}}$, and we denote the

collection of all messages by $m = \{m_{v \to u}, (v \to u) \in \vec{\mathcal{E}}\}$. Note that the full collection of messages m takes values in the product space $\mathcal{M}^{|\vec{\mathcal{E}}|}$.

At an abstract level, the BP algorithm generates a sequence of message densities $\{m^t\}$ in the space $\mathcal{M}^{|\vec{\mathcal{E}}|}$, where $t = 0, 1, 2 \ldots$ is the iteration number. The update of message m^t to message m^{t+1} can be written in the form $m^{t+1} = \mathcal{F}(m^t)$, where $\mathcal{F} : \mathcal{M}^{|\vec{\mathcal{E}}|} \to \mathcal{M}^{|\vec{\mathcal{E}}|}$ is a non-linear operator. This global operator is defined by the local update operators[1] $\mathcal{F}_{v \to u} : \mathcal{M}^{|\vec{\mathcal{E}}|} \to \mathcal{M}$, one for each directed edge of the graph, such that $m_{v \to u}^{t+1} = \mathcal{F}_{v \to u}(m^t)$.

More precisely, in terms of these local updates, the BP algorithm operates as follows. At each iteration $t = 0, 1, \ldots$, each node $v \in \mathcal{V}$ performs the following steps:

- for each one of its neighbors $u \in \mathcal{N}(v)$, it computes $m_{v \to u}^{t+1} = \mathcal{F}_{v \to u}(m^t)$.

- it transmits message $m_{v \to u}^{t+1}$ to neighbor $u \in \mathcal{N}(v)$.

In more detail, the message update takes the form

$$\underbrace{[\mathcal{F}_{v \to u}(m^t)](\cdot)}_{m_{v \to u}^{t+1}(\cdot)} := \kappa \int_{\mathcal{X}} \left\{ \psi_{uv}(\cdot, x_v) \, \psi_v(x_v) \prod_{w \in \mathcal{N}(v) \setminus \{u\}} m_{w \to v}^t(x_v) \right\} \mu(dx_v), \qquad (4)$$

where κ is a normalization constant chosen to enforce the normalization condition

$$\int_{\mathcal{X}} m_{v \to u}^{t+1}(x_u) \, \mu(dx_u) = 1.$$

By concatenating the local updates (4), we obtain a global update operator $\mathcal{F} : \mathcal{M}^{|\vec{\mathcal{E}}|} \to \mathcal{M}^{|\vec{\mathcal{E}}|}$, as previously discussed. The goal of belief propagation message-passing is to obtain a *fixed point*, meaning an element $m^* \in \mathcal{M}^{|\vec{\mathcal{E}}|}$ such that $\mathcal{F}(m^*) = m^*$. Under mild conditions, it can be shown that there always exists at least one fixed point, and for any tree-structured graph, the fixed point is unique.

Given a fixed point m^*, each node $u \in \mathcal{V}$ computes its marginal approximation $\tau_u^* \in \mathcal{M}$ by combining the local potential function ψ_u with a product of all incoming messages as

$$\tau_u^*(x_u) \propto \psi_u(x_u) \prod_{v \in \mathcal{N}(u)} m_{v \to u}^*(x_u). \qquad (5)$$

Figure 2 provides a graphical representation of the flow of the information in these local updates. For tree-structured (cycle-free) graphs, it is known that BP updates (4) converge to the unique fixed point in a finite number of iterations (Wainwright and Jordan, 2008). Moreover, the quantity $\tau_u^*(x_u)$ is equal to the single-node marginal, as previously defined (2). For general graphs, uniqueness of the fixed point is no longer guaranteed (Wainwright and Jordan, 2008); however, the same message-passing updates can be applied, and are known to be extremely effective for computing approximate marginals in numerous applications.

---

1. It is worth mentioning, and important for the computational efficiency of BP, that $m_{v \to u}$ is only a function of the messages $m_{w \to v}$ for $w \in \mathcal{N}(v) \setminus \{u\}$. Therefore, we have $\mathcal{F}_{v \to u} : \mathcal{M}^{d_v - 1} \to \mathcal{M}$, where d_v is the degree of the node v. However, we suppress this local dependence so as to reduce notational clutter.
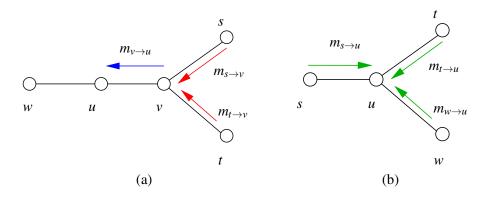
Figure 2: Graphical representation of message-passing algorithms. (a) Node v transmits the message $m_{v \to u} = \mathcal{F}_{v \to u}(m)$, derived from Equation (4), to its neighbor u. (b) Upon receiving all the messages, node u updates its marginal estimate according to (5).

Although the BP algorithm is considerably more efficient than the brute force approach to marginalization, the message update Equation (4) still involves computing an integral and transmitting a real-valued function (message). With certain exceptions (such as multivariate Gaussians), these continuous-valued messages do not have finite representations, so that this approach is computationally very expensive. Although integrals can be computed by numerical methods, the BP algorithm requires performing many such integrals *at each iteration*, which becomes very expensive in practice.

3. Description of the Algorithm

We now turn to the description of the SOSMP algorithm. Before doing so, we begin with some background on the main underlying ingredients: orthogonal series expansion, and stochastic message updates.

3.1 Orthogonal Series Expansion

As described in the previous section, for continuous random variables, each message is a density function in the space $\mathcal{M} \subset L^2(X;\mu)$. We measure distances in this space using the usual L^2 norm $\|f - g\|_2^2 := \int_X (f(x) - g(x))^2 \mu(dx)$. A standard way in which to approximate functions is via orthogonal series expansion. In particular, let $\{\phi_j\}_{j=1}^{\infty}$ be an orthonormal basis of $L^2(X;\mu)$, meaning a collection of functions such that

$$\underbrace{\int_X \phi_i(x)\phi_j(x)\mu(dx)}_{:=\langle \phi_i, \phi_j \rangle_{L^2}} = \begin{cases} 1 & \text{when } i = j \\ 0 & \text{otherwise.} \end{cases}$$

Any function $f \in \mathcal{M} \subset L^2(X;\mu)$ then has an expansion of the form $f = \sum_{j=1}^{\infty} a_j \phi_j$, where $a_j = \langle f, \phi_j \rangle_{L^2}$ are the basis expansion coefficients.

Of course, maintaining the infinite sequence of basis coefficients $\{a_j\}_{j=1}^{\infty}$ is also computationally intractable, so that any practical algorithm will maintain only a finite number r of basis coefficients. For a given r, we let $\widehat{f}_r \propto \left[\sum_{j=1}^{r} a_j \phi_j\right]_+$ be the approximation based on the first r coefficients. (Applying the operator $[t]_+ = \max\{0,t\}$ amounts to projecting $\sum_{j=1}^{r} a_j \phi_j$ onto the space of non-negative functions, and we also normalize to ensure that it is a density function.) In using only r coefficients, we incur the *approximation error*

$$\|\widehat{f}_r - f\|_2^2 \overset{(i)}{\leq} \|\sum_{j=1}^{r} a_j \phi_j - f\|_2^2 \overset{(ii)}{=} \sum_{j=r+1}^{\infty} a_j^2, \tag{6}$$

where inequality (i) uses non-expansivity of the projection, and step (ii) follows from Parseval's theorem. Consequently, the approximation error will depend both on

- how many coefficients r that we retain, and

- the decay rate of the expansion coefficients $\{a_j\}_{j=1}^{\infty}$.

For future reference, it is worth noting that the local message update (4) is defined in terms of an integral operator of the form

$$f(\cdot) \mapsto \int_{\mathcal{X}} \psi_{uv}(\cdot,y) f(y) \mu(dy). \tag{7}$$

Consequently, whenever the edge potential function ψ_{uv} has desirable properties—such as differentiability and/or higher order smoothness—then the messages also inherit these properties. With an appropriate choice of the basis $\{\phi_j\}_{j=1}^{\infty}$, such properties translate into decay conditions on the basis coefficients $\{a_j\}_{j=1}^{\infty}$. For instance, for α-times differentiable functions expanded into the Fourier basis, the Riemann-Lebesgue lemma guarantees that the coefficients a_j decay faster than $(1/j)^{2\alpha}$. We develop these ideas at greater length in the sequel.

3.2 Stochastic Message Updates

In order to reduce the approximation error (6), the number of coefficients r needs to be increased (as a function of the ultimate desired error δ). Since increases in r lead to increases in computational complexity, we need to develop effective reduced-complexity methods. In this section, we describe (at a high-level) how this can be done via a stochastic version of the BP message-passing updates.

We begin by observing that message update (4), following the appropriate normalization, can be cast as an expectation operation. This equivalence is essential, because it allows us to obtain unbiased approximations of the message update using stochastic techniques. In particular, for every directed edge $(v \to u)$ let us define

$$\Gamma_{uv}(\cdot,y) := \frac{\psi_{uv}(\cdot,y)}{\int_{\mathcal{X}} \psi_{uv}(x,y)\,\mu(dx)}, \quad \text{and} \quad \beta_{v \to u}(y) := \psi_v(y) \int_{\mathcal{X}} \psi_{uv}(x,y)\,\mu(dx), \tag{8}$$

the normalized compatibility function and the marginal potential weight respectively. By construction, for each y, we have $\int_{\mathcal{X}} \Gamma_{uv}(x,y)\mu(dx) = 1$.

Lemma 1 *Given an input collection of messages m, let Y be a random variable with density proportional to*

$$[p_{v \to u}(m)](y) \propto \beta_{v \to u}(y) \prod_{w \in \mathcal{N}(v) \setminus \{u\}} m_{w \to v}(y). \tag{9}$$

Then the message update Equation (4) *can be written as*

$$[\mathcal{F}_{v \to u}(m)](\cdot) = \mathbb{E}_Y \left[\Gamma_{uv}(\cdot, Y) \right].$$

Proof Let us introduce the convenient shorthand $M(y) = \prod_{w \in \mathcal{N}(v) \setminus \{u\}} m_{w \to v}(y)$. By definition (4) of the message update, we have

$$[\mathcal{F}_{v \to u}(m)](\cdot) = \frac{\int_X \left(\psi_{uv}(\cdot, y) \, \psi_v(y) \, M(y) \mu(dy) \right)}{\int_X \int_X \left(\psi_{uv}(x, y) \, \psi_v(y) M(y) \right) \mu(dy) \, \mu(dx)}.$$

Since the integrand is positive, by Fubini's theorem (Durrett, 1995), we can exchange the order of integrals in the denominator. Doing so and simplifying the expression yields

$$[\mathcal{F}_{v \to u}(m)](\cdot) = \int_X \underbrace{\frac{\psi_{uv}(\cdot, y)}{\int_X \psi_{uv}(x, y) \mu(dx)}}_{\Gamma_{uv}(\cdot, y)} \underbrace{\frac{\beta_{v \to u}(y) M(y)}{\int_X \beta_{v \to u}(z) M(z) \mu(dz)}}_{[p_{v \to u}(m)](y)} \mu(dy),$$

which establishes the claim. ∎

Based on Lemma 1, we can obtain a stochastic approximation to the message update by drawing k i.i.d. samples Y_i from the density (9), and then computing $\sum_{i=1}^k \Gamma_{uv}(\cdot, Y_i) / k$. Given the non-negativity and chosen normalization of Γ_{uv}, note that this estimate belongs to $\mathcal{M}$ by construction. Moreover, it is an unbiased estimate of the correctly updated message, which plays an important role in our analysis.

3.3 Precise Description of the Algorithm

The SOSMP algorithm involves a combination of the orthogonal series expansion techniques and stochastic methods previously described. Any particular version of the algorithm is specified by the choice of basis $\{\phi_j\}_{j=1}^\infty$ and two positive integers: the number of coefficients r that are maintained, and the number of samples k used in the stochastic update. Prior to running the algorithm, for each directed edge $(v \to u)$, we pre-compute the inner products

$$\gamma_{v \to u; j}(y) := \underbrace{\int_X \Gamma_{uv}(x, y) \phi_j(x) \mu(dx)}_{\langle \Gamma_{uv}(\cdot, y), \phi_j(\cdot) \rangle_{L^2}}, \qquad \text{for } j = 1, \dots, r. \tag{10}$$

When ψ_{uv} is a symmetric and positive semidefinite kernel function, these inner products have an explicit and simple representation in terms of its Mercer eigendecomposition (see Section 4.3). In the general setting, these r inner products can be computed via standard numerical integration techniques. Note that this is a fixed (one-time) cost prior to running the algorithm.

SOSMP algorithm:

1. At time $t = 0$, initialize the message coefficients

$$a^0_{v \to u; j} = 1/r \quad \text{for all } j = 1, \ldots, r, \text{ and } (v \to u) \in \vec{\mathcal{E}}.$$

2. For iterations $t = 0, 1, 2, \ldots$, and for each directed edge $(v \to u)$

   (a) Form the projected message approximation $\widehat{m}^t_{w \to v}(\cdot) = \left[\sum_{j=1}^{r} a^t_{w \to v; j} \phi_j(\cdot)\right]_+$, for all $w \in \mathcal{N}(v) \setminus \{u\}$.

   (b) Draw k i.i.d. samples Y_i^{t+1} from the probability density proportional to

   $$\beta_{v \to u}(y) \prod_{w \in \mathcal{N}(v) \setminus \{u\}} \widehat{m}^t_{w \to v}(y),$$

   where $\beta_{v \to u}$ was previously defined in Equation (8).

   (c) Use the samples $\{Y_1^{t+1}, \ldots, Y_k^{t+1}\}$ from step (b) to compute

   $$\widetilde{b}^{t+1}_{v \to u; j} := \frac{1}{k} \sum_{i=1}^{k} \gamma_{v \to u; j}(Y_i^{t+1}) \qquad \text{for } j = 1, 2, \ldots, r, \tag{12}$$

   where the function $\gamma_{v \to u; j}$ is defined in Equation (10).

   (d) For step size $\eta^t = 1/(t+1)$, update the r-dimensional message coefficient vectors $a^t_{v \to u} \mapsto a^{t+1}_{v \to u}$ via

   $$a^{t+1}_{v \to u} = (1 - \eta^t) a^t_{v \to u} + \eta^t \widetilde{b}^{t+1}_{v \to u}. \tag{13}$$

Figure 3: The *SOSMP* algorithm for continuous state space marginalization.

At each iteration $t = 0, 1, 2, \ldots$, the algorithm maintains an r-dimensional vector of basis expansion coefficients

$$a^t_{v \to u} = (a^t_{v \to u; 1}, \ldots, a^t_{v \to u; r}) \in \mathbb{R}^r, \quad \text{on directed edge } (v \to u) \in \vec{\mathcal{E}}.$$

This vector should be understood as defining the current message approximation $m^t_{v \to u}$ on edge $(v \to u)$ via the expansion

$$m^t_{v \to u}(\cdot) := \sum_{j=1}^{r} a^t_{v \to u; j} \, \phi_j(\cdot). \tag{11}$$

We use $a^t = \{a^t_{v \to u}, (v \to u) \in \vec{\mathcal{E}}\}$ to denote the full set of $r|\vec{\mathcal{E}}|$ coefficients that are maintained by the algorithm at iteration t. With this notation, the algorithm consists of a sequence of steps, detailed in Figure 3, that perform the update $a^t \mapsto a^{t+1}$, and hence implicitly the update $m^t \mapsto m^{t+1}$.

3.4 Computational Complexity

The sampling in Step 2(b) of the algorithm may be performed using standard sampling methods (Ripley, 1987). For instance, accept-reject sampling involves drawing from an auxiliary distribution, and then performing an accept-reject step. Let c denote the average number of times that an auxiliary sample must be drawn until acceptance. From Figure 3, it can be seen that each iteration requires $O(crk)$ floating point operations on average per edge. Note that the accept-reject factor c depends only on the state space X and the auxiliary distribution, and it is independent of r, k, and the graph structure.

It is also interesting to compare the computational complexity of our algorithm to that of competing methods. For instance, the non-parametric (Gaussian mixture) BP algorithm, as proposed in past works (Sudderth et al., 2010; Isard, 2003), involves approximating each message with an ℓ-component mixture of Gaussians. The associated computational complexity is $O(\ell^{d_{\max}})$ per iteration per edge, where $d_{\max}$ is the maximum degree of the graph, so that the method is suitable for graphs of relatively low degree. Ihler and McAllester (2009) suggested an improvement known as particle BP, in which messages are approximated by particles associated with the nodes (as opposed to the edges). Given an approximation based on ℓ particles at each node, the computational complexity of particle BP is $O(\ell^2)$ per iteration per edge. More recently, Song et al. (2011) proposed a method known as kernel BP, designed for jointly estimating the potential functions and computing (approximate) marginal distributions. Their approach is based on representing the potentials as weighted sums of kernels in a reproducing kernel Hilbert space, which leads to simple BP updates. For a training set with M samples, the computational complexity of the kernel BP is $O(M^2)$ per iteration per edge. They also proposed a more efficient variant of the kernel BP by approximating the feature matrices by weighted combination of subsets of their columns.

4. Theoretical Guarantees

We now turn to the theoretical analysis of the SOSMP algorithm, and guarantees relative to the fixed points of the true BP algorithm. For any tree-structured graph, the BP algorithm is guaranteed to have a unique message fixed point $m^* = \{m^*_{v \to u}, (v \to u) \in \vec{E}\}$. For graphs with cycles, uniqueness is no longer guaranteed, which would make it difficult to compare with the SOSMP algorithm. Accordingly, in our analysis of the loopy graph, we make a natural contractivity assumption, which guarantees uniqueness of the fixed point m^*.

The SOSMP algorithm generates a random sequence $\{a^t\}_{t=0}^{\infty}$, which define message approximations $\{m^t\}_{t=0}^{\infty}$ via the expansion (11). Of interest to us are the following questions:

- under what conditions do the message iterates approach a neighborhood of the BP fixed point m^* as $t \to +\infty$?

- when such convergence takes place, how fast is it?

In order to address these questions, we separate the error in our analysis into two terms: algorithmic error and approximation error. For a given r, let Π^r denote the projection operator onto the span of $\{\phi_1, \ldots, \phi_r\}$. In detail, given a function f represented in terms of the infinite series

expansion $f = \sum_{j=1}^{\infty} a_j \phi_j$, we have

$$\Pi^r(f) := \sum_{j=1}^{r} a_j \phi_j.$$

For each directed edge $(v \to u) \in \vec{\mathcal{E}}$, define the functional error

$$\Delta_{v \to u}^t := m_{v \to u}^t - \Pi^r(m_{v \to u}^*) \tag{14}$$

between the message approximation at time t, and the BP fixed point projected onto the first r basis functions. Moreover, define the approximation error at the BP fixed point as

$$A_{v \to u}^r := m_{v \to u}^* - \Pi^r(m_{v \to u}^*). \tag{15}$$

Since $\Delta_{v \to u}^t$ belongs to the span of the first r basis functions, the Pythagorean theorem implies that the overall error can be decomposed as

$$\|m_{v \to u}^t - m_{v \to u}^*\|_{L^2}^2 \quad = \quad \underbrace{\|\Delta_{v \to u}^t\|_{L^2}^2}_{\text{Estimation error}} \quad + \quad \underbrace{\|A_{v \to u}^r\|_{L^2}^2}_{\text{Approximation error}}.$$

Note that the approximation error term is independent of the iteration number t, and can only be reduced by increasing the number r of coefficients used in the series expansion. Our analysis of the estimation error is based on controlling the $|\vec{\mathcal{E}}|$-dimensional error vector

$$\rho^2(\Delta^t) := \left\{ \|\Delta_{v \to u}^t\|_{L^2}^2, (v \to u) \in \vec{\mathcal{E}} \right\} \in \mathbb{R}^{|\vec{\mathcal{E}}|}, \tag{16}$$

and in particular showing that it decreases as $O(1/t)$ up to an error floor imposed by the approximation error. In order to analyze the error, we also introduce the $|\vec{\mathcal{E}}|$-dimensional vector of approximation errors

$$\rho^2(A^r) := \left\{ \|A_{v \to u}^r\|_{L^2}^2, (v \to u) \in \vec{\mathcal{E}} \right\} \in \mathbb{R}^{|\vec{\mathcal{E}}|}. \tag{17}$$

By increasing r, we can reduce this approximation error term, but at the same time, we increase the computational complexity of each update. In Section 4.3, we discuss how to choose r so as to trade-off the estimation and approximation errors with computational complexity.

4.1 Bounds for Tree-Structured Graphs

With this set-up, we now turn to bounds for tree-structured graphs. Our analysis of the tree-structured case controls the vector of estimation errors $\rho^2(\Delta^t)$ using a nilpotent matrix $N \in \mathbb{R}^{|\vec{\mathcal{E}}| \times |\vec{\mathcal{E}}|}$ determined by the tree structure (Noorshams and Wainwright, 2013). Recall that a matrix N is nilpotent with order ℓ if $N^\ell = 0$ and $N^{\ell-1} \neq 0$ for some ℓ. As illustrated in Figure 4, the rows and columns of N are indexed by directed edges. For the row indexed by $(v \to u)$, there can be non-zero entries only for edges in the set $\{(w \to v), w \in \mathcal{N}(v) \backslash \{u\}\}$. These directed edges are precisely those that pass messages relevant in updating the message from v to u, so that N tracks the propagation of message information in the graph. As shown in our previous work (see Lemma 1 in the paper by Noorshams and Wainwright, 2013), the matrix N with such structure is nilpotent with degree at
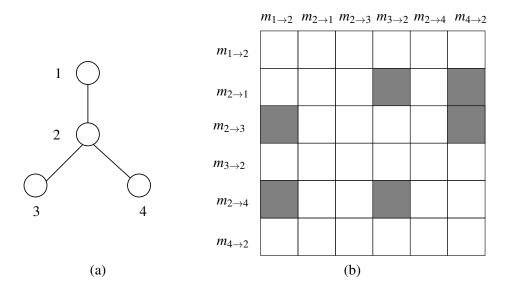
Figure 4: (a) A simple tree with $|\mathcal{E}| = 3$ edges and hence $|\vec{\mathcal{E}}| = 6$ directed edges. (b) Structure of nilpotent matrix $N \in \mathbb{R}^{|\vec{\mathcal{E}}| \times |\vec{\mathcal{E}}|}$ defined by the graph in (a). Rows and columns of the matrix are indexed by directed edges $(v \to u) \in \vec{\mathcal{E}}$; for the row indexed by $(v \to u)$, there can be non-zero entries only for edges in the set $\{(w \to v), w \in \mathcal{N}(v) \setminus \{u\}\}$.

most the diameter of the tree. (In a tree, there is always a unique edge-disjoint path between any pair of nodes; the diameter of the tree is the length of the longest of these paths.)

Moreover, our results on tree-structured graphs impose one condition on the vector of approximation errors A^r, namely that

$$\inf_{y \in X} \Pi^r\left(\Gamma_{uv}(x,y)\right) > 0, \quad \text{and} \quad |A^r_{v \to u}(x)| \leq \frac{1}{2} \inf_{y \in X} \Pi^r\left(\Gamma_{uv}(x,y)\right) \tag{18}$$

for all $x \in X$ and all directed edges $(v \to u) \in \vec{\mathcal{E}}$. This condition ensures that the L^2-norm of the approximation error is not too large relative to the compatibility functions. Since $\sup_{x,y \in X} |\Pi^r\left(\Gamma_{uv}(x,y)\right) - \Gamma_{uv}(x,y)| \to 0$ and $\sup_{x \in X} |A^r_{v \to u}(x)| \to 0$ as $r \to +\infty$, assuming that the compatibility functions are uniformly bounded away from zero, condition (18) will hold once the number of basis expansion coefficients r is sufficiently large. Finally, our bounds involve the constants

$$B_j := \max_{(v \to u) \in \vec{\mathcal{E}}} \sup_{y \in X} \langle \Gamma_{uv}(\cdot, y), \phi_j \rangle_{L^2}. \tag{19}$$

With this set-up, we have the following guarantees:

Theorem 2 (tree-structured graphs) *Suppose that X is closed and bounded, the node and edge potential functions are continuous, and that condition* (18) *holds. Then for any tree-structured model, the sequence of messages $\{m^t\}_{t=0}^{\infty}$ generated by the SOSMP algorithm have the following properties:*

(a) *There is a nilpotent matrix $N \in \mathbb{R}^{|\vec{E}| \times |\vec{E}|}$ such that the error vector $\rho^2(\Delta^t)$ converges almost surely to the set*

$$\mathcal{B} := \left\{ e \in \mathbb{R}^{|\vec{E}|} \mid |e| \preceq N(I-N)^{-1}\rho^2(A^r) \right\},$$

where $\preceq$ denotes elementwise inequality for vectors.

(b) *Furthermore, for all iterations $t = 1, 2, \ldots$, we have*

$$\mathbb{E}\left[\rho^2(\Delta^t)\right] \preceq \left(12 \sum_{j=1}^{r} B_j^2\right) \frac{(I - \log t \, N)^{-1}}{t} (N\vec{1} + 8) + N(I-N)^{-1}\rho^2(A^r).$$

To clarify the statement in part (a), it guarantees that the difference $\rho^2(\Delta^t) - \Pi_{\mathcal{B}}(\rho^2(\Delta^t))$ between the error vector and its projection onto the set $\mathcal{B}$ converges almost surely to zero. Part (b) provides a quantitative guarantee on how quickly the expected absolute value of this difference converges to zero. In particular, apart from logarithmic factors in t, the convergence rate guarantees is of the order $O(1/t)$.

4.2 Bounds for General Graphs

Our next theorem addresses the case of general graphical models. The behavior of the ordinary BP algorithm to a graph with cycles—in contrast to the tree-structured case—is more complicated. On one hand, for strictly positive potential functions (as considered in this paper), a version of Brouwer's fixed point theorem can be used to establish existence of fixed points (Wainwright and Jordan, 2008). However, in general, there may be multiple fixed points, and convergence is not guaranteed. Accordingly, various researchers have studied conditions that are sufficient to guarantee uniqueness of fixed points and/or convergence of the ordinary BP algorithm: one set of sufficient conditions, for both uniqueness and convergence, involve assuming that the BP update operator is a contraction in a suitable norm (e.g., Tatikonda and Jordan, 2002; Ihler et al., 2005; Mooij and Kappen, 2007; Roosta et al., 2008).

In our analysis of the SOSMP algorithm for a general graph, we impose the following form of *contractivity*: there exists a constant $0 < \gamma < 2$ such that

$$\|\mathcal{F}_{v \to u}(m) - \mathcal{F}_{v \to u}(m')\|_{L^2} \leq \left(1 - \frac{\gamma}{2}\right) \sqrt{\frac{1}{|\mathcal{N}(v)\backslash\{u\}|} \sum_{w \in \mathcal{N}(v)\backslash\{u\}} \|m_{w \to v} - m'_{w \to v}\|_{L^2}^2}, \qquad (20)$$

for all directed edges $(v \to u) \in \vec{E}$, and feasible messages m, and m'. We say that the ordinary BP algorithm is γ-contractive when condition (20) holds.

Theorem 3 (general graphs) *Suppose that the ordinary BP algorithm is γ-contractive (20), and consider the sequence of messages $\{m^t\}_{t=0}^{\infty}$ generated with step-size $\eta^t = 1/(\gamma(t+1))$. Then for all $t = 1, 2, \ldots$, the error sequence $\{\Delta_{v \to u}^t\}_{t=0}^{\infty}$ is bounded in mean-square as*

$$\mathbb{E}\left[\rho^2(\Delta^t)\right] \preceq \left[\left(\frac{8\sum_{j=1}^{r} B_j^2}{\gamma^2}\right) \frac{\log t}{t} + \frac{1}{\gamma} \max_{(v \to u) \in \vec{E}} \|A_{v \to u}^r\|_{L^2}^2\right] \vec{1}. \qquad (21)$$

where $A_{v \to u}^r = m_{v \to u}^* - \Pi^r(m_{v \to u}^*)$ is the approximation error on edge $(v \to u)$.

Theorem 3 guarantees that under the contractivity condition (20), the SOSMP iterates converge to a neighborhood of the BP fixed point. The error offset depends on the approximation error term that decays to zero as r is increased. Moreover, disregarding the logarithmic factor, the convergence rate is $O(1/t)$, which is the best possible for a stochastic approximation scheme of this type (Nemirovsky and Yudin, 1983; Agarwal et al., 2012).

4.3 Explicit Rates for Kernel Classes

Theorems 2 and 3 are generic results that apply to any choices of the edge potential functions. In this section, we pursue a more refined analysis of the number of arithmetic operations that are required to compute a δ-*uniformly accurate* approximation to the BP fixed point m^*, where $\delta > 0$ is a user-specified tolerance parameter. By a δ-uniformly accurate approximation, we mean a collection of messages m such that

$$\max_{(v \to u) \in \vec{\mathcal{E}}} \mathbb{E}\left[\|m_{v \to u} - m^*_{v \to u}\|^2_{L^2}\right] \leq \delta.$$

In order to obtain such an approximation, we need to specify both the number of coefficients r to be retained, and the number of iterations that we should perform. Based on these quantities, our goal is to specify the *minimal number of basic arithmetic operations* $T(\delta)$ that are sufficient to compute a δ-accurate message approximation.

In order to obtain concrete answers, we study this issue in the context of kernel-based potential functions. In many applications, the edge potentials $\psi_{uv} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ are symmetric and positive semidefinite (PSD) functions, frequently referred to as kernel functions.[2] Commonly used examples include the Gaussian kernel $\psi_{uv}(x,y) = \exp(-\gamma\|x-y\|^2_2)$, the closely related Laplacian kernel, and other types of kernels that enforce smoothness priors. Any kernel function defines a positive semidefinite integral operator, namely via Equation (7). When $\mathcal{X}$ is compact and the kernel function is continuous, then Mercer's theorem (Riesz and Nagy, 1990) guarantees that this integral operator has a countable set of eigenfunctions $\{\phi_j\}_{j=1}^{\infty}$ that form an orthonormal basis of $L^2(\mathcal{X};\mu)$. Moreover, the kernel function has the expansion

$$\psi_{uv}(x,y) = \sum_{j=1}^{\infty} \lambda_j \, \phi_j(x) \, \phi_j(y),$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ are the eigenvalues, all guaranteed to be non-negative. In general, the eigenvalues might differ from edge to edge, but we suppress this dependence for simplicity in exposition. We study kernels that are trace class, meaning that the eigenvalues are absolutely summable (i.e., $\sum_{j=1}^{\infty} \lambda_j < \infty$).

For a given eigenvalue sequence $\{\lambda_j\}_{j=1}^{\infty}$ and some tolerance $\delta > 0$, we define the *critical dimension* $r^* = r^*(\delta; \{\lambda_j\})$ to be the smallest positive integer r such that

$$\lambda_r \leq \delta. \tag{22}$$

Since $\lambda_j \to 0$, the existence of $r^* < \infty$ is guaranteed for any $\delta > 0$.

---

2. In detail, a PSD kernel function has the property that for all natural numbers n and $\{x_1, \ldots, x_n\} \subset \mathcal{X}$, the $n \times n$ kernel matrix with entries $\psi_{uv}(x_i, x_j)$ is symmetric and positive semidefinite.

Theorem 4 (complexity vs. accuracy) *In addition to the conditions of Theorem 3, suppose that the compatibility functions are defined by a symmetric PSD trace-class kernel with eigenvalues $\{\lambda_j\}$. If we run the SOSMP algorithm with $r^* = r^*(\delta; \{\lambda_j\})$ basis coefficients, and $k = O(1)$ samples, then it suffices to perform*

$$T(\delta; \{\lambda_j\}) = O\left(r^* \left(\sum_{j=1}^{r^*} \lambda_j^2\right) (1/\delta) \log(1/\delta)\right) \tag{23}$$

arithmetic operations per edge in order to obtain a δ-accurate message vector m.

The proof of Theorem 4 is provided in Section 6.3. It is based on showing that the choice (22) suffices to reduce the approximation error to $O(\delta)$, and then bounding the total operation complexity required to also reduce the estimation error.

Theorem 4 can be used to derive explicit estimates of the complexity for various types of kernel classes. We begin with the case of kernels in which the eigenvalues decay at an inverse polynomial rate: in particular, given some $\alpha > 1$, we say that they exhibit α-*polynomial decay* if there is a universal constant C such that

$$\lambda_j \leq C/j^\alpha \quad \text{for all } j = 1, 2, \ldots. \tag{24}$$

Examples of kernels in this class include Sobolov spline kernels (Gu, 2002), which are a widely used type of smoothness prior. For example, the spline class associated with functions that are s-times differentiable satisfies the decay condition (24) with $\alpha = 2s$.

Corollary 5 *In addition to the conditions of Theorem 3, suppose that the compatibility functions are symmetric kernels with α-polynomial decay* (24). *Then it suffices to perform*

$$T_{\text{poly}}(\delta) = O\left((1/\delta)^{\frac{1+\alpha}{\alpha}} \log(1/\delta)\right)$$

operations per edge in order to obtain a δ-accurate message vector m.

The proof of this corollary is immediate from Theorem 4: given the assumption $\lambda_j \leq C/j^\alpha$, we see that $r^* \leq (C/\delta)^{\frac{1}{\alpha}}$ and $\sum_{j=1}^{r^*} \lambda_j^2 = O(1)$. Substituting into the bound (23) yields the claim. Corollary 5 confirms a natural intuition—namely, that it should be easier to compute an approximate BP fixed point for a graphical model with smooth potential functions. Disregarding the logarithmic factor (which is of lower-order), the operation complexity $T_{\text{poly}}(\delta)$ ranges from $O((1/\delta)^2)$, obtained as $\alpha \to 1^+$ all the way down to $O(1/\delta)$, obtained as $\alpha \to +\infty$.

Another class of widely used kernels are those with exponentially decaying eigenvalues: in particular, for some $\alpha > 0$, we say that the kernel has α-*exponential decay* if there are universal constants (C, c) such that

$$\lambda_j \leq C \exp(-cj^\alpha) \qquad \text{for all } j = 1, 2, \ldots. \tag{25}$$

Examples of such kernels include the Gaussian kernel, which satisfies the decay condition (25) with $\alpha = 2$ (e.g., Steinwart and Christmann, 2008).
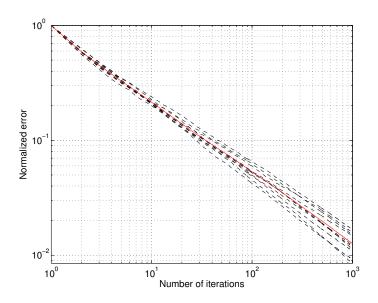
Figure 5: Plot of normalized error e^t/e^0 vs. the number of iterations t for 10 different sample paths on a chain of size $n = 100$. The dashed lines are sample paths whereas the solid line is the mean square error. In this experiment node and edge potentials are mixtures of three Gaussians and we implemented SOSMP using the first $r = 10$ Fourier coefficients with $k = 5$ samples.

Corollary 6 *In addition to the conditions of Theorem 3, suppose that the compatibility functions are symmetric kernels with* α*-exponential decay* (25). *Then it suffices to perform*

$$T_{\exp}(\delta) = O\Big((1/\delta)\left(\log(1/\delta)\right)^{\frac{1+\alpha}{\alpha}}\Big) \tag{26}$$

operations per edge in order to obtain a uniformly δ*-accurate message vector m.*

As with our earlier corollary, the proof of this claim is a straightforward consequence of Theorem 4. Corollary 6 demonstrates that kernel classes with exponentially decaying eigenvalues are not significantly different from parametric function classes, for which a stochastic algorithm would have operation complexity $O(1/\delta)$. Apart from the lower order logarithmic terms, the complexity bound (26) matches this parametric rate.

5. Experimental Results

In this section, we describe some experimental results that help to illustrate the theoretical predictions of the previous section.

5.1 Synthetic Data

We begin by running some experiments for a simple model, in which both the node and edge potentials are mixtures of Gaussians. More specifically, we form a graphical model with potential

functions of the form

$$\psi_u(x_u) = \sum_{i=1}^{3} \pi_{u;i} \exp\left(-(x_u - \mu_{u;i})^2/(2\sigma_{u;i}^2)\right), \quad \text{for all } u \in \mathcal{V}, \text{ and} \tag{27}$$

$$\psi_{uv}(x_u, x_v) = \sum_{i=1}^{3} \pi_{uv;i} \exp\left(-(x_v - x_u)^2/(2\sigma_{uv;i}^2)\right) \quad \text{for all } (u,v) \in \mathcal{E}, \tag{28}$$

where the non-negative mixture weights are normalized (i.e., $\sum_{i=1}^{3} \pi_{uv;i} = \sum_{i=1}^{3} \pi_{u;i} = 1$). For each vertex and edge and for all $i = 1, 2, 3$, the mixture parameters are chosen randomly from uniform distributions over the range $\sigma_{u;i}^2, \sigma_{uv;i}^2 \in (0, 0.5]$ and $\mu_{u;i} \in [-3, 3]$.

5.1.1 SAMPLE PATHS ON A TREE-STRUCTURED GRAPH

For a chain-structured graph with $n = 100$ nodes, we first compute the fixed point of the standard BP, using direct numerical integration to compute the integrals,[3] so to compute (a very accurate approximation of) the fixed point m^*. We compare this "exact" answer to the approximation obtained by running the SOSMP algorithm using the first $r = 10$ Fourier basis coefficients and $k = 5$ samples. Having run the SOSMP algorithm, we compute the average squared error

$$e^t := \frac{1}{|\vec{\mathcal{E}}|} \sum_{(v \to u) \in \vec{\mathcal{E}}} \sum_{j=1}^{r} (a_{v \to u;j}^t - a_{v \to u;j}^*)^2 \tag{29}$$

at each time $t = 1, 2, \ldots$.

Figure 5 provides plots of the error (29) versus the number of iterations for 10 different trials of the SOSMP algorithm. (Since the algorithm is randomized, each path is slightly different.) The plots support our claim of of almost sure convergence, and moreover, the straight lines seen in the log-log plots confirm that convergence takes place at a rate inverse polynomial in t.

5.1.2 EFFECT OF THE NUMBER OF COEFFICIENTS r AND THE NUMBER OF SAMPLES k

In the next few simulations, we test the algorithm's behavior with respect to the number of expansion coefficients r, and number of samples k. In particular, Figure 6(a) illustrates the expected error, averaged over several sample paths, vs. the number of iterations for different number of expansion coefficients $r \in \{2, 3, 5, 10\}$ when $k = 5$ fixed; whereas Figure 6(b) depicts the expected error vs. the number of iterations for different number of samples $k \in \{1, 2, 5, 10\}$ when $r = 10$ is fixed. As expected, in Figure 6(a), the error decreases monotonically, with the rate of $1/t$, till it hits a floor corresponding the offset incurred by the approximation error. Moreover, the error floor decreases with the number of expansion coefficients. On the other hand, in Figure 6(b), increasing the number of samples causes a downward shift in the error. This behavior is also expected since increasing the number of samples reduces the variance of the empirical expectation in Equation (12).

5.1.3 EFFECT OF THE EDGE POTENTIAL SMOOTHNESS

In our next set of experiments, still on a chain with $n = 100$ vertices, we test the behavior of the SOSMP algorithm on graphs with edge potentials of varying degrees of smoothness. In all cases,

---

3. In particular, we approximate the integral update (4) with its Riemann sum over the range $\mathcal{X} = [-5, 5]$ and with 100 samples per unit time.
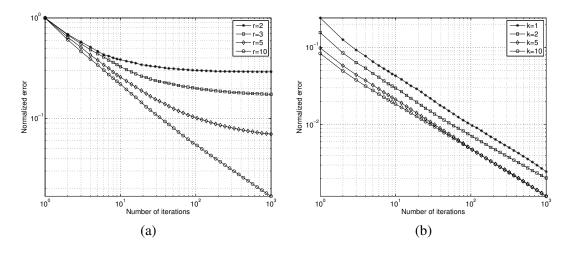
Figure 6: Normalized mean squared error $\mathbb{E}[e^t/e^0]$ verses the number of iterations for a Markov chain with $n = 100$ nodes, using potential functions specified by the mixture of Gaussians model (27) and (28). (a) Behavior as the number of expansion coefficients is varied over the range $r \in \{2, 3, 5, 10\}$ with $k = 5$ samples in all cases. As predicted by the theory, the error drops monotonically with the number of iterations until it hits a floor. The error floor, which corresponds to the approximation error incurred by message expansion truncation, decreases as the number of coefficients r is increased. (b) Mean squared error $\mathbb{E}[e^t]$ verses the number of iterations t for different number of samples $k \in \{1, 2, 5, 10\}$, in all cases using $r = 10$ coefficients. Increasing the number of samples k results in a downward shift in the error.

we use node potentials from the Gaussian mixture ensemble (27) previously discussed, but form the edge potentials in terms of a family of kernel functions. More specifically, consider the basis functions

$$\phi_j(x) = \sin\left((2j-1)\pi(x+5)/10\right) \quad \text{for } j = 1, 2, \ldots.$$

each defined on the interval $[-5, 5]$. It is straightforward that the family $\{\phi_j\}_{j=1}^{\infty}$ forms an orthonormal basis of $L^2[-5, 5]$. We use this basis to form the edge potential functions

$$\psi_{uv}(x, y) = \sum_{j=1}^{1000} (1/j)^{\alpha} \phi_j(x) \phi_j(y),$$

where $\alpha > 0$ is a parameter to be specified. By construction, each edge potential is a positive semidefinite kernel function satisfying the α-polynomial decay condition (24).

Figure 7 illustrate the error curves for two different choices of the smoothness parameter: panel (a) shows $\alpha = 0.1$, whereas panel (b) shows $\alpha = 1$. For the larger value of α shown in panel (b), the messages in the BP algorithm are smoother, so that the SOSMP estimates are more accurate with the same number of expansion coefficients. Moreover, similar to what we have observed previously, the error decays with the rate of $1/t$ till it hits the error floor. Note that this error floor is lower for
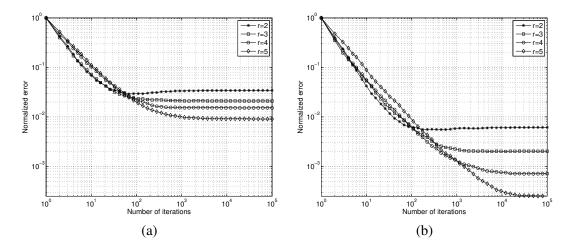
(a)   (b)

Figure 7: Plot of the estimation error e^t/e^0 verses the number of iterations t for the cases of (a) $\alpha = 0.1$ and (b) $\alpha = 1$. The BP messages are smoother when $\alpha = 1$, and accordingly the SOSMP estimates are more accurate with the same number of expansion coefficients. Moreover, the error decays with the rate of $1/t$ till it hits a floor corresponding to the approximation error incurred by truncating the message expansion coefficients.

the smoother kernel ($\alpha = 1$) compared to the rougher case ($\alpha = 0.1$). Moreover, as predicted by our theory, the approximation error decays faster for the smoother kernel, as shown by the plots in Figure 8, in which we plot the final error, due purely to approximation effects, versus the number of expansion coefficients r. The semilog plot of Figure 8 shows that the resulting lines have different slopes, as would be expected.

5.2 Computer Vision Application

Moving beyond simulated problems, we conclude by showing the SOSMP algorithm in application to a larger scale problem that arises in computer vision—namely, that of optical flow estimation (Baker et al., 2011). In this problem, the input data are two successive frames of a video sequence. We model each frame as a collection of pixels arrayed over a $\sqrt{n} \times \sqrt{n}$ grid, and measured intensity values at each pixel location of the form $\{I(i,j), I'(i,j)\}_{i,j=1}^{\sqrt{n}}$. Our goal is to estimate a 2-dimensional motion vector $x_u = (x_{u;1}, x_{u;2})$ that captures the local motion at each pixel $u = (i,j)$, $i,j = 1,2,\ldots,\sqrt{n}$ of the image sequence.

In order to cast this optical flow problem in terms of message-passing on a graph, we adopt the model used by Boccignone et al. (2007). We model the local motion X_u as a 2-dimensional random vector taking values in the space $\mathcal{X} = [-d,d] \times [-d,d]$, and associate the random vector X_u with vertex u, in a 2-dimensional grid (see Figure 1(a)). At node $u = (i,j)$, we use the change between the two image frames to specify the node potential

$$\psi_u(x_{u;1}, x_{u;2}) \propto \exp\left(-\frac{(I(i,j) - I'(i+x_{u;1},\ j+x_{u;2}))^2}{2\sigma_u^2}\right).$$
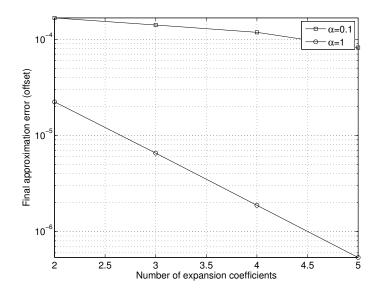
2817

Figure 8: Final approximation error vs. the number of expansion coefficients for the cases of $\alpha = 0.1$ and $\alpha = 1$. As predicted by the theory, the error floor decays with a faster pace for the smoother edge potential.

(a)   (b)

Figure 9: Two frames, each of dimension 250×250 pixels, taken from a video sequence of moving cars.

On each edge (u, v), we introduce the potential function

$$\psi_{uv}(x_u, x_v) \propto \exp\left(-\frac{\|x_u - x_v\|^2}{2\sigma_{uv}^2}\right),$$

which enforces a type of smoothness prior over the image.

To estimate the motion of a truck, we applied the SOSMP algorithm using the 2-dimensional Fourier expansion as our orthonormal basis to two 250×250 frames from a truck video sequence (see Figure 9). We apply the SOSMP algorithm using the first $r = 9$ coefficients and $k = 3$ samples. Figure 10 shows the HSV (hue, saturation, value) codings of the estimated motions after $t = 1, 10, 40$ iterations, in panels (a), (b) and (c) respectively. Panel (d) provides an illustration of the HSV encoding: hue is used to represent the angular direction of the motion whereas the speed (magnitude of the motion) is encoded by the saturation (darker colors meaning higher speeds). The initial estimates of the motion vectors are noisy, but it fairly rapidly converges to a reasonable optical flow field. (To be clear, the purpose of this experiment is not to show the effectiveness of SOSMP or BP as a particular method for optical flow, but rather to demonstrate its correctness and feasibility of the SOSMP in an applied setting.)

6. Proofs

We now turn to the proofs of our main results, which involve various techniques from concentration of measure, stochastic approximation, and functional analysis. For the reader's convenience, we provide a high-level outline here. We begin in Section 6.1 by proving Theorem 2 dealing with the case of tree-structured graphical models. In Section 6.2, we turn to the proof of Theorem 3 concerning the case of general graphs. Finally, in Section 6.3 we characterize the trade-off between computational complexity and accuracy by proving Theorem 4. Each section involve technical steps some of which are presented as lemmas. To increase the readability of the paper, proofs of the lemmas are deferred to the appendices and can be ignored without affecting the flow of the main argument.

6.1 Trees: Proof of Theorem 2

Our goal is to bound the error

$$\|\Delta_{v \to u}^{t+1}\|_{L^2}^2 \;=\; \|m_{v \to u}^{t+1} - \Pi^r(m_{v \to u}^*)\|_{L^2}^2 \;=\; \sum_{j=1}^{r} \left(a_{v \to u; j}^{t+1} - a_{v \to u; j}^*\right)^2, \tag{30}$$

where the final equality follows by Parseval's theorem. Here $\{a_{v \to u; j}^*\}_{j=1}^{r}$ are the basis expansion coefficients that define the best r-approximation to the BP fixed point m^*. The following lemma provides an upper bound on this error in terms of two related quantities. First, we let $\{b_{v \to u; j}^t\}_{j=1}^{\infty}$ denote the basis function expansion coefficients of the $\mathcal{F}_{v \to u}(\widehat{m}^t)$—that is, $[\mathcal{F}_{v \to u}(\widehat{m}^t)](\cdot) = \sum_{j=1}^{\infty} b_{v \to u; j}^t \phi_j(\cdot)$. Second, for each $j = 1, 2, \ldots, r$, define the deviation $\zeta_{v \to u; j}^{t+1} := \widetilde{b}_{v \to u; j}^{t+1} - b_{v \to u; j}^t$, where the coefficients $\widetilde{b}_{v \to u; j}^{t+1}$ are updated in Step 2(c) Figure 3.

Lemma 7 *For each iteration $t = 0, 1, 2, \ldots$, we have*

$$\|\Delta_{v \to u}^{t+1}\|_{L^2}^2 \;\leq\; \underbrace{\frac{2}{t+1} \sum_{j=1}^{r} \sum_{\tau=0}^{t} \left[b_{v \to u; j}^{\tau} - a_{v \to u; j}^*\right]^2}_{\text{Deterministic term } D_{v \to u}^{t+1}} + \underbrace{\frac{2}{(t+1)^2} \sum_{j=1}^{r} \left[\sum_{\tau=0}^{t} \zeta_{v \to u; j}^{\tau+1}\right]^2}_{\text{Stochastic term } S_{v \to u}^{t+1}}. \tag{31}$$
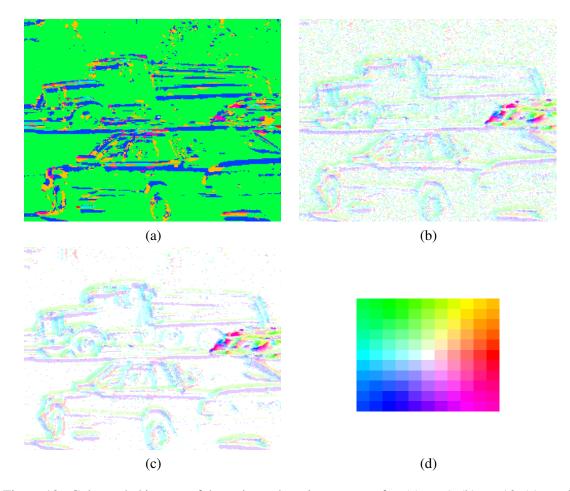
(a)

(b)

(c)

(d)

Figure 10: Color coded images of the estimated motion vectors after (a) $t = 1$, (b) $t = 10$, (c) $t = 40$ iterations. Panel (d) illustrates the hsv color coding of the flow. The color hue is used to encode the angular dimension of the motion, whereas the saturation level corresponds to the speed (length of motion vector). We implemented the SOSMP algorithm by expanding in the two-dimensional Fourier basis, using $r = 9$ coefficients and $k = 3$ samples. Although the initial estimates are noisy, it converges to a reasonable optical flow estimate after around 40 iterations.

The proof of this lemma is relatively straightforward; see Appendix A for the details. Note that inequality (31) provides an upper bound on the error that involves two terms: the first term $D_{v \to u}^{t+1}$ depends only on the expansion coefficients $\{b_{v \to u;j}^{\tau}, \tau = 0, \ldots, t\}$ and the BP fixed point, and therefore is a deterministic quantity when we condition on all randomness in stages up to step t. The second term $S_{v \to u}^{t+1}$, even when conditioned on randomness through step t, remains stochastic, since the coefficients $\tilde{b}_{v \to u}^{t+1}$ (involved in the error term $\zeta_{v \to u}^{t+1}$) are updated stochastically in moving from iteration t to $t + 1$.

We split the remainder of our analysis into three parts: (a) control of the deterministic component; (b) control of the stochastic term; and (c) combining the pieces to provide a convergence bound.

6.1.1 UPPER-BOUNDING THE DETERMINISTIC TERM

By the Pythagorean theorem, we have

$$\sum_{\tau=0}^{t} \sum_{j=1}^{r} \left[b_{v\to u;j}^{\tau} - a_{v\to u;j}^{*} \right]^2 \le \sum_{\tau=0}^{t} \| \mathcal{F}_{v\to u}(\widehat{m}^t) - \mathcal{F}_{v\to u}(m^*) \|_{L^2}^2. \tag{32}$$

In order to control this term, we make use of the following lemma, proved in Appendix B:

Lemma 8 *For all directed edges* $(v\to u) \in \vec{\mathcal{E}}$, *there exist constants* $\{L_{v\to u;w}, w \in \mathcal{N}(v) \setminus \{u\}\}$ *such that*

$$\| \mathcal{F}_{v\to u}(\widehat{m}^t) - \mathcal{F}_{v\to u}(m^*) \|_{L^2} \le \sum_{w\in\mathcal{N}(v)\setminus\{u\}} L_{v\to u;w} \| \widehat{m}_{w\to v}^t - m_{w\to v}^* \|_{L^2},$$

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

Substituting the result of Lemma 8 in Equation (32) and performing some algebra, we find that

$$\sum_{\tau=0}^{t} \sum_{j=1}^{r} \left[b_{v\to u;j}^{\tau} - a_{v\to u;j}^{*} \right]^2 \le \sum_{\tau=0}^{t} \left(\sum_{w\in\mathcal{N}(v)\setminus\{u\}} L_{v\to u;w} \| \widehat{m}_{w\to v}^{\tau} - m_{w\to v}^* \|_{L^2} \right)^2$$

$$\le (d_v - 1) \sum_{\tau=0}^{t} \sum_{w\in\mathcal{N}(v)\setminus\{u\}} L_{v\to u;w}^2 \| \widehat{m}_{w\to v}^{\tau} - m_{w\to v}^* \|_{L^2}^2, \tag{33}$$

where d_v is the degree of node $v \in \mathcal{V}$. By definition, the message $\widehat{m}_{w\to v}^{\tau}$ is the L^2-projection of $m_{w\to v}^{\tau}$ onto $\mathcal{M}$. Since $m_{w\to v}^* \in \mathcal{M}$ and projection is non-expansive, we have

$$\| \widehat{m}_{w\to v}^{\tau} - m_{w\to v}^* \|_{L^2}^2 \le \| m_{w\to v}^{\tau} - m_{w\to v}^* \|_{L^2}^2$$

$$= \| \Delta_{w\to v}^{\tau} \|_{L^2}^2 + \| A_{w\to v}^r \|_{L^2}^2, \tag{34}$$

where in the second step we have used the Pythagorean identity and recalled the definitions of estimation error as well as approximation error from (14) and (15). Substituting the inequality (34) into the bound (33) yields

$$\sum_{\tau=0}^{t} \sum_{j=1}^{r} \left[b_{v\to u;j}^{\tau} - a_{v\to u;j}^{*} \right]^2 \le (d_v - 1) \sum_{\tau=0}^{t} \sum_{w\in\mathcal{N}(v)\setminus\{u\}} L_{v\to u;w}^2 \left(\| \Delta_{w\to v}^{\tau} \|_{L^2}^2 + \| A_{w\to v}^r \|_{L^2}^2 \right).$$

Therefore, introducing the convenient shorthand $\widetilde{L}_{v\to u,w} := 2(d_v - 1) L_{v\to u;w}^2$, we have shown that

$$D_{v\to u}^{t+1} \le \frac{1}{t+1} \sum_{\tau=0}^{t} \sum_{w\in\mathcal{N}(v)\setminus\{u\}} \widetilde{L}_{v\to u,w} \left(\| \Delta_{w\to v}^t \|_{L^2}^2 + \| A_{w\to v}^r \|_{L^2}^2 \right). \tag{35}$$

We make further use of this inequality shortly.

6.1.2 CONTROLLING THE STOCHASTIC TERM

We now turn to the stochastic part of the inequality (31). Our analysis is based on the following fact, proved in Appendix C:

Lemma 9 *For each $t \geq 0$, let $\mathcal{G}^t := \sigma(m^0, \ldots, m^t)$ be the σ-field generated by all messages through time t. Then for every fixed $j = 1, 2, \ldots, r$, the sequence $\zeta_{v \to u; j}^{t+1} = \widetilde{b}_{v \to u; j}^{t+1} - b_{v \to u; j}^t$ is a bounded martingale difference with respect to $\{\mathcal{G}^t\}_{t=0}^{\infty}$. In particular, we have $|\zeta_{v \to u; j}^{t+1}| \leq 2B_j$, where B_j was previously defined* (19).

Based on Lemma 9, standard martingale convergence results (Durrett, 1995) guarantee that for each $j = 1, 2, \ldots, r$, we have $\sum_{\tau=0}^{t} \zeta_{v \to u; j}^{\tau+1} / (t+1)$ converges to 0 almost surely (a.s.) as $t \to \infty$, and hence

$$S_{v \to u}^{t+1} = \frac{2}{(t+1)^2} \sum_{j=1}^{r} \left\{ \sum_{\tau=0}^{t} \zeta_{v \to u; j}^{\tau+1} \right\}^2 = 2 \sum_{j=1}^{r} \left\{ \frac{1}{t+1} \sum_{\tau=0}^{t} \zeta_{v \to u; j}^{\tau+1} \right\}^2 \xrightarrow{\text{a.s.}} 0. \tag{36}$$

Furthermore, we can apply the Azuma-Hoeffding inequality (Chung and Lu, 2006) in order to characterize the rate of convergence. For each $j = 1, 2, \ldots, r$, define the non-negative random variable $Z_j := \left\{ \sum_{\tau=0}^{t} \zeta_{v \to u; j}^{\tau+1} \right\}^2 / (t+1)^2$. Since $|\zeta_{v \to u; j}^{\tau+1}| \leq 2B_j$, for any $\delta \geq 0$, we have

$$\mathbb{P}(Z_j \geq \delta) = \mathbb{P}(\sqrt{Z_j} \geq \sqrt{\delta}) \leq 2 \exp\left(-\frac{(t+1)\,\delta}{8\,B_j^2} \right),$$

for all $\delta > 0$. Moreover, Z_j is non-negative; therefore, integrating its tail bound we can compute the expectation

$$\mathbb{E}[Z_j] = \int_0^{\infty} \mathbb{P}(Z_j \geq \delta)\, d\delta \leq 2 \int_0^{\infty} \exp\left(-\frac{(t+1)\,\delta}{8\,B_j^2} \right) d\delta = \frac{16B_j^2}{t+1},$$

and consequently

$$\mathbb{E}\big[|S_{v \to u}^{t+1}|\big] \leq \frac{32 \sum_{j=1}^{r} B_j^2}{t+1}.$$

6.1.3 ESTABLISHING CONVERGENCE

We now make use of the results established so far to prove the claims. Substituting the upper bound (35) on $D_{v \to u}^{t+1}$ into the decomposition (31) from Lemma 7, we find that

$$\|\Delta_{v \to u}^{t+1}\|_{L^2}^2 \leq \frac{1}{t+1} \sum_{\tau=0}^{t} \sum_{w \in \mathcal{N}(v) \setminus \{u\}} \widetilde{L}_{v \to u, w} \left\{ \|\Delta_{w \to v}^{\tau}\|_{L^2}^2 + \|A_{w \to v}^{r}\|_{L^2}^2 \right\} + S_{v \to u}^{t+1}. \tag{37}$$

For convenience, let us introduce the vector $T^{t+1} = \{T_{v \to u}^{t+1}, (v \to u) \in \vec{\mathcal{E}}\} \in \mathbb{R}^{|\vec{\mathcal{E}}|}$ with entries

$$T_{v \to u}^{t+1} := \frac{1}{t+1} \left\{ \sum_{w \in \mathcal{N}(v) \setminus \{u\}} \widetilde{L}_{v \to u, w} \|\Delta_{w \to v}^{0}\|_{L^2}^2 \right\} + S_{v \to u}^{t+1}. \tag{38}$$

Now define a matrix $N \in \mathbb{R}^{|\vec{\mathcal{E}}| \times |\vec{\mathcal{E}}|}$ with entries indexed by the directed edges and set to

$$N_{v \to u,\, w \to s} := \begin{cases} \widetilde{L}_{v \to u, w} & \text{if } s = v \text{ and } w \in \mathcal{N}(v) \setminus \{u\} \\ 0 & \text{otherwise.} \end{cases}$$

In terms of this matrix and the error terms $\rho^2(\cdot)$ previously defined in Equations (16) and (17), the scalar inequalities (37) can be written in the matrix form

$$\rho^2(\Delta^{t+1}) \preceq N \left[\frac{1}{t+1} \sum_{\tau=1}^{t} \rho^2(\Delta^\tau) \right] + N \rho^2(A^r) + T^{t+1}, \tag{39}$$

where $\preceq$ denotes the element-wise inequality based on the orthant cone.

From Lemma 1 in the paper by Noorshams and Wainwright (2013), we know that the matrix N is guaranteed to be nilpotent with degree ℓ equal to the graph diameter. Consequently, unwrapping the recursion (39) for a total of $\ell = \text{diam}(\mathcal{G})$ times yields

$$\rho^2(\Delta^{t+1}) \preceq T_0^{t+1} + N T_1^{t+1} + \ldots + N^{\ell-1} T_{\ell-1}^{t+1} + (N + N^2 + \ldots + N^\ell) \rho^2(A^r),$$

where we define $T_0^{t+1} \equiv T^{t+1}$, and then recursively $T_s^{t+1} := (\sum_{\tau=1}^{t} T_{s-1}^\tau)/(t+1)$ for $s = 1, 2, \ldots, \ell - 1$. By the nilpotency of N, we have the identity $I + N + \ldots + N^{\ell-1} = (I - N)^{-1}$; so we can further simplify the last inequality

$$\rho^2(\Delta^{t+1}) \preceq \sum_{s=0}^{\ell-1} N^s T_s^{t+1} + N (I - N)^{-1} \rho^2(A^r). \tag{40}$$

Recalling the definition $\mathcal{B} := \{ e \in \mathbb{R}^{|\vec{\mathcal{E}}|} \mid |e| \preceq N(I-N)^{-1}\rho^2(A^r) \}$, inequality (40) implies that

$$\left| \rho^2(\Delta^{t+1}) - \Pi_{\mathcal{B}}(\rho^2(\Delta^{t+1})) \right| \preceq \sum_{s=0}^{\ell-1} N^s T_s^{t+1}. \tag{41}$$

We now use the bound (41) to prove both parts of Theorem 2.

To prove the almost sure convergence claim in part (a), it suffices to show that for each $s = 0, 1, \ldots, \ell - 1$, we have $T_s^t \xrightarrow{\text{a.s.}} 0$ as $t \to +\infty$. From Equation (36) we know $S_{v \to u}^{t+1} \to 0$ almost surely as $t \to \infty$. In addition, the first term in (38) is at most $O(1/t)$, so that also converges to zero as $t \to \infty$. Therefore, we conclude that $T_0^t \xrightarrow{\text{a.s.}} 0$ as $t \to \infty$. In order to extend this argument to higher-order terms, let us recall the following elementary fact from real analysis (Royden, 1988): for any sequence of real numbers $\{x^t\}_{t=0}^{\infty}$ such that $x^t \to 0$, then we also have $(\sum_{\tau=0}^{t} x^\tau)/t \to 0$. In order to apply this fact, we observe that $T_0^t \xrightarrow{\text{a.s.}} 0$ means that for almost every sample point ω the deterministic sequence $\{T_0^{t+1}(\omega)\}_{t=0}^{\infty}$ converges to zero. Consequently, the above fact implies that $T_1^{t+1}(\omega) = (\sum_{\tau=1}^{t} T_0^\tau(\omega))/(t+1) \to 0$ as $t \to \infty$ for almost all sample points ω, which is equivalent to asserting that $T_1^t \xrightarrow{\text{a.s.}} \vec{0}$. Iterating the same argument, we establish $T_s^{t+1} \xrightarrow{\text{a.s.}} \vec{0}$ for all $s = 0, 1, \ldots, \ell - 1$, thereby concluding the proof of Theorem 2(a).

Taking the expectation on both sides of the inequality (41) yields

$$\mathbb{E}\left[|\rho^2(\Delta^{t+1}) - \Pi_{\mathcal{B}}(\rho^2(\Delta^{t+1}))| \right] \preceq \sum_{s=0}^{\ell-1} N^s \mathbb{E}[T_s^{t+1}] \tag{42}$$

so that it suffices to upper bound the expectations $\mathbb{E}[T_s^{t+1}]$ for $s = 0, 1, \ldots, \ell - 1$. In Appendix D, we prove the following result:

Lemma 10 *Define the $|\vec{\mathcal{E}}|$-vector $\vec{v} := \left\{ \sum_{j=1}^{r} B_j^2 \right\} (4N\vec{1} + 32)$. Then for all $s = 0, 1, \ldots, \ell - 1$ and $t = 0, 1, 2, \ldots,$*

$$\mathbb{E}[T_s^{t+1}] \preceq \frac{\vec{v}}{t+1} \left(\sum_{u=0}^{s} \frac{(\log(t+1))^u}{u!} \right). \tag{43}$$

Using this lemma, the proof of part (b) follows easily. In particular, substituting the bounds (43) into Equation (42) and doing some algebra yields

$$\mathbb{E}\big[|\rho^2(\Delta^{t+1}) - \Pi_{\mathcal{B}}(\rho^2(\Delta^{t+1}))|\big] \preceq \sum_{s=0}^{\ell-1} N^s \sum_{u=0}^{s} \frac{(\log(t+1))^u}{u!} \left(\frac{\vec{v}}{t+1} \right)$$

$$\preceq 3 \sum_{s=0}^{\ell-1} (\log(t+1))^s N^s \left(\frac{\vec{v}}{t+1} \right)$$

$$\preceq 3 \left(I - \log(t+1) N \right)^{-1} \left(\frac{\vec{v}}{t+1} \right),$$

where again we used the fact that $N^\ell = 0$.

6.2 General Graphs: Proof of Theorem 3

Recall the definition of the estimation error $\Delta_{v \to u}^t$ from (14). By Parseval's identity we know that $\|\Delta_{v \to u}^t\|_{L^2}^2 = \sum_{j=1}^{r} (a_{v \to u;j}^t - a_{v \to u;j}^*)^2$. For convenience, we introduce the following shorthands for mean squared error on the directed edge $(v \to u)$

$$\bar{\rho}^2(\Delta_{v \to u}^t) := \mathbb{E}[\|\Delta_{v \to u}^t\|_{L^2}^2] = \mathbb{E}\left[\sum_{j=1}^{r} (a_{v \to u;j}^t - a_{v \to u;j}^*)^2 \right],$$

as well as the ℓ_∞ error

$$\bar{\rho}_{\max}^2(\Delta^t) := \max_{(v \to u) \in \vec{\mathcal{E}}} \mathbb{E}[\|\Delta_{v \to u}^t\|_{L^2}^2],$$

similarly defined for approximation error

$$\rho_{\max}^2(A^r) := \max_{(v \to u) \in \vec{\mathcal{E}}} \|A_{v \to u}^r\|_{L^2}^2.$$

Using the definition of $\bar{\rho}^2(\Delta_{v \to u}^t)$, some algebra yields

$$\bar{\rho}^2(\Delta_{v \to u}^{t+1}) - \bar{\rho}^2(\Delta_{v \to u}^t) = \mathbb{E}\left[\sum_{j=1}^{r} \left(a_{v \to u;j}^{t+1} - a_{v \to u;j}^* \right)^2 - \sum_{j=1}^{r} \left(a_{v \to u;j}^t - a_{v \to u;j}^* \right)^2 \right]$$

$$= \mathbb{E}\left[\sum_{j=1}^{r} \left\{ a_{v \to u;j}^{t+1} - a_{v \to u;j}^t \right\} \left\{ \left(a_{v \to u;j}^{t+1} - a_{v \to u;j}^t \right) + 2 \left(a_{v \to u;j}^t - a_{v \to u;j}^* \right) \right\} \right].$$

From the update Equation (13), we have

$$a_{v \to u;j}^{t+1} - a_{v \to u;j}^t = \eta^t \left(\widetilde{b}_{v \to u;j}^{t+1} - a_{v \to u;j}^t \right),$$

and hence

$$\bar{\rho}^2(\Delta_{v\to u}^{t+1}) - \bar{\rho}^2(\Delta_{v\to u}^t) \; = \; U_{v\to u}^t + V_{v\to u}^t, \tag{44}$$

where

$$U_{v\to u}^t \; := \; (\eta^t)^2 \sum_{j=1}^r \mathbb{E}\left[\left(\tilde{b}_{v\to u;j}^{t+1} - a_{v\to u;j}^t\right)^2\right], \quad \text{and}$$

$$V_{v\to u}^t \; := \; 2\eta^t \sum_{j=1}^r \mathbb{E}\left[\left(\tilde{b}_{v\to u;j}^{t+1} - a_{v\to u;j}^t\right)\left(a_{v\to u;j}^t - a_{v\to u;j}^*\right)\right].$$

The following lemma, proved in Appendix E, provides upper bounds on these two terms.

Lemma 11 *For all iterations $t = 0, 1, 2, \ldots$, we have*

$$U_{v\to u}^t \; \leq \; 4(\eta^t)^2 \sum_{j=1}^r B_j^2, \quad \text{and} \tag{45}$$

$$V_{v\to u}^t \; \leq \; \eta^t\left(1 - \frac{\gamma}{2}\right)\rho_{max}^2(A^r) + \eta^t\left(1 - \frac{\gamma}{2}\right)\bar{\rho}_{max}^2(\Delta^t) - \eta^t\left(1 + \frac{\gamma}{2}\right)\bar{\rho}^2(\Delta_{v\to u}^t). \tag{46}$$

We continue upper-bounding $\bar{\rho}^2(\Delta_{v\to u}^{t+1})$ by substituting the results of Lemma 11 into Equation (44), thereby obtaining

$$\bar{\rho}^2(\Delta_{v\to u}^{t+1}) \; \leq \; 4(\eta^t)^2 \sum_{j=1}^r B_j^2 + \eta^t\left(1 - \frac{\gamma}{2}\right)\rho_{max}^2(A^r) + \eta^t\left(1 - \frac{\gamma}{2}\right)\bar{\rho}_{max}^2(\Delta^t)$$

$$+ \left\{1 - \eta^t\left(1 + \frac{\gamma}{2}\right)\right\}\bar{\rho}^2(\Delta_{v\to u}^t)$$

$$\leq \; 4(\eta^t)^2 \sum_{j=1}^r B_j^2 + \eta^t\left(1 - \frac{\gamma}{2}\right)\rho_{max}^2(A^r) + \left(1 - \eta^t\gamma\right)\bar{\rho}_{max}^2(\Delta^t).$$

Since this equation holds for all directed edges $(v \to u)$, taking the maximum over the left-hand side yields the recursion

$$\bar{\rho}_{max}^2(\Delta^{t+1}) \; \leq \; (\eta^t)^2 \bar{B}^2 + \eta^t\left(1 - \frac{\gamma}{2}\right)\rho_{max}^2(A^r) + \left(1 - \eta^t\gamma\right)\bar{\rho}_{max}^2(\Delta^t),$$

where we have introduced the shorthand $\bar{B}^2 = 4\sum_{j=1}^r B_j^2$. Setting $\eta^t = 1/(\gamma(t+1))$ and unwrapping this recursion, we find that

$$\bar{\rho}_{max}^2(\Delta^{t+1}) \; \leq \; \frac{\bar{B}^2}{\gamma^2}\sum_{\tau=1}^{t+1}\frac{1}{\tau(t+1)} + \frac{2 - \gamma}{2\gamma}\rho_{max}^2(A^r)$$

$$\leq \; \frac{2\bar{B}^2}{\gamma^2}\frac{\log(t+1)}{t+1} + \frac{1}{\gamma}\rho_{max}^2(A^r),$$

which establishes the claim.

6.3 Complexity versus Accuracy: Proof of Theorem 4

As discussed earlier, each iteration of the SOSMP algorithm requires $O(r)$ operations per edge. Consequently, it suffices to show that running the algorithm with $r = r^*$ coefficients for $(\sum_{j=1}^{r} \lambda_j^2)(1/\delta)\log(1/\delta)$ iterations suffices to achieve mean-squared error of the order of δ.

The bound (21) consists of two terms. In order to characterize the first term (estimation error), we need to bound B_j defined in (19). Using the orthonormality of the basis functions and the fact that the supremum is attainable over the compact space $\mathcal{X}$, we obtain

$$B_j = \max_{(v \to u) \in \vec{\mathcal{E}}} \sup_{y \in \mathcal{X}} \frac{\lambda_j \phi_j(y)}{\int_{\mathcal{X}} \psi_{uv}(x,y)\mu(dx)} = O(\lambda_j).$$

Therefore, the estimation error decays at the rate $O\big((\sum_{j=1}^{r} \lambda_j^2)(\log t/t)\big)$, so that a total of $t = O\big((\sum_{j=1}^{r} \lambda_j^2)(1/\delta)\log(1/\delta)\big)$ iterations are sufficient to reduce it to $O(\delta)$.

The second term (approximation error) in the bound (21) depends only on the choice of r, and in particular on the r-term approximation error $\|A_{v \to u}^r\|_{L^2}^2 = \|m_{v \to u}^* - \Pi^r(m_{v \to u}^*)\|_{L^2}^2$. To bound this term, we begin by representing $m_{v \to u}^*$ in terms of the basis expansion $\sum_{j=1}^{\infty} a_j^* \phi_j$. By the Pythagorean theorem, we have

$$\|m_{v \to u}^* - \Pi^r(m_{v \to u}^*)\|_{L^2}^2 = \sum_{j=r+1}^{\infty} (a_j^*)^2. \tag{47}$$

Our first claim is that $\sum_{j=1}^{\infty} (a_j^*)^2/\lambda_j < \infty$. Indeed, since m^* is a fixed point of the message update equation, we have

$$m_{v \to u}^*(\cdot) \propto \int_{\mathcal{X}} \psi_{uv}(\cdot, y) M(y) \mu(dy),$$

where $M(\cdot) := \psi_v(\cdot) \prod_{w \in \mathcal{N}(v) \setminus \{u\}} m_{w \to v}^*(\cdot)$. Exchanging the order of integrations using Fubini's theorem, we obtain

$$a_j^* = \langle m_{v \to u}^*, \phi_j \rangle_{L^2} \propto \int_{\mathcal{X}} \langle \phi_j(\cdot), \psi_{uv}(\cdot, y) \rangle_{L^2} M(y) \mu(dy). \tag{48}$$

By the eigenexpansion of ψ_{uv}, we have

$$\langle \phi_j(\cdot), \psi_{uv}(\cdot, y) \rangle_{L^2} = \sum_{k=1}^{\infty} \lambda_k \langle \phi_j, \phi_k \rangle_{L^2} \phi_k(y) = \lambda_j \phi_j(y).$$

Substituting back into our initial Equation (48), we find that

$$a_j^* \propto \lambda_j \int_{\mathcal{X}} \phi_j(y) M(y) \mu(dy) = \lambda_j \widetilde{a}_j,$$

where $\widetilde{a}_j$ are the basis expansion coefficients of M. Since the space $\mathcal{X}$ is compact, one can see that $M \in L^2(\mathcal{X})$, and hence $\sum_{j=1}^{\infty} \widetilde{a}_j^2 < \infty$. Therefore, we have

$$\sum_{j=1}^{\infty} \frac{(a_j^*)^2}{\lambda_j} \propto \sum_{j=1}^{\infty} \lambda_j \widetilde{a}_j^2 < +\infty,$$

where we used the fact that $\sum_{j=1}^{\infty} \lambda_j < \infty$.

We now use this bound to control the approximation error (47). For any $r = 1, 2, \ldots$, we have

$$\sum_{j=r+1}^{\infty} (a_j^*)^2 = \sum_{j=r+1}^{\infty} \lambda_j \frac{(a_j^*)^2}{\lambda_j} \leq \lambda_r \sum_{j=r+1}^{\infty} \frac{(a_j^*)^2}{\lambda_j} = O(\lambda_r),$$

using the non-increasing nature of the sequence $\{\lambda_j\}_{j=1}^{\infty}$. Consequently, by definition of r^* (22), we have

$$\|m_{v \to u}^* - \Pi^{r^*}(m_{v \to u}^*)\|_{L^2}^2 = O(\delta),$$

as claimed.

7. Conclusion

Belief propagation is a widely used message-passing algorithm for computing (approximate) marginals in graphical models. In this paper, we have presented and analyzed the SOSMP algorithm for running BP in models with continuous variables. It is based on two forms of approximation: a *deterministic approximation* that involves projecting messages onto the span of r basis functions, and a *stochastic approximation* that involves approximating basis coefficients via Monte Carlo techniques and damped updates. These approximations, while leading to an algorithm with substantially reduced complexity, are also controlled: we provide upper bounds on the convergence of the stochastic error, showing that it goes to zero as $O(\log t / t)$ with the number of iterations, and also control on the deterministic error. For graphs with relatively smooth potential functions, as reflected in the decay rate of their basis coefficients, we provided a quantitative bound on the total number of basic arithmetic operations required to compute the BP fixed point to within δ-accuracy. We illustrated our theoretical predictions using experiments on simulated graphical models, as well as in a real-world instance of optical flow estimation.

Our work leaves open a number of interesting questions. First, although we have focused exclusively on models with pairwise interactions, it should be possible to develop forms of SOSMP for higher-order factor graphs. Second, the bulk of our analysis was performed under a type of contractivity condition, as has been used in past works (Tatikonda and Jordan, 2002; Ihler et al., 2005; Mooij and Kappen, 2007; Roosta et al., 2008) on convergence of the standard BP updates. However, we suspect that this condition might be somewhat relaxed, and doing so would demonstrate applicability of the SOSMP algorithm to a larger class of graphical models. Finally, it would be interesting to see if the ideas presented in this work can be applied to other graph-based learning problems.

Acknowledgments

Appendix A. Proof of Lemma 7

Subtracting $a^*_{v \to u; j}$ from both sides of the update (13) in Step 2(c), we obtain

$$a^{t+1}_{v \to u; j} - a^*_{v \to u; j} = (1 - \eta^t) \left[a^t_{v \to u; j} - a^*_{v \to u; j} \right] + \eta^t \left[b^t_{v \to u; j} - a^*_{v \to u; j} \right] + \eta^t \zeta^{t+1}_{v \to u; j}. \qquad (49)$$

Setting $\eta^t = 1/(t+1)$ and unwrapping the recursion (49) then yields

$$a^{t+1}_{v \to u; j} - a^*_{v \to u; j} = \frac{1}{t+1} \sum_{\tau=0}^{t} \left[b^\tau_{v \to u; j} - a^*_{v \to u; j} \right] + \frac{1}{t+1} \sum_{\tau=0}^{t} \zeta^{\tau+1}_{v \to u; j}.$$

Squaring both sides of this equality and using the upper bound $(a+b)^2 \leq 2a^2 + 2b^2$, we obtain

$$\left(a^{t+1}_{v \to u; j} - a^*_{v \to u; j} \right)^2 \leq \frac{2}{(t+1)^2} \left\{ \sum_{\tau=0}^{t} \left[b^\tau_{v \to u; j} - a^*_{v \to u; j} \right] \right\}^2 + \frac{2}{(t+1)^2} \left\{ \sum_{\tau=0}^{t} \zeta^{\tau+1}_{v \to u; j} \right\}^2.$$

Summing over indices $j = 1, 2, \ldots, r$ and recalling the expansion (30), we find that

$$\| \Delta^t_{v \to u} \|^2_{L^2} \leq \sum_{j=1}^{r} \left\{ \frac{2}{(t+1)^2} \left\{ \sum_{\tau=0}^{t} \left[b^\tau_{v \to u; j} - a^*_{v \to u; j} \right] \right\}^2 + \frac{2}{(t+1)^2} \left\{ \sum_{\tau=0}^{t} \zeta^{\tau+1}_{v \to u; j} \right\}^2 \right\}$$

$$\overset{(i)}{\leq} \underbrace{\frac{2}{(t+1)} \sum_{j=1}^{r} \sum_{\tau=0}^{t} \left[b^\tau_{v \to u; j} - a^*_{v \to u; j} \right]^2}_{\text{Deterministic term } D^{t+1}_{v \to u}} + \underbrace{\frac{2}{(t+1)^2} \sum_{j=1}^{r} \left\{ \sum_{\tau=0}^{t} \zeta^{\tau+1}_{v \to u; j} \right\}^2}_{\text{Stochastic term } S^{t+1}_{v \to u}}.$$

Here step (i) follows from the elementary inequality

$$\left\{ \sum_{\tau=0}^{t} \left[b^\tau_{v \to u; j} - a^*_{v \to u; j} \right] \right\}^2 \leq (t+1) \sum_{\tau=0}^{t} \left[b^\tau_{v \to u; j} - a^*_{v \to u; j} \right]^2.$$

Appendix B. Proof of Lemma 8

Recall the probability density

$$[p_{v \to u}(m)](\cdot) \propto \beta_{v \to u}(\cdot) \prod_{w \in \mathcal{N}(v) \setminus \{u\}} m_{w \to v}(\cdot),$$

defined in Step 2 of the SOSMP algorithm. Using this shorthand notation, the claim of Lemma 1 can be re-written as $[\mathcal{F}_{v \to u}(m)](x) = \langle \Gamma_{uv}(x, \cdot), [p_{v \to u}(m)](\cdot) \rangle_{L^2}$. Therefore, applying the Cauchy-Schwartz inequality yields

$$\left| [\mathcal{F}_{v \to u}(m)](x) - [\mathcal{F}_{v \to u}(m')](x) \right|^2 \leq \| \Gamma_{uv}(x, \cdot) \|^2_{L^2} \, \| p_{v \to u}(m) - p_{v \to u}(m') \|^2_{L^2}.$$

Integrating both sides of the previous inequality over $\mathcal{X}$ and taking square roots yields

$$\| \mathcal{F}_{v \to u}(m) - \mathcal{F}_{v \to u}(m') \|_{L^2} \leq C_{uv} \, \| p_{v \to u}(m) - p_{v \to u}(m') \|_{L^2},$$

where we have denoted the constant $C_{uv} := \left(\int_{\mathcal{X}} |\Gamma_{uv}(x, y)|^2 \mu(dy) \mu(dx) \right)^{1/2}$.

The next step is to upper bound the term $\|p_{v\to u}(m) - p_{v\to u}(m')\|_{L^2}$. In order to do so, we first show that $p_{v\to u}(m)$ is a Frechet differentiable[4] operator on the space $\mathcal{M}' := \text{convhull}\{m^*, \oplus_{(v\to u)\in\vec{\mathcal{E}}} \mathcal{M}'_{v\to u}\}$, where

$$\mathcal{M}'_{v\to u} := \left\{ \widehat{m}_{v\to u} \,\middle|\, \widehat{m}_{v\to u} = \left[\mathbb{E}_{Y\sim f}\left[\Pi^r\left(\Gamma_{uv}(\cdot, Y)\right)\right]\right]_+, \text{ for some probability density } f \right\},$$

denotes the space of all feasible SOSMP messages on the directed edge $(v \to u)$. Doing some calculus using the chain rule, we calculate the partial directional (Gateaux) derivative of the operator $p_{v\to u}(m)$ with respect to the function $m_{w\to v}$. More specifically, for an arbitrary function $h_{w\to v}$, we have

$$[\mathcal{D}_w\, p_{v\to u}(m)](h_{w\to v}) = \frac{\beta_{v\to u}\prod_{s\in\mathcal{N}(v)\setminus\{u,w\}} m_{s\to v}}{\langle M_{v\to u}, \beta_{v\to u}\rangle_{L^2}}\, h_{w\to v}$$
$$- \frac{\beta_{v\to u}M_{v\to u}}{\langle M_{v\to u}, \beta_{v\to u}\rangle_{L^2}^2}\, \langle h_{w\to v}, \beta_{v\to u} \prod_{s\in\mathcal{N}(v)\setminus\{u,w\}} m_{s\to v}\rangle_{L^2},$$

where $M_{v\to u} = \prod_{w\in\mathcal{N}(v)\setminus\{u\}} m_{w\to v}$. Clearly the Gateaux derivative is linear and continuous. It is also bounded as will be shown now. Massaging the operator norm's definition, we obtain

$$\sup_{m\in\mathcal{M}'} \|\!|\mathcal{D}_w\, p_{v\to u}(m)|\!\|_2 = \sup_{m\in\mathcal{M}'}\sup_{h_{w\to v}\in\mathcal{M}'_{w\to v}} \frac{\|[\mathcal{D}_w\, p_{v\to u}(m)](h_{w\to v})\|_{L^2}}{\|h_{w\to v}\|_{L^2}}$$
$$\leq \sup_{m\in\mathcal{M}'} \frac{\sup_{x\in\mathcal{X}}\beta_{v\to u}(x)\prod_{s\in\mathcal{N}(v)\setminus\{u,w\}} m_{s\to v}(x)}{\langle M_{v\to u}, \beta_{v\to u}\rangle_{L^2}}$$
$$+ \sup_{m\in\mathcal{M}'} \frac{\|\beta_{v\to u}M_{v\to u}\|_{L^2}\,\|\beta_{v\to u}\prod_{s\in\mathcal{N}(v)\setminus\{u,w\}} m_{s\to v}\|_{L^2}}{\langle M_{v\to u}, \beta_{v\to u}\rangle_{L^2}^2}. \tag{50}$$

Since the space $\mathcal{X}$ is compact, the continuous functions $\beta_{v\to u}$ and $m_{s\to v}$ achieve their maximum over $\mathcal{X}$. Therefore, the numerator of (50) is bounded and we only need to show that the denominator is bounded away from zero.

For an arbitrary message $m_{v\to u}\in\mathcal{M}'_{v\to u}$ there exist $0 < \alpha < 1$ and a bounded probability density f so that

$$m_{v\to u}(x) = \alpha\, m^*_{v\to u}(x) + (1-\alpha)\left[\mathbb{E}_{Y\sim f}\left[\widetilde{\Gamma}_{uv}(x, Y)\right]\right]_+,$$

---

4. For the convenience of the reader, we state the notions of Gateaux and Frechet differentiability (Clarke, 2013; Fabian et al., 2011). For normed spaces X and Y, let U be an open subset of X, and let $F : X \to Y$ be an operator. For $x \in U$ and $z \in X$, the F is Gateaux differentiable at x in the direction z if and only if the following limit exists

$$[\mathcal{D}F(x)](z) := \lim_{t\to 0}\frac{F(x+tz) - F(x)}{t} = \frac{d}{dt}F(x+tz)\,|_{t=0}\,.$$

Moreover, the operator F is Frechet differentiable at x if there exists a bounded linear operator $\mathcal{D}F(x) : X \to Y$ such that

$$\lim_{z\to 0}\frac{\|F(x+z) - F(x) - [\mathcal{D}F(x)](z)\|}{\|z\|} = 0.$$

where we have introduced the shorthand $\widetilde{\Gamma}_{uv}(\cdot,y) := \Pi^r(\Gamma_{uv}(\cdot,y))$. According to Lemma 1, we know $m^*_{v\to u} = \mathbb{E}_Y[\Gamma_{uv}(\cdot,Y)]$, where $Y \sim p_{v\to u}(m^*)$. Therefore, denoting $p^* = p_{v\to u}(m^*)$, we have

$$
\begin{aligned}
m_{v\to u}(x) &\geq \alpha\,\mathbb{E}_{Y\sim p^*}[\Gamma_{uv}(x,Y)] + (1-\alpha)\,\mathbb{E}_{Y\sim f}[\widetilde{\Gamma}_{uv}(x,Y)]\\
&= \mathbb{E}_{Y\sim(\alpha p^* + (1-\alpha)f)}[\widetilde{\Gamma}_{uv}(x,Y)] + \alpha\,\mathbb{E}_{Y\sim p^*}[\Gamma_{uv}(x,Y) - \widetilde{\Gamma}_{uv}(x,Y)].
\end{aligned}
\tag{51}
$$

On the other hand, since X is compact, we can exchange the order of expectation and projection using Fubini's theorem to obtain

$$
\mathbb{E}_{Y\sim p^*}[\Gamma_{uv}(\cdot,Y) - \widetilde{\Gamma}_{uv}(\cdot,Y)] = m^*_{v\to u} - \Pi^r(m^*_{v\to u}) = A^r_{v\to u}.
$$

Substituting the last equality into the bound (51) yields

$$
m_{v\to u}(x) \geq \inf_{y\in X}\widetilde{\Gamma}_{uv}(x,y) - |A^r_{v\to u}(x)|.
$$

Recalling the assumption (18), one can conclude that the right hand side of the above inequality is positive for all directed edges $(v \to u)$. Therefore, the denominator of the expression (50) is bounded away from zero and more importantly $\sup_{m\in\mathcal{M}'}\||\mathcal{D}_w\,p_{v\to u}(m)\||_2$ is attainable.

Since the derivative is a bounded, linear, and continuous operator, the Gateaux and Frechet derivatives coincides and we can use Proposition 2 (Luenberger, 1969, page 176) to obtain the following upper bound

$$
\|p_{v\to u}(m) - p_{v\to u}(m')\|_{L^2} \leq \sum_{w\in\mathcal{N}(v)\setminus\{u\}}\sup_{0\leq\alpha\leq 1}\||\mathcal{D}_w\,p_{v\to u}(m'+\alpha\,(m-m'))\||_2\,\|m_{w\to v} - m'_{w\to v}\|_{L^2}.
$$

Setting $L_{v\to u;w} := C_{uv}\sup_{m\in\mathcal{M}'}\||\mathcal{D}_w\,p_{v\to u}(m)\||_2$ and putting the pieces together yields

$$
\|\mathcal{F}_{v\to u}(m) - \mathcal{F}_{v\to u}(m')\|_{L^2} \leq \sum_{w\in\mathcal{N}(v)\setminus\{u\}}L_{v\to u;w}\,\|m_{w\to v} - m'_{w\to v}\|_{L^2},
$$

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

The last step of the proof is to verify that $m^* \in \mathcal{M}'$, and $\widehat{m}^t \in \mathcal{M}'$ for all $t = 1, 2, \ldots$. By definition we have $m^* \in \mathcal{M}'$. On the other hand, unwrapping the update (13) we obtain

$$
\begin{aligned}
a^t_{v\to u;j} &= \frac{1}{t}\sum_{\tau=0}^{t-1}\widetilde{b}^{\tau+1}_{v\to u;j}\\
&= \frac{1}{t}\sum_{\tau=0}^{t-1}\frac{1}{k}\sum_{\ell=1}^{k}\int_X \Gamma_{uv}(x,Y^{\tau+1}_\ell)\,\phi_j(x)\,\mu(dx)\\
&= \int_X \mathbb{E}_{Y\sim\hat{p}}[\Gamma_{uv}(x,Y)]\,\phi_j(x)\,\mu(dx),
\end{aligned}
$$

where $\hat{p}$ denotes the empirical probability density. Therefore, $m^t_{v\to u} = \sum_{j=1}^{r}a^t_{v\to u;j}\,\phi_j$ is equal to $\Pi^r(\mathbb{E}_{Y\sim\hat{p}}[\Gamma_{uv}(\cdot,Y)])$, thereby completing the proof.

Appendix C. Proof of Lemma 9

We begin by taking the conditional expectation of $\widetilde{b}_{v \to u;j}^{t+1}$, previously defined (12), given the filtration $\mathcal{G}^t$ and with respect to the random samples $\{Y_1^{t+1}, \ldots, Y_k^{t+1}\} \overset{\text{i.i.d.}}{\sim} [p_{v \to u}(\widehat{m})](\cdot)$. Exchanging the order of expectation and integral[5] and exploiting the result of Lemma 1, we obtain

$$\mathbb{E}[\widetilde{b}_{v \to u;j}^{t+1} \mid \mathcal{G}^t] = \int_X [\mathcal{F}_{v \to u}(\widehat{m}^t)](x)\, \phi_j(x)\, \mu(dx) = b_{v \to u;j}^t, \tag{52}$$

and hence $\mathbb{E}[\zeta_{v \to u;j}^{t+1} \mid \mathcal{G}^t] = 0$, for all $j = 1, 2, \ldots, r$ and all directed edges $(v \to u) \in \vec{\mathcal{E}}$. Also it is clear that $\zeta_{v \to u;j}^{t+1}$ is $\mathcal{G}^t$-measurable. Therefore, $\{\zeta_{v \to u;j}^{\tau+1}\}_{\tau=0}^{\infty}$ forms a martingale difference sequence with respect to the filtration $\{\mathcal{G}^\tau\}_{\tau=0}^{\infty}$. On the other hand, recalling the bound (19), we have

$$|\widetilde{b}_{v \to u;j}^{t+1}| \leq \frac{1}{k} \sum_{\ell=1}^{k} |\langle \Gamma_{uv}(\cdot, Y_\ell), \phi_j \rangle_{L^2}| \leq B_j.$$

Moreover, exploiting the result of Lemma 1 and exchanging the order of the integration and expectation once more yields

$$|b_{v \to u;j}^t| = |\langle \mathbb{E}_Y[\Gamma_{uv}(\cdot, Y)], \phi_j \rangle_{L^2}| = |\mathbb{E}_Y[\langle \Gamma_{uv}(\cdot, Y), \phi_j \rangle_{L^2}]| \leq B_j, \tag{53}$$

where we have $Y \sim [p_{v \to u}(\widehat{m}^t)](y)$. Therefore, the martingale difference sequence is bounded, in particular with

$$|\zeta_{v \to u;j}^{t+1}| \leq |\widetilde{b}_{v \to u;j}^{t+1}| + |b_{v \to u;j}^t| \leq 2B_j.$$

Appendix D. Proof of Lemma 10

We start by uniformly upper-bounding the terms $\mathbb{E}[|T_{v \to u}^{t+1}|]$. To do so we first need to bound $\|\Delta_{v \to u}^t\|_{L^2}$. By definition we know $\|\Delta_{v \to u}^t\|_{L^2}^2 = \sum_{j=1}^{r} [a_{v \to u;j}^t - a_{v \to u;j}^*]^2$; therefore we only need to control the terms $a_{v \to u;j}^t$ and $a_{v \to u;j}^*$ for $j = 1, 2, \ldots, r$.

By construction, we always have $|\widetilde{b}_{v \to u;j}^{t+1}| \leq B_j$ for all iterations $t = 0, 1, \ldots$. Also, assuming that $|a_{v \to u;j}^0| \leq B_j$, without loss of generality, a simple induction using the update Equation (13) shows that $|a_{v \to u;j}^t| \leq B_j$ for all t. Moreover, using a similar argument leading to (53), we obtain

$$|a_{v \to u;j}^*| = |\langle \mathbb{E}_Y[\Gamma_{uv}(\cdot, Y)], \phi_j \rangle_{L^2}| = |\mathbb{E}_Y[\langle \Gamma_{uv}(\cdot, Y), \phi_j \rangle_{L^2}]| \leq B_j,$$

where we have $Y \sim [p_{v \to u}(m^*)](y)$. Therefore, putting the pieces together, recalling the definition (38) of $T_{v \to u}^{t+1}$ yields

$$\mathbb{E}[|T_{v \to u}^{t+1}|] \leq \frac{4}{t+1} \sum_{w \in \mathcal{N}(v) \setminus \{u\}} \widetilde{L}_{v \to u,w} \sum_{j=1}^{r} B_j^2 + \frac{32}{t+1} \sum_{j=1}^{r} B_j^2.$$

---

5. Since $\Gamma_{uv}(x,y)\phi_i(x)[p_{v \to u}(\widehat{m}^t)](y)$ is absolutely integrable, we can exchange the order of the integrals using Fubini's theorem.

Concatenating the previous scalar inequalities yields $\mathbb{E}[T_0^{t+1}] \preceq \vec{v}/(t+1)$, for all $t \geq 0$, where we have defined the $|\vec{\mathcal{E}}|$-vector $\vec{v} := \{\sum_{j=1}^{r} B_j^2\}(4N\vec{1}+32)$. We now show, using an inductive argument, that

$$\mathbb{E}[T_s^{t+1}] \preceq \frac{\vec{v}}{t+1} \sum_{u=0}^{s} \frac{(\log(t+1))^u}{u!},$$

for all $s = 0, 1, 2, \ldots$ and $t = 0, 1, 2, \ldots$. We have already established the base case $s = 0$. For some $s > 0$, assume that the claim holds for $s - 1$. By the definition of T_s^{t+1}, we have

$$\mathbb{E}[T_s^{t+1}] = \frac{1}{t+1} \sum_{\tau=1}^{t} \mathbb{E}[T_{s-1}^{\tau}]$$

$$\preceq \frac{\vec{v}}{t+1} \sum_{\tau=1}^{t} \left\{ \frac{1}{\tau} + \sum_{u=1}^{s-1} \frac{(\log \tau)^u}{u! \, \tau} \right\},$$

where the inequality follows from the induction hypothesis. We now make note of the elementary inequalities $\sum_{\tau=1}^{t} 1/\tau \leq 1 + \log t$, and

$$\sum_{\tau=1}^{t} \frac{(\log \tau)^u}{u! \, \tau} \leq \int_1^t \frac{(\log x)^u}{u! \, x} \, dx = \frac{(\log t)^{(u+1)}}{(u+1)!}, \qquad \text{for all } u \geq 1$$

from which the claim follows.

Appendix E. Proof of Lemma 11

By construction, we always have $|\widetilde{b}_{v \to u; j}^{t+1}| \leq B_j$ for all iterations $t = 0, 1, 2, \ldots$. Moreover, assuming $|a_{v \to u; j}^0| \leq B_j$, without loss of generality, a simple induction on the update equation shows that $|a_{v \to u; j}^t| \leq B_j$ for all iterations $t = 0, 1, \ldots$. On this basis, we find that

$$U_{v \to u}^t = (\eta^t)^2 \sum_{j=1}^{r} \mathbb{E}\left[\left(\widetilde{b}_{v \to u; j}^{t+1} - a_{v \to u; j}^t \right)^2 \right] \leq 4(\eta^t)^2 \sum_{j=1}^{r} B_j^2,$$

which establishes the bound (45).

It remains to establish the bound (46) on $V_{v \to u}^t$. We first condition on the σ-field $\mathcal{G}^t = \sigma(m^0, \ldots, m^t)$ and take expectations over the remaining randomness, thereby obtaining

$$V_{v \to u}^t = 2\eta^t \, \mathbb{E}\left[\mathbb{E}\left[\sum_{j=1}^{r} \left(\widetilde{b}_{v \to u; j}^{t+1} - a_{v \to u; j}^t \right) \left(a_{v \to u; j}^t - a_{v \to u; j}^* \right) \,\big|\, \mathcal{G}^t \right] \right]$$

$$= 2\eta^t \, \mathbb{E}\left[\sum_{j=1}^{r} \left(b_{v \to u; j}^t - a_{v \to u; j}^t \right) \left(a_{v \to u; j}^t - a_{v \to u; j}^* \right) \right],$$

where $\{b_{v \to u; j}^t\}_{j=1}^{\infty}$ are the expansion coefficients of the function $\mathcal{F}_{v \to u}(\widehat{m}^t)$ (i.e., $b_{v \to u; j}^t = \langle \mathcal{F}_{v \to u}(\widehat{m}^t), \phi_j \rangle_{L^2}$), and we have recalled the result $\mathbb{E}[\widetilde{b}_{v \to u; j}^{t+1} | \mathcal{G}^t] = b_{v \to u; j}^t$ from (52). By Parseval's identity, we have

$$T := \sum_{j=1}^{r} \left(b_{v \to u; j}^t - a_{v \to u; j}^t \right) \left(a_{v \to u; j}^t - a_{v \to u; j}^* \right)$$

$$= \langle \Pi^r(\mathcal{F}_{v \to u}(\widehat{m}^t)) - m_{v \to u}^t, \, m_{v \to u}^t - \Pi^r(m_{v \to u}^*) \rangle_{L^2}.$$

Here we have used the basis expansions

$$m_{v \to u}^t = \sum_{j=1}^{r} a_{v \to u; j}^t \phi_j, \quad \text{and} \quad \Pi^r(m_{v \to u}^*) = \sum_{j=1}^{r} a_{v \to u; j}^* \phi_j.$$

Since $\Pi^r(m_{v \to u}^t) = m_{v \to u}^t$ and $\mathcal{F}_{v \to u}(m^*) = m_{v \to u}^*$, we have

$$
\begin{aligned}
T &= \langle \Pi^r (\mathcal{F}_{v \to u}(\widehat{m}^t) - \mathcal{F}_{v \to u}(m^*)), m_{v \to u}^t - \Pi^r(m_{v \to u}^*) \rangle_{L^2} - \|m_{v \to u}^t - \Pi^r(m_{v \to u}^*)\|_{L^2}^2 \\
&\overset{(i)}{\leq} \|\Pi^r(\mathcal{F}_{v \to u}(\widehat{m}^t) - \mathcal{F}_{v \to u}(m^*))\|_{L^2} \|m_{v \to u}^t - \Pi^r(m_{v \to u}^*)\|_{L^2} - \|m_{v \to u}^t - \Pi^r(m_{v \to u}^*)\|_{L^2}^2 \\
&\overset{(ii)}{\leq} \|\mathcal{F}_{v \to u}(\widehat{m}^t) - \mathcal{F}_{v \to u}(m^*)\|_{L^2} \|m_{v \to u}^t - \Pi^r(m_{v \to u}^*)\|_{L^2} - \|m_{v \to u}^t - \Pi^r(m_{v \to u}^*)\|_{L^2}^2.
\end{aligned}
$$

where step (i) uses the Cauchy-Schwarz inequality, and step (ii) uses the non-expansivity of projection. Applying the contraction condition (20), we obtain

$$
\begin{aligned}
T &\leq \left(1 - \frac{\gamma}{2}\right) \sqrt{\frac{\sum_{w \in \mathcal{N}(v) \setminus \{u\}} \|\widehat{m}_{w \to v}^t - m_{w \to v}^*\|_{L^2}^2}{|\mathcal{N}(v)| - 1}} \, \|m_{v \to u}^t - \Pi^r(m_{v \to u}^*)\|_{L^2} \\
&\quad - \|m_{v \to u}^t - \Pi^r(m_{v \to u}^*)\|_{L^2}^2 \\
&\leq \left(1 - \frac{\gamma}{2}\right) \left\{ \frac{1}{2} \frac{\sum_{w \in \mathcal{N}(v) \setminus \{u\}} \|m_{w \to v}^t - m_{w \to v}^*\|_{L^2}^2}{|\mathcal{N}(v)| - 1} + \frac{1}{2} \|m_{v \to u}^t - \Pi^r(m_{v \to u}^*)\|_{L^2}^2 \right\} \\
&\quad - \|m_{v \to u}^t - \Pi^r(m_{v \to u}^*)\|_{L^2}^2,
\end{aligned}
$$

where the second step follows from the elementary inequality $ab \leq a^2/2 + b^2/2$ and the non-expansivity of projection onto the space of non-negative functions. By the Pythagorean theorem, we have

$$
\begin{aligned}
\|m_{w \to v}^t - m_{w \to v}^*\|_{L^2}^2 &= \|m_{w \to v}^t - \Pi^r(m_{w \to v}^*)\|_{L^2}^2 + \|\Pi^r(m_{w \to v}^*) - m_{w \to v}^*\|_{L^2}^2 \\
&= \|\Delta_{w \to v}^t\|_{L^2}^2 + \|A_{w \to v}^r\|_{L^2}^2.
\end{aligned}
$$

Using this equality and taking expectations, we obtain

$$
\begin{aligned}
\mathbb{E}[T] &\leq \left(1 - \frac{\gamma}{2}\right) \left\{ \frac{1}{2} \frac{\sum_{w \in \mathcal{N}(v) \setminus \{u\}} [\bar{\rho}^2(\Delta_{w \to v}^t) + \|A_{w \to v}^r\|_{L^2}^2]}{|\mathcal{N}(v)| - 1} + \frac{1}{2} \bar{\rho}^2(\Delta_{v \to u}^t) \right\} - \bar{\rho}^2(\Delta_{v \to u}^t) \\
&\leq \left(\frac{1}{2} - \frac{\gamma}{4}\right) \rho_{\max}^2(A^r) + \left(\frac{1}{2} - \frac{\gamma}{4}\right) \bar{\rho}_{\max}^2(\Delta^t) - \left(\frac{1}{2} + \frac{\gamma}{4}\right) \bar{\rho}^2(\Delta_{v \to u}^t).
\end{aligned}
$$

Since $V_{v \to u}^t = 2\eta^t \mathbb{E}[T]$, the claim follows.

References

A. Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, May 2012.

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

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.

S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, March 2011.

G. Boccignone, A. Marcelli, P. Napoletano, and M. Ferraro. Motion estimation via belief propagation. In *Proceedings of the International Conference on Image Analysis and Processing*, pages 55–60, 2007.

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

F. Clarke. *Functional Analysis, Calculus of Variations and Optimal Control*. Springer-Verlag, London, 2013.

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

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

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

M. Fabian, P. Habala, P. Hajek, V. Montesinos, and V. Zizler. *Banach Space Theory: The Basis for Linear and Nonlinear Analysis*. Springer, New York, 2011.

C. Gu. *Smoothing Spline ANOVA Models*. Springer Series in Statistics. Springer, New York, NY, 2002.

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

A. T. Ihler and D. McAllester. Particle belief propagation. In *Proceedings of the Conference on Artificial Intelligence and Statistics*, pages 256–263, 2009.

M. Isard. PAMPAS: Real-valued graphical models for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 613–620, 2003.

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

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.

H. A. Loeliger. An introduction to factor graphs. *IEEE Signal Processing Magazine*, 21(1):28–41, 2004.

D. G. Luenberger. *Optimization by Vector Space Methods*. Wiley, New York, 1969.

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.

A. S. Nemirovsky and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. New York, 1983.

N. Noorshams and M. J. Wainwright. Quantized stochastic belief propagation: Efficient message-passing for continuous state spaces. In *Proceedings of the IEEE International Symposium on Information Theory*, 2012.

N. Noorshams and M. J. Wainwright. Stochastic belief propagation: A low-complexity alternative to the sum-product algorithms. *IEEE Transactions on Information Theory*, 59(4):1981–2000, April 2013.

F. Riesz and B.S. Nagy. *Functional Analysis*. Dover Publications Inc., New York, 1990.

B. D. Ripley. *Stochastic Simulation*. Wiley, New York, 1987.

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.

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

L. Song, A. Gretton, D. Bickson, Y. Low, and C. Guestrin. Kernel belief propagation. In *Proceedings of the Artificial Intelligence and Statistics*, 2011.

I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, New York, 2008.

E. B. Sudderth, A. T. Ihler, M. Israd, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. *Communications of the ACM Magazine*, 53(10):95–103, 2010.

S. Tatikonda and M. I. Jordan. Loopy belief propagation and Gibbs measures. In *Proc. Uncertainty in Artificial Intelligence*, volume 18, pages 493–500, August 2002.

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

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.