Particle Belief Propagation

Alexander Ihler
ihler@ics.uci.edu
Dept. of Computer Science
University of California, Irvine

David McAllester
mcallester@tti-c.org
Toyota Technological Institite, Chicago

Abstract

The popularity of particle filtering for inference in Markov chain models defined over random variables with very large or continuous domains makes it natural to consider sample–based versions of belief propagation (BP) for more general (tree–structured or loopy) graphs. Already, several such algorithms have been proposed in the literature. However, many questions remain open about the behavior of particle–based BP algorithms, and little theory has been developed to analyze their performance. In this paper, we describe a generic particle belief propagation (PBP) algorithm which is closely related to previously proposed methods. We prove that this algorithm is consistent, approaching the true BP messages as the number of samples grows large. We then use concentration bounds to analyze the finite-sample behavior and give a convergence rate for the algorithm on tree–structured graphs. Our convergence rate is $O(1/\sqrt{n})$ where n is the number of samples, independent of the domain size of the variables.

1 Introduction

Graphical models provide a powerful framework for representing structure in distributions over many random variables. This structure can then be used to efficiently compute or approximate many quantities of interest such as the posterior modes, means, or marginals of the distribution, often using "message-passing" algorithms such as belief propagation (Pearl, 1988). Traditionally, most such work has focused on systems of many variables, each of which has a relatively small state space (number of possible values), or particularly nice parametric forms (such as jointly Gaussian distributions). For systems with continuous-valued variables, or discrete-valued variables with very large domains, one possibility is to reduce the effective state space through gating, or discarding low-probability states (Freeman et al., 2000; Coughlan and Ferreira, 2002), or through random sampling (Arulampalam et al., 2002; Koller et al., 1999; Sudderth et al., 2003; Isard, 2003; Neal et al., 2003). The best-known example of the latter technique is *particle filtering*, defined on Markov chains, in which each distribution is represented using a finite collection of samples, or particles. It is therefore only natural to consider generalizations of particle filtering applicable to more general graphs ("particle" belief propagation); several variations have thus far been proposed, corresponding to different choices for certain fundamental questions.

As an example, consider the question of how to represent the messages computed during inference using particles. Broadly speaking, one might consider two possible approaches: to draw a set of particles for each message in the graph (Arulampalam et al., 2002; Sudderth et al., 2003; Isard, 2003), or to create a set of particles for each variable, for example by drawing samples from the estimated posterior marginal (Koller et al., 1999). This decision is closely related to the choice of proposal distribution in particle filtering; indeed, choosing better proposal distributions from which to draw the samples, or moving the samples via Markov chain Monte Carlo (MCMC) to match subsequent observations, comprises a large part of modern work on particle filters (Thrun et al., 2000; van der Merwe et al., 2001; Doucet et al., 2001; Khan et al., 2005).

Either method can be made asymptotically consistent, i.e., will produce the correct answer in the limit as the

Appearing in Proceedings of the 12^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5.

number of samples becomes infinite. However, consistency is a very weak condition—fundamentally, we are interested in the behavior of particle belief propagation for relatively small numbers of particles, ensuring computational efficiency. So far, little theory describes the finite sample behavior of these algorithms.

In this work, we give a convergence rate for the accuracy of a relatively generic PBP algorithm, most closely related to that described in Koller et al. (1999). Our convergence rate is $O(1/\sqrt{n})$ where n is the number of particles *independent of the domain size of the nodes of the graphical model.* The convergence rate is reminiscent, and has a similar proof, to convergence rates for learning algorithms derived from Chernoff bounds applied to IID samples.

2 Definitions and Notation

Let G be an undirected graph consisting of nodes $V = \{1, \ldots, k\}$ and edges E, and let Γ_s denote the set of neighbors of node s in G, i.e., the set of nodes t such that $\{s, t\}$ is an edge of G. In a probabilistic graphical model, each node $s \in V$ is associated with a random variable X_s taking on values in some domain, $\mathcal{X}_s$. We assume that each node s and edge $\{s, t\}$ are associated with potential functions Ψ_s and $\Psi_{s,t}$ respectively, and given these potential functions we define a probability distribution on assignments of values to nodes as

$$P(\vec{x}) = \frac{1}{Z}\left(\prod_s \Psi_s(\vec{x}_s)\right)\left(\prod_{\{s,t\}\in E} \Psi_{s,t}(\vec{x}_s, \vec{x}_t)\right) \quad (1)$$

Here $\vec{x}$ is an assignment of values to all k variables, $\vec{x}_s$ is the value assigned to X_s by $\vec{x}$, and Z is a scalar chosen to normalize the distribution P (also called the partition function). We consider the problem of computing marginal probabilities, defined by

$$P_s(x_s) = \sum_{\vec{x}:\vec{x}_s = x_s} P(\vec{x}). \quad (2)$$

Equation (1) defines a pairwise Markov random field model. Our results are also directly applicable to more general graphical model formulations such as Bayes' nets (Pearl, 1988) and factor graphs (Kschischang et al., 2001), but for notational convenience we restrict our development to the pairwise form (1).

3 Review of Belief Propagation

In the case where G is a tree and the sets $\mathcal{X}_s$ are small, the marginal probabilities can be computed efficiently by belief propagation (Pearl, 1988). This is done by computing messages $m_{t\to s}$ each of which is a function on the state space of the target node, $\mathcal{X}_s$. These messages are defined recursively as

$$m_{t\to s}(x_s) = \sum_{x_t\in\mathcal{X}_t} \Psi_{t,s}(x_t, x_s)\Psi_t(x_t) \prod_{u\in\Gamma_t\setminus s} m_{u\to t}(x_t) \quad (3)$$

When G is a tree this recursion is well founded (loop-free) and Equation (3) uniquely determines the messages. We define the unnormalized belief function as

$$B_s(x_s) = \Psi_s(x_s) \prod_{t\in\Gamma_s} m_{t\to s}(x_s). \quad (4)$$

When G is a tree the belief function is proportional to the marginal probability P_s defined by (2). It is sometimes useful to define the "pre-message" $M_{t\to s}$ as

$$M_{t\to s}(x_t) = \Psi_t(x_t) \prod_{u\in\Gamma_t\setminus s} m_{u\to t}(x_t) \quad (5)$$

for $x_t \in \mathcal{X}_t$. Note that the pre-message $M_{t\to s}$ defines a weighting on the state space of the source node $\mathcal{X}_t$, while the message $m_{t\to s}$ defines a weighting on the state space of the destination, $\mathcal{X}_s$. We can then re-express (3)–(4) as

$$m_{t\to s}(x_s) = \sum_{x_t\in\mathcal{X}_t} \Psi_{t,s}(x_t, x_s)M_{t\to s}(x_t)$$
$$B_t(x_t) = M_{t\to s}(x_t)m_{s\to t}(x_t)$$

Although we develop our results for tree–structured graphs, it is common to apply belief propagation to graphs with cycles as well ("loopy" belief propagation). In this case the belief functions (4) will in general not equal the true marginals, but often provide good approximations in practice. We discuss the application of our results to particle-based versions of loopy BP at the end of Section 5.

For reasons of numerical stability, it is common to normalize each message $m_{t\to s}$ so that it has unit sum. However, such normalization of messages has no other effect on the (normalized) belief functions (4). Thus for conceptual simplicity in developing and analyzing particle belief propagation we avoid any explicit normalization of the messages; such normalization can be included in the algorithms in practice.

Additionally, for reasons of computational efficiency it is common to use the alternative expression

$$m_{t\to s}(x_s) = \sum \Psi_{t,s}(x_t, x_s)\frac{B_t(x_t)}{m_{s\to t}(x_t)} \quad (6)$$

when computing the messages. By storing and updating the belief values $B_t(x_t)$ incrementally as incoming messages are re-computed, one can significantly reduce the number of operations required. Although our development of particle belief propagation uses the update form (3), this alternative formulation can be applied to improve its efficiency as well.

4 Particle Belief Propagation

We now consider the case where $|\mathcal{X}_s|$ is too large to enumerate in practice and define a generic particle (sample) based BP algorithm (PBP). This algorithm essentially corresponds to a non-iterative version of the method described in Koller et al. (1999).

4.1 Particle Belief Propagation Algorithm

PBP samples a set of particles $x_s^{(1)}, \ldots, x_s^{(n)}$ with $x_s^{(i)} \in \mathcal{X}_s$ at each node s of the network[1], drawn from a sampling distribution (or weighting) $W_s(x_s) > 0$ (corresponding to the proposal distribution in particle filtering). The selection of an appropriate sampling distribution is discussed in detail in section 6. First we note that (3) can be written as the following importance-sampling corrected expectation.

$$m_{t\to s}(x_s) = \mathrm{E}_{x_t\sim W_t}\left[\Psi_{s,t}(x_s,x_t)\frac{\Psi_t(x_t)}{W_t(x_t)}\prod_{u\in\Gamma_t\setminus s} m_{u\to t}(x_t)\right] \quad (7)$$

Given a sample $x_t^{(1)}, \ldots, x_t^{(n)}$ of points drawn from W_t we can estimate $m_{t\to s}(x_s^{(i)})$ as

$$\widehat{m}_{t\to s}^{(i)} = \frac{1}{n}\sum_{j=1}^{n}\Psi_{t,s}(x_t^{(j)},x_s^{(i)})\frac{\Psi_t(x_t^{(j)})}{W_t(x_t^{(j)})}\prod_{u\in\Gamma_t\setminus s}\widehat{m}_{u\to t}^{(j)} \quad (8)$$

Equation (8) represents a finite sample estimate for (7). Alternatively, (8) defines a belief propagation algorithm where messages are defined on particles rather than the entire set $\mathcal{X}_s$. As in classical belief propagation, for tree structured graphs and fixed particle locations there is a unique set of messages satisfying (8). Equation (8) can also be applied for loopy graphs (again observing that message normalization can be conceptually ignored). In this simple version, the sample values $x_s^{(i)}$ and weights $W_s(x_s^{(i)})$ remain unchanged as messages are updated.

4.2 Consistency of Particle BP

We now show that equation (8) is consistent—it agrees with (3) in the limit as $n \to \infty$. For any finite collection of samples, define the particle domain $\widehat{\mathcal{X}}_s$ and the count $c_s(x)$ for $x \in \widehat{\mathcal{X}}_s$ as

$$\widehat{\mathcal{X}}_s = \{x_s \in \mathcal{X}_s : \exists i\ x_s^{(i)} = x_s\}$$
$$c_s(x_s) = |\{i : x_s^{(i)} = x_s\}|$$

Equation (8) has the property that if $x_s^{(i)} = x_s^{(i')}$ then $m_{t\to s}^{(i)} = m_{t\to s}^{(i')}$; thus we can rewrite (8) as

$$\widehat{m}_{t\to s}(x_s) = \frac{1}{n}\sum_{x_t\in\widehat{\mathcal{X}}_t}\frac{c_t(x_t)}{W_t(x_t)}\Psi_{t,s}(x_t,x_s)\Psi_t(x_t) \cdot \prod_{u\in\Gamma_t\setminus s}\widehat{m}_{u\to t}(x_t) \quad (9)$$

for $x_s \in \widehat{\mathcal{X}}_s$. Since we have assumed $W_t(x_t) > 0$, in the limit of an infinite sample $\widehat{\mathcal{X}}_t$ becomes all of $\mathcal{X}_t$ and the ratio $(c_t(x_t)/n)$ converges to $W_t(x_t)$. So for sufficiently large samples the estimate (9) approaches the true message (3).

4.3 Connections to Non-parametric BP

Another popular technique for approximating belief propagation using sample-based messages is *nonparametric belief propagation*, or NBP (Sudderth et al., 2003). In NBP, each message is represented using a collection of samples, which are smoothed by a Gaussian kernel to ensure a well-defined product. Samples are drawn from the product of the incoming messages (the pre-message) and are propagated stochastically through $\Psi_{s,t}$ to produce samples representing the new message from s to t. These samples are again smoothed to ensure a well-defined product, and the process is repeated.

The algorithm's most computationally expensive step is sampling from the product of messages; thus, the alternative expression (6) suggests an alternative approach in which one draws a single set of samples from the product of all messages (the belief) and weights these samples by the inverse of each incoming message (Ihler et al., 2005a); we refer to this procedure as *belief-based* sampling for NBP. The default approach, in which samples are drawn for each pre-message, we refer to as *message-based* sampling.

A key difference between NBP and PBP is that in NBP the incoming messages to each node do not share their collection of samples. In other words, in NBP node s draws the samples it will use to represent its message to t, while in PBP node s uses t's samples to represent its outgoing message. This difference sidesteps the need to smooth the sample set when taking products.

[1] It is also possible to sample a set of particles $\{x_{st}^{(i)}\}$ for each *pre-message* $M_{s\to t}$ in the network from potentially different distributions $W_{st}(x_s)$, for which our analysis remains essentially unchanged. However, for notational simplicity and to be able to apply the more computationally efficient message expression described in Section 3, we use a single distribution and sample set for each node.

5 Finite Sample Analysis

Fundamentally, we are interested in particle-based approximations to belief propagation for their finite–sample behavior, i.e., we hope that a relatively small collection of samples will give rise to an accurate estimate of the beliefs. To analyze particle belief propagation's performance for finite numbers of samples, we apply a concentration bound on the estimated messages and beliefs. We use the shorthand $x \in y(1 \pm \epsilon)$ to abbreviate the upper and lower bounds $y(1-\epsilon) \leq x \leq y(1+\epsilon)$.

We begin by stating a variant of Bernstein's inequality. Consider n IID random variables $\{x_i\}$ with mean $\overline{x}$ and satisfying (with probability 1), $0 \leq x_i \leq R\ \overline{x}$. Then, with probability at least $(1-\delta)$ over the choice of values $x_1, \ldots, x_n$ we have that

$$\frac{1}{n}\sum_{i=1}^{n} x_i \ \in\ \overline{x}\ (1 \pm \epsilon(R, n, \delta)) \tag{10}$$

where

$$\epsilon(R, n, \delta) = \sqrt{\frac{R}{n}}\left(\eta + \sqrt{\eta^2 + 2\ln(2/\delta)}\right) \tag{11}$$

and

$$\eta = \frac{\ln(2/\delta)}{3}\sqrt{\frac{R}{n}}$$

Equation (11) can be derived by applying Bernstein's inequality to upper and lower intervals of size $\overline{x}|\epsilon|$ and observing that the variance of each x_i is bounded by $R\overline{x}^2$. To ensure that $\epsilon \ll 1$ (the range of primary interest) we require $n \gg R$, in which case $\epsilon \approx \sqrt{2\ln(2/\delta)R/n}$.

We now consider the following form of (7).

$$m_{t\to s}(x_s) = \mathrm{E}_{x_t \sim W_t}\left[\Psi_{s,t}(x_s, x_t)\frac{M_{t\to s}(x_t)}{W_t(x_t)}\right] \tag{12}$$

Given that we intend to apply (10) to (12), it is natural to define the constant

$$R_W = \max_{s,t\in E}\ \max_{x_s\in\mathcal{X}_s,\ x_t\in\mathcal{X}_t} \frac{\Psi_{s,t}(x_s, x_t)M_{t\to s}(x_t)}{W_t(x_t)m_{t\to s}(x_s)}$$

so that

$$\begin{aligned}\Psi_{s,t}(x_s, x_t)\frac{M_{t\to s}(x_t)}{W_t(x_t)} &\leq R_W m_{t\to s}(x_s) \\ &= R_W \mathrm{E}_{x_t\sim W_t}\left[\Psi_{s,t}(x_s, x_t)\frac{M_{t\to s}(x_t)}{W_t(x_t)}\right]\end{aligned}$$

Note that the constant R_W depends on the choice of sampling distributions W_s; we discuss this relationship further in Section 6.

We can now state our first main result.

Theorem 1. *For a tree with k nodes, if we sample n particles at each node with $n > k^2 R_W \ln(kn/\delta)$, and compute the message values defined by (8), then with probability at least $1-\delta$ over the choice of particles we have that the following holds simultaneously for all nodes s and all particles $x_s^{(i)}$.*

$$\begin{aligned}\widehat{B}_s(x_s^{(i)}) = \Psi_s(x_s^{(i)})\prod_{t\in\Gamma_s}\widehat{m}_{t\to s}^{(i)} \quad \in \\ B_s(x_s^{(i)})\left(1 \pm O\left(k\sqrt{n^{-1}R_W\ln(kn/\delta)}\right)\right)\end{aligned} \tag{13}$$

Proof. Let $\epsilon(R, n, \delta)$ be as defined in (11). We apply a union bound over all $2(k-1)$ messages and n particles evaluated by each message to (10)–(11). Then, with probability at least $(1-\delta)$ over the choice of particles at neighbors t, the following holds simultaneously for all directed edges $(t \to s)$ and particles $x_s^{(i)}$:

$$\begin{aligned}\frac{1}{n}\sum_{j=1}^{n}\Psi_{s,t}(x_s^{(i)}, x_t^{(j)})\frac{M_{t\to s}(x_t^{(j)})}{W_t(x_t^{(j)})} \in \\ m_{t\to s}(x_s^{(i)})\left(1 \pm \epsilon\left(R_W, n, \frac{\delta}{2(k-1)n}\right)\right)\end{aligned} \tag{14}$$

Now for each directed edge $(t \to s)$ we define k_{ts} to be the number of nodes connected to t (including t itself) by a path that does not go through s. Given (14), we prove the following upper and lower bounds:

$$\widehat{m}_{t\to s}^{(i)} \geq m_{t\to s}(x_s^{(i)})\left(1 - k_{ts}\epsilon\left(R_W, n, \frac{\delta}{2(k-1)n}\right)\right) \tag{15}$$

$$\widehat{m}_{t\to s}^{(i)} \leq m_{t\to s}(x_s^{(i)})\exp\left(k_{ts}\epsilon\left(R_W, n, \frac{\delta}{2(k-1)n}\right)\right) \tag{16}$$

The proof of (15)–(16) proceeds by induction on k_{ts}. If $k_{ts} = 1$, so that t is a leaf node, then $\widehat{M}_{t\to s} = \psi_t = M_{t\to s}$ and thus the outgoing message $\widehat{m}_{t\to s}^{(i)}$ is equal to the left-hand side of (14). In this case (15) is immediate and (16) follows from the fact that $1+\epsilon \leq \exp(\epsilon)$.

For the inductive step we assume that (15) and (16) hold for the messages coming into t from all nodes other than s. This means that these incoming messages can be written as the true messages multiplied by correction factors bounded by (15) and (16). Applying (14) along with the fact that $(1-d\epsilon) \leq (1-\epsilon)^d$ and $\exp(\epsilon)^d = \exp(d\epsilon)$, we prove the inductive step.

By invoking (15)–(16) on all incoming messages to s,

and using a similar argument, we have that

$$\widehat{B}_s^{(i)} \geq B_s(x_s^k)\left(1-(k-1)\epsilon\left(R_W, n, \frac{\delta}{2(k-1)n}\right)\right)$$
$$\widehat{B}_s^{(i)} \leq B_s(x_s^k)\exp\left((k-1)\epsilon\left(R_W, n, \frac{\delta}{2(k-1)n}\right)\right)$$

Finally, if we require $n > k^2 R_W \ln(kn/\delta)$ we have $\epsilon(R_W, n, \delta/(2(k-1)n)) \leq O(1/k)$, which in turn implies that $\exp(k\epsilon) \leq (1+O(k\epsilon))$. □

Theorem 1 argues that with high probability, the messages in particle belief propagation will be accurate at the particle locations themselves. Alternatively however, we need not restrict the domain of our estimated messages and beliefs at node s to the sampled values $\{x_s^{(i)}\}$. The potential functions Ψ define functions valid at any $x_s \in \mathcal{X}_s$ given samples at the neighboring nodes:

$$\widetilde{m}_{t\to s}(x_s) = \frac{1}{n}\sum_{j=1}^{n}\Psi_{t,s}(x_t^{(j)}, x_s)\frac{\Psi_t(x_t^{(j)})}{W_t(x_t^{(j)})}\prod_{u\in\Gamma_t\setminus s}\widehat{m}_{u\to t}^{(j)}$$
$$\widetilde{B}_s(x_s) = \Psi_s(x_s)\prod_{t\in\Gamma_s}\widetilde{m}_{t\to s}(x_s) \qquad (17)$$
$$\widetilde{P}_s(x_s) = \frac{1}{Z}\widetilde{B}_s(x_s)$$

Note that here we have used the true normalizing constant Z (i.e., the normalizing constant for $B_s(x_s)$) in defining $\widetilde{P}_s(x_s)$.

We can now state our second main result.

Theorem 2. *Under the same conditions as Theorem 1, with probability at least $1-\delta'$ over the choice of the particles we have for all nodes s that.*

$$\left|\left|P_s - \widetilde{P}_s\right|\right|_1 \leq O\left(\sqrt{\frac{k^3 R_W \ln(knR_W/\delta')}{n}}\right) \qquad (18)$$

Proof. The proof of Theorem 1 can be modified to show that if one fixes s and $x_s^{(i)}$ arbitrarily before drawing any samples, and then draws samples at all nodes other than s, then with probability at least $1-\delta$ over the draw of the sample at the other nodes we have the bound (13). Now consider a given $\epsilon > 0$. By setting $\delta = \exp(-\Omega(n\epsilon^2/(R_W k^2 \ln(kn))))$ into (13), we can set the width of the confidence interval to ϵ, yielding the following (where the probability P_S is over the draw of the samples at nodes other than s): $\forall x_s \in \mathcal{X}_s$,

$$\mathrm{P}_S\left[\widetilde{B}_s(x_s) \notin B_s(x_s)(1\pm\epsilon)\right] \leq \exp\left(-\Omega\left(\frac{n\epsilon^2}{R_W k^2 \ln kn}\right)\right)$$

We can then convert this probability bound into one on the two distributions' L_1 distance; intuitively, if the two functions substantially disagree only on a set of small measure, they must also be close in an L_1 sense. We begin by dividing the beliefs by Z to make the true belief a probability distribution, so that $\forall x_s \in \mathcal{X}_s$,

$$\mathrm{P}_S\left[\widetilde{P}_s(x_s) \notin P_s(x_s)(1\pm\epsilon)\right] \leq \exp\left(-\Omega\left(\frac{n\epsilon^2}{R_W k^2 \ln kn}\right)\right)$$

Taking the expected value of both sides with respect to x_s, we have

$$\mathrm{E}_{x_s\sim P_s}\left[\mathrm{P}_S\left[\widetilde{P}_s(x_s) \notin P_s(x_s)(1\pm\epsilon)\right]\right] \leq \exp\left(-\Omega\left(\frac{n\epsilon^2}{R_W k^2 \ln kn}\right)\right)$$

Noting that, for binary z, $P[z] = E[z]$, these two expectations commute and we can rewrite this as

$$\mathrm{E}_S\left[\mathrm{P}_{x_s\sim P_s}\left[\widetilde{P}_s(x_s) \notin P_s(x_s)(1\pm\epsilon)\right]\right] \leq \exp\left(-\Omega\left(\frac{n\epsilon^2}{R_W k^2 \ln kn}\right)\right)$$

and applying Markov's inequality, we obtain

$$\mathrm{P}_S\left[\mathrm{P}_{x_s\sim P_s}\left[\widetilde{P}_s(x_s) \notin P_s(x_s)(1\pm\epsilon)\right] \geq \gamma\right] \leq \frac{1}{\gamma}\exp\left(-\Omega\left(\frac{n\epsilon^2}{R_W k^2 \ln kn}\right)\right).$$

Now by taking ϵ to be $O\left(k\sqrt{R_W \ln(kn/(\gamma\delta'))/n}\right)$, we can set the right hand side to be δ'. Then with probability at least $1-\delta'$ over the choice of the sample points at nodes other than s we have,

$$\mathrm{P}_{x_s\sim P_s}\left[\widetilde{P}_s(x_s) \notin P_s(x_s)\left(1\pm O\left(k\sqrt{\frac{R_W}{n}\ln\frac{kn}{\gamma\delta'}}\right)\right)\right] \leq \gamma.$$

Because the constant R_W bounds the ratio of a message sample to its expected value (the true message), the product of incoming messages (of which there are at most k) is also bounded, so that $\widetilde{P}_s(x_s)/P_s(x_s) \leq (R_W)^k$.

$$\left|\left|P_s - \widetilde{P}_s\right|\right|_1 = \mathrm{E}_{x_s\sim P_s}\left[\left|1-\frac{\widetilde{P}_s}{P_s}\right|\right] \leq (1-\gamma)O\left(k\sqrt{\frac{R_W}{n}\ln\frac{kn}{\gamma\delta'}}\right) + \gamma(R_W)^k.$$

Setting $\gamma = 1/(n(R_W)^k)$ now gives the result for the single node s. To get the result simultaneously for all s we take a union bound over all k nodes ($\delta' \to \delta'/k$), which does not change to order of the bound. □

Although Theorems 1 and 2 are formulated for tree-structured graphs, they can also be applied (in a limited way) to loopy belief propagation. Loopy BP is analyzed in terms of its *Bethe tree*, a tree-structured "unrolling" of the graph to a depth equal to the number of iterations of loopy BP (Ihler et al., 2005b). This tree is then analyzed in the same way as before; although the random samples are correlated among nodes of the Bethe tree, the union bound still applies and the end result is unchanged. However, the potentially exponential growth of the number of nodes k in the Bethe tree causes additional complications, since our bounds depend polynomially on k.

6 Selecting Sampling Distributions and Resampling

The preceding section's analysis of finite sample accuracy is sensitive to the parameter R_W, which is itself sensitive to the choice of the sampling distributions W_s. Unfortunately, it appears difficult to optimize R_W over the choice of the sampling distributions W_s (or even compute its value) *a priori*. On the other hand, the following observations seem worth noting:

$$\begin{aligned} R_W &= \max_{s,t,x_s,x_t} \frac{M_{s\to t}(x_t)\Psi_{t,s}(x_t,x_s)}{W_t(x_t)m_{t\to s}(x_s)} \\ &= \max_{s,t,x_s,x_t} \frac{P_{s,t}(x_s,x_t)}{W_t(x_t)P_s(x_s)} \\ &= \max_{s,t,x_s,x_t} \frac{P_{s,t}(x_t|x_s)}{W_t(x_t)} \end{aligned} \tag{19}$$

So we want W_t to simultaneously match all possible conditional distributions on x_t given the value of a single neighboring node. The form of (19) suggests that, although not necessarily optimal in the sense of minimizing R_W, a good choice may be to sample from the marginal distribution: $W_t(x_t) = P_t(x_t)$. This idea was originally proposed in Koller et al. (1999), based on an intuitive description of the issues.

Unfortunately, the true marginal $P_t(x_t)$ is unavailable for sampling—indeed, this is the very quantity we are trying to compute. However, at any stage of BP we can use our current marginal estimate to construct a new sampling distribution for node t and draw a new set of particles $\{x_t^{(i)}\}$. This leads to an iterative algorithm which continues to improve its estimates as the sampling distributions become more accurately targeted. Unfortunately, such an iterative resampling process is significantly more difficult to analyze.

In Koller et al. (1999), the sampling distributions were constructed using a density estimation step (fitting mixtures of Gaussians). However, the fact that the belief estimate $\widetilde{B}_t(x_t)$ can be computed at any value of x_t allows us to use another approach, which has also been applied to particle filters with success (Doucet et al., 2001; Khan et al., 2005). By running a short MCMC simulation such as the Metropolis-Hastings algorithm, we can draw samples directly from $\widetilde{B}_t$ without it needing to be explicitly constructed or fitted. We simply apply the definition (17) to evaluate the acceptance probability of each step.

This approach manages to avoid any distributional assumptions or biases inherent in density estimation methods. One disadvantage, however, is that it can be difficult to assess convergence during MCMC sampling, and the MCMC chain is run at each iteration of BP and each node. On the other hand, many implementations of NBP also use MCMC steps at each iteration to draw samples from the message product (e.g., Sudderth et al., 2003), and thus have similar cost and convergence assessment issues.

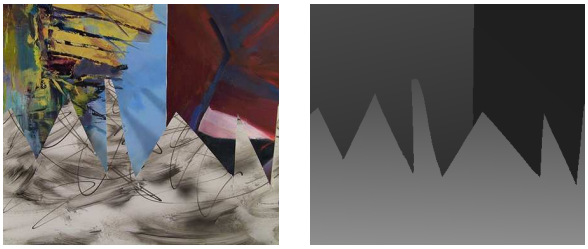
7 Comparison to Previous Results

Our results describe the consistency and accuracy of particle-based representations of belief propagation. Considerable work has gone into analyzing particle representations on Markov chains (particle filtering); see for example Del Moral (2004) for details. Like our results, the convergence behavior of particle filters is also $O(1/\sqrt{n})$, but becomes independent of the number of nodes k as $k \to \infty$.

Fundamentally, these results are based on the mixing properties of the conditional distributions. For example, Del Moral (2004) applies Dobrushin's contraction coefficient, which bounds the reduction in the total variation norm between two probability measures. The total variation is a natural distance for comparing distributions, essentially equivalent to the L_1 norm (by Scheffé's theorem). These norms are also well-behaved with respect to sampling and are thus well-suited to analysis of particle filters and density estimation.

Unfortunately, these measures are less well-suited to dealing with the product operation of BP. In work analyzing the properties of BP, the norm of choice is Hilbert's projective measure, to which generalizations of Birkhoff's contraction coefficient are applied (Ihler et al., 2005b; Mooij and Kappen, 2007). Our results are stated in terms of multiplicative error, which is equivalent to the projective norm up to first-order; see e.g. Ihler et al. (2005b), Lemma 3.

Using the projective norm complicates the precise



(a) (b)

Figure 1: "Sawtooth" example from the stereo dataset of Scharstein and Szeliski (2002). (a) Left-hand image; (b) ground truth disparity (depth) map.

statement of the bound, but does not change its qualitative form: rather than a simple additive recursion outlined in Theorem 1, the projective norm would follow the recursion outlined in Ihler et al. (2005b) Sec. 5.4. For Markov chains, and assuming some degree of mixing as measured by Ihler et al. (2005b) Eq. (7), the errors decrease at an exponential rate so that there exists a constant range $k_0(\epsilon)$ beyond which errors can be ignored without affecting the order of the bounds, and for $k \geq k_0(\epsilon)$ renders the bounds independent of k.

8 Experimental Results

To see how our theoretical results carry over into a practical setting, we evaluate the performance of particle BP on the problem of reconstructing depth (or equivalently pixel disparity) from stereo image pairs. Belief propagation was applied to the stereo vision problem in Sun et al. (2003), and since then a number of more sophisticated models have also used BP for inference (e.g., Sun et al., 2005; Klaus et al., 2006; Yang et al., 2006). These methods are quite successful; BP-based methods currently comprise half of the best ten algorithms[2] for stereo disparity estimation. We use the original model of Sun et al. (2003) and the "Sawtooth" image from Scharstein and Szeliski (2002) for our comparisons, shown in Figure 1.

Our experiments are not designed to showcase an application requiring particle BP, since there are already many such applications described in the literature (e.g., Coughlan and Ferreira, 2002; Sigal et al., 2004; Sudderth et al., 2004; Ihler et al., 2005a). Instead, our purpose is to assess the accuracy of particle BP compared to its deterministic counterpart. In the stereo model each x_s is univariate, allowing us to also create a high-quality discretized solution. We compare this solution to the estimated beliefs found via PBP and NBP with different sampling methods.

[2] See `http://vision.middlebury.edu/stereo/eval/`.

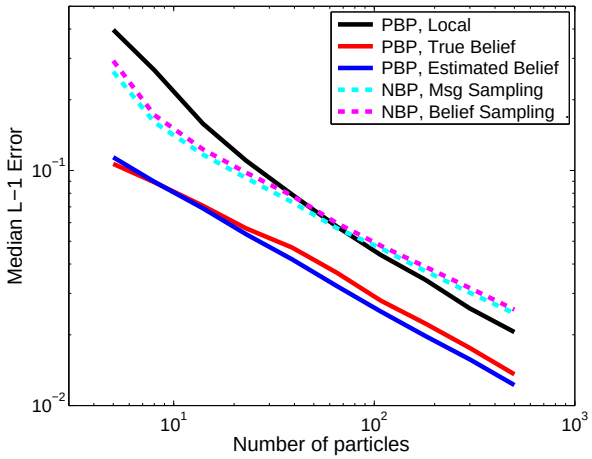


Figure 2: Log-log plot of the median L_1 error between the estimated beliefs $\widetilde{B}_s(x_s)$ and the true beliefs $B_s(x_s)$ as a function of the number of samples n used in the particle representation. We show PBP under three sampling distributions W_s: the local potentials Ψ_s, the current belief estimates $\widetilde{B}_s(x_s)$ at each iteration, and the true beliefs $B_s(x_s)$. All three decrease at a rate of $n^{-1/2}$. Also shown is NBP using message- and belief-based sampling; both have slightly higher error and appear to decrease at a slower rate corresponding to NBP's smoothing parameter.

Figure 2 shows a log-log plot of the median L_1 error between the true beliefs $B_s(x_s)$ at each pixel and the estimated beliefs $\widetilde{B}_s(x_s)$ found via PBP and NBP using various sampling distributions. For PBP, we show three sampling distributions: drawing samples from the local potentials, $W_s(x_s) = \psi_s(x_s)$; drawing samples from the *true* beliefs, $W_s(x_s) = B_s(x)$; and resampling at each iteration according to the currently estimated beliefs, $W_s(x_s) = \widetilde{B}_s(x_s)$ as described in Section 6. Note that the second option (sampling from the true beliefs) is not possible in general. Moreover, in our experiments, its performance is nearly identical to that of sampling from the estimated beliefs at each iteration. Sampling from the local potentials performed slightly less well. In all three cases, the errors decrease at a rate of $1/\sqrt{n}$.

We also compare to two versions of NBP: sampling from the messages (message-based sampling), and sampling from the beliefs and reweighting to form messages (belief-based sampling). To enable a fair comparison with PBP, for NBP we evaluate $\widetilde{B}_s(x_s)$ as in (17) using samples drawn from the pre-message product $\widehat{M}$ at the neighbors of node s. Both message- and belief-based sampling performed nearly identically on this problem. We note two things about NBP's performance: first, that the error is slightly higher than

that for PBP with belief-based sampling; and second, that the rate of decrease appears slower than $1/\sqrt{n}$. Both of these effects are likely due to the kernel-based smoothing performed on the samples when messages are constructed in NBP. This step is necessary to make NBP's message products well-defined, but biases the estimated messages to be smoother than the true messages. The Gaussian kernel's variance (smoothing parameter) used in our experiments decreases at a rate of $n^{-2/5}$, which visually matches the rate observed in NBP's L_1 error in Figure 2. Thus, avoiding the smoothing step required by NBP appears to provide a measureable improvement.

9 Summary and Conclusions

In this paper we have described a generic algorithm for sample–based or particle belief propagation in systems of variables with large or continuous domains, and showed that the algorithm is consistent, i.e., approaches the true values of the message and belief functions as the number of samples grow large. We then demonstrated a convergence rate, showing that the beliefs obtained are accurate both at the particle locations themselves, and in an L_1 sense, at a rate of $O(1/\sqrt{n})$ where n is the number of particles. Finally, we illustrated the algorithm on a stereo vision data set, showing that the algorithm's behavior in practice corresponds to the theory and comparing its performance across different sampling distributions and to two sampling approaches for NBP.

The relationship of the quantity R_W to the convergence rate highlights the importance of selecting a good sampling distribution (as in any Monte Carlo estimation process). Although it is difficult to select the optimal sampling distributions *a priori*, the form of (19) indicates that the quality of the sampling distribution can be evaluated as the inference process progresses, and that a new sampling distribution could be selected using the current marginal estimates for guidance. Although such "adaptive" choices for the sampling distribution are much more difficult to analyze, the form of R_W seems to support the notion of sampling from the current marginal estimates themselves; in our experiments this technique performed just as well as sampling from the true marginal distributions.

References

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

J. M. Coughlan and S. J. Ferreira. Finding deformable shapes using loopy belief propagation. In *ECCV*, 2002.

P. Del Moral. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer-Verlag, New York, 2004.

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

W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *Int. J. Comp. Vis.*, 40(1):25–47, 2000.

A. Ihler, J. Fisher, R. Moses, and A. Willsky. Nonparametric belief propagation for self-calibration in sensor networks. *IEEE JSAC*, pages 809–819, Apr. 2005a.

A. Ihler, J. Fisher, and A. Willsky. Loopy belief propagation: Convergence and effects of message errors. *J. Mach. Learn. Res.*, 6:905–936, May 2005b.

M. Isard. PAMPAS: Real–valued graphical models for computer vision. In *CVPR*, 2003.

Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Trans. PAMI*, pages 1805–1918, Nov. 2005.

A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR*, 2006.

D. Koller, U. Lerner, and D. Angelov. A general algorithm for approximate inference and its application to hybrid Bayes nets. In *UAI*, pages 324–333, 1999.

F. Kschischang, B. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. IT*, 47(2): 498–519, Feb. 2001.

J. M. Mooij and H. J. Kappen. Sufficient conditions for convergence of the sum-product algorithm. *IEEE Trans. IT*, 53(12):4422–4437, Dec. 2007.

R. Neal, M. Beal, and S. Roweis. Inferring state sequences for non-linear systems with embedded hidden Markov models. In *NIPS*, 2003.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufman, San Mateo, 1988.

D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comp. Vis.*, 47(1/2/3):7–42, Apr. 2002.

L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard. Tracking loose-limbed people. In *CVPR*, 2004.

E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. Nonparametric belief propagation. In *CVPR*, 2003.

E. Sudderth, M. Mandel, W. Freeman, and A. Willsky. Distributed occlusion reasoning for tracking with nonparametric belief propagation. In *NIPS*, 2004.

J. Sun, H. Y. Shum, and N. N. Zheng. Stereo matching using belief propagation. *IEEE Trans. PAMI*, 25(7):787–800, July 2003.

J. Sun, Y. Li, S. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *CVPR*, 2005.

S. Thrun, D. Fox, and W. Burgard. Monte carlo localization with mixture proposal distribution. In *AAAI*, 2000.

R. van der Merwe, N. de Freitas, A. Doucet, and E. Wan. The unscented particle filter. In *NIPS*, 2001.

Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. In *CVPR*, 2006.