Survey Propagation beyond Constraint Satisfaction Problems

Christopher Srinivasa
Probabilistic & Statistical
Inference Group
University of Toronto

Siamak Ravanbakhsh
School of Computer Science
Carnegie Mellon University

Brendan Frey
Probabilistic & Statistical
Inference Group
University of Toronto

Abstract

Survey propagation (SP) is a message passing procedure that attempts to model all the fixed points of Belief Propagation (BP), thereby improving BP's approximation in loopy graphs where BP's assumptions do not hold. For this, SP messages represent distributions over BP messages. Unfortunately this requirement makes SP intractable beyond constraint satisfaction problems because, to perform general SP updates, one has to operate on distributions over a continuous domain. We propose an approximation scheme to efficiently extend the application of SP to marginalization in binary pairwise graphical models. Our approximate SP has $\mathcal{O}(DK \log(DK)\tau)$ complexity per iteration, where τ is the complexity of BP per iteration, D is the maximum node degree and K is a resolution constant controlling the approximation's fidelity. Our experiments show that this method can track many BP fixed points, achieving a high marginalization accuracy within a few iterations, in difficult settings where BP is often non-convergent and inaccurate.

1 Introduction

Complex probabilistic models are often defined over large sets of interacting variables, and answering interesting queries about these models involves inference in the form of marginalization. If such a model factorizes into lower order interactions in the form of a hyper-tree, an efficient way to perform marginalization is to use the distributive law (Aji and McEliece, 2000). Belief Propagation (BP) is a method that achieves this goal by passing messages on a hyper-graph representing the conditional independencies between the variables. BP is exact on hyper-graphs with no loops and often yields good approximations on loopy graphs (Frey and MacKay, 1997; Murphy et al., 1999).

Loopy BP (LBP) is an iterative message update procedure, where messages are passed on the edges of the graph until they converge to a fixed point (Yedidia et al., 2000; Heskes, 2003). For a loopy graph, many such LBP fixed points could exist. Survey Propagation (SP) attempts to improve BP's approximation by capturing all such fixed points and weighing each one by the corresponding Bethe partition function (Mezard and Montanari, 2009). SP is also powered by the distributive law (Ravanbakhsh and Greiner, 2014) and takes the form of a message exchange on a hyper-graph. Today, SP and its potential in the field of Artificial Intelligence (AI) remains obscure, and its analysis and extensions have been confined to Constraint Satisfaction Problems (CSPs; *e.g.*, Kroc et al., 2007; Maneva et al., 2007; Ravanbakhsh and Greiner, 2015). This is partly due to the intractability of SP in its general form. We introduce a method to extend the application of SP to the Ising model – *i.e.*, binary Markov networks.

After reviewing BP and SP in Section 2, Section 3 describes an efficient approximation to SP for the Ising model. Section 4 showcases a variety of Ising models where our approximation outperforms BP and it further provides intuition into the behaviour of SP in practice. We hope our first attempt at applying SP in this setting will open doors to its applications in AI and machine learning. Section 5 outlines some of these directions for future work.

2 Background

Given a function $F(\underline{x})$ where $\underline{x} = \{x_1, ..., x_N\}$ and w.l.o.g. each x_i takes one of M possible values, many

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51. Copyright 2016 by the authors.

queries that characterize the behaviour of F with regards to a subset of its variables $\alpha \subseteq \{1, \ldots, N\}$, are answered by marginalizing $\tilde{p}_\alpha(\underline{x}_\alpha) = \sum_{\underline{x}_{\backslash\alpha}} F(\underline{x})$, where $\backslash\alpha$ denotes the set of variables excluding α. Computationally, this operation can be very expensive. For example, if we wish to marginalize over all but one variable x_i (i.e. $\alpha = i$), the cost of marginalization is $\mathcal{O}\left(M^{N-1}\right)$).

If $F(\underline{x})$ factorizes as a combination of local functions $f_I(\underline{x}_I)$, it can be visualized using a factor graph (Kschischang et al., 2001). This graph is bipartite where each node in one part represents a local function $\{f_I\}_I$ (a.k.a. a factor) and each node in other part represents a variable $\{x_i\}_i$. Each variable is connected via an edge to each factor in which it appears.

2.1 Belief Propagation

If $F(\underline{x})$ factorizes as a product of local functions – *i.e.*,

$$F(\underline{x}) = \prod_I f_I(\underline{x}_I) \tag{1}$$

then the original marginalization problem can be solved using the sum-product algorithm (a.k.a. BP). BP relies on the sum and product operations satisfying the distributive law (Aji and McEliece, 2000): $ab+ac = a(b+c)$. Using this property, marginalization is efficiently performed by first applying summation to the factors f_I (rather than the entire function $F(\underline{x})$) and then combining the results via the product operation. This can be viewed as passing messages along the edges of a factor graph where the sum-product result of each subsection of the graph is a message.

For any factor graph, where ∂i denotes all factors neighbouring variable i and ∂I denotes all variables neighbouring factor I, two kinds of BP message updates exist (note that we also assign a functional $\widetilde{P}(\cdot)$ to each update). The message from factor I to variables i

$$\begin{aligned}\tilde{p}_{I\to i}(x_i) &= \sum_{\underline{x}_{\backslash i}} f_I(\underline{x}_I) \prod_{j \in \partial I \backslash i} \hat{p}_{j\to I}(x_j) \\ &\overset{\text{def}}{=} \widetilde{P}_{I\to i}(\underline{\hat{p}}_{\partial I\backslash i\to I})(x_i)\end{aligned}$$

where $\partial I \setminus i$ denotes the set of all variables neighbouring factor I, minus the variable indexed by i. Here, $\underline{\hat{p}}_{\partial I\backslash i\to I}$ is short form for the set $\{\hat{p}_{j\to I}\}_{j\in\partial I\backslash i}$, where we use underline to distinguish sets (or tuples).

Likewise, messages from variables to factors $p_{i\to I}$ are

$$\tilde{p}_{i\to I}(x_i) = \prod_{J\in\partial i\backslash I} \hat{p}_{J\to i}(x_i) \overset{\text{def}}{=} \widetilde{P}_{i\to I}(\underline{\hat{p}}_{\partial i\backslash I\to i})(x_i).$$

With these messages, we can compute the marginal at any variable as the product of all incoming messages

$$\tilde{p}_i(x_i) = \prod_{I\in\partial i} \hat{p}_{I\to i}(x_i) \overset{\text{def}}{=} \widetilde{P}_i(\underline{\hat{p}}_{\partial i\to i})(x_i)$$

and similarly the marginal at any factor is the product of that factor with all incoming messages – *i.e.*,

$$\tilde{p}_I(\underline{x}_I) = f_I(\underline{x}_I) \prod_{i\in\partial I} \underline{\hat{p}}_{i\to I}(x_i) \overset{\text{def}}{=} \widetilde{P}_I(\underline{\hat{p}}_{\partial I\to I})(\underline{x}_I).$$

For any of the above quantities, $\hat{p}$ represents the normalized version of $\tilde{p}$ (*e.g.*, $\hat{p}(x_i) = \tilde{p}(x_i)/\tilde{p}(\emptyset)$), where $\tilde{p}(\emptyset) \overset{\text{def}}{=} \sum_{x_i} \tilde{p}(x_i)$. As can be seen from the equations, each node requires all incoming messages to compute its marginal. When the graph is a tree, this is done by choosing an arbitrary root node, passing messages to it from the leaves, and then back down to the leaves. When the graph has loops, one way by which we can still perform message passing is by initializing all messages in the graph, sending them around the graph until convergence, and then using them to compute the marginals. This process is sometimes referred to as Loopy BP (LBP) to distinguish it from BP on trees.

While LBP is known to give good results, it does not have any guarantees and sometimes does not converge at all or simply converges to an incorrect answer (Frey and MacKay, 1997; Murphy et al., 1999; Weiss, 2000). One reason is that the set of messages at any iteration in loopy BP is computed from the set at the previous iteration, making LBP sensitive to the set of initial messages. Another is that LBP's Bethe approximation to the true partition function can be inaccurate. This has motivated many to improve on the quality of LBP's approximation. Most prominent classes are: a) using region-based techniques (Pelizzola, 2005; Yedida et al., 2005); b) convex and convergent alternatives (Wainwright et al., 2005; Yuille, 2002; Heskes, 2006); c) loop series (Chertkov and Chernyak, 2006; Gómez et al., 2006); d) loop correction methods (Montanari and Rizzo, 2005; Mooij and Kappen, 2007) and e) cut-set conditioning techniques (Pearl, 2014; Darwiche, 2001).

However, in practice, one often observes that when LBP converges, the quality of its results are better than many of its convex/convergent variants. Alternatively, region-based, loop correction techniques and cut-set conditioning techniques could have an exponential cost (in the size of regions, number of nodes/edges, maximum degrees and cut-set size respectively) and therefore none of these alternatives has substituted LBP in practice. Interestingly, we see that by tracking all BP fixed points, our approximate SP is convergent in many settings where BP fails to converge and has a polynomial time complexity.

2.2 Survey Propagation

Initially, SP was used to solve K-SAT problems where the variables are binary (Braunstein et al., 2005b; Mezard et al., 2002). It was later extended to CSPs with non-binary variable alphabets such as the graph coloring problem (Braunstein et al., 2005a, 2003). In all these problems, all the factors in the graph represent hard constraints, so that the BP (over Boolean semiring, a.k.a. warning propagation) messages assume only a finite number of values and the SP messages that are essentially distributions over BP messages remain tractable. This property is lost when the factors are no longer hard constraints. We aim to develop SP for applications beyond CSP by addressing the new algorithmic issues which arise in this setting. We provide an overview of SP before introducing our approximation scheme in Section 3.

When the messages in LBP converge to a fixed point, they represent a set of messages $\underline{\hat{p}}$ that simultaneously satisfy all BP update equations outlined in the previous section. Any such fixed point provides an approximation to the partition function (a.k.a. Bethe approximation)

$$G(\underline{\hat{p}})(\emptyset) = \prod_I \tilde{P}_I(\underline{\hat{p}}_{\partial I \to I})(\emptyset) \prod_i \tilde{P}_i(\underline{\hat{p}}_{\partial i \to i})(\emptyset) \left(\prod_{i, I \in \partial i} \tilde{P}_{i \leftrightarrow I}(\hat{p}_{i \to I}, \hat{p}_{I \to i})(\emptyset) \right)^{-1} \qquad (2)$$

where $\tilde{P}_{i \leftrightarrow I}(\hat{p}_{i \to I}, \hat{p}_{I \to i})(x_i) \overset{\text{def}}{=} \hat{p}_{i \to I}(x_i) \hat{p}_{I \to i}(x_i)$ (see Ravanbakhsh and Greiner, 2014).

A factor graph with loops can have many BP fixed points and, from a variational perspective, each stable fixed point represents a local optimum of the Bethe approximation to the free energy (Heskes, 2003). Note that each BP fixed point also represents a joint distribution as a product of BP marginals

$$\hat{p}(\underline{x}) \propto \prod_I \tilde{p}_I(\underline{x}_I) \prod_i \tilde{p}_i(x_i)^{1 - |\partial i|}. \qquad (3)$$

Now, assuming that the joint distribution of eq. (1) is approximated by the "sum" of all such BP joint forms – *i.e.*, $p(\underline{x}) \approx \sum_{\hat{p}} \hat{p}(\underline{x})$, SP attempts to track all such fixed points. Thus in SP, a marginalization problem can now be viewed as marginalizing once over the variables of interest for a particular fixed point, and then a second time over all BP fixed points – that is the SP approximation to the partition function is $\sum_{\underline{\hat{p}}} G_{(}\underline{\hat{p}})(\emptyset)$.

The settings satisfying these SP assumptions are characterized as a phase in 1st order Replica Symmetry Breaking (1RSB; see Mezard and Montanari, 2009), known as dynamical or clustering phase. The upshot is that the Gibbs measure [1] decomposes into a set of (exponentially) many sub-measures of roughly equal size. Here, our objective is to show that approximations to SP can indeed operate on a wide range of problems where BP fails, even if the validity of SP's assumptions may be hard to verify.

Here, following Ravanbakhsh and Greiner (2014), we quickly demonstrate the elegant way in which SP tracks (possibly exponentially large number of) BP fixed points. From the definition of G in eq. (2), we see that it can be expressed as the product of three types of factors: 1) Variable-type factors containing the component of G for each variable node in the original factor graph $\tilde{P}_i(\underline{\hat{p}}_{\partial i \to i})(\emptyset)$; 2) Factor-type factors containing the component of G for each factor node in the original factor graph $\tilde{P}_I(\underline{\hat{p}}_{\partial I \to I})(\emptyset)$; and lastly, 3) edge-type factors containing the component of G for each edge in the original factor graph $\tilde{P}_{i \leftrightarrow I}(\hat{p}_{i \to I}, \hat{p}_{I \to i})(\emptyset)$. The variables in G are the messages in the original factor graph. Figure 1 shows the resulting SP factor graph for a portion of its BP counterpart.

Since + distributes over ×, the *sum-product* algorithm can be used on the SP factor graph to obtain the *sum* $\sum_{\underline{\hat{p}}} G(\underline{\hat{p}})(\emptyset)$, where G is the *product* of the three factor types. Using indicator functions $\mathbf{1}(.)$ to guarantee that the BP updates are satisfied (*i.e.*, that we only consider BP fixed points), sum-product message passing on the SP factor graph can be reduced to two message updates: 1) message from variable-type factors to factor-type factors

$$S_{i \to I}(\hat{p}_{i \to I}) \propto \sum_{\underline{\hat{p}}_{\partial i \setminus I \to i}} \left(\mathbf{1}\left(\hat{p}_{i \to I} = P_{i \to I}(\underline{\hat{p}}_{\partial i \setminus I \to i}) \right) P_{i \to I}(\underline{\hat{p}}_{\partial i \setminus I \to i})(\emptyset) \prod_{J \in \partial i \setminus I} S_{J \to i} \ (\hat{p}_{J \to i}) \right) \qquad (4)$$

and 2) message from factor-type factors to variable-type factors

$$S_{I \to i}(\hat{p}_{I \to i}) \propto \sum_{\underline{\hat{p}}_{\partial I \setminus i \to I}} \left(\mathbf{1}\left(\hat{p}_{I \to i} = P_{I \to i}(\underline{\hat{p}}_{\partial I \setminus i \to I}) \right) P_{I \to i}(\underline{\hat{p}}_{\partial I \setminus i \to I})(\emptyset) \prod_{j \in \partial I \setminus i} S_{j \to I} \ (\hat{p}_{j \to I}) \right). \qquad (5)$$

[1] Statistical physicists tend to study these phenomena for ensembles and at the thermodynamic limit, where $N \to \infty$. Gibbs measure (Georgii, 2011) is the equivalent of the joint distribution $p(\underline{x})$ in such infinite random fields.

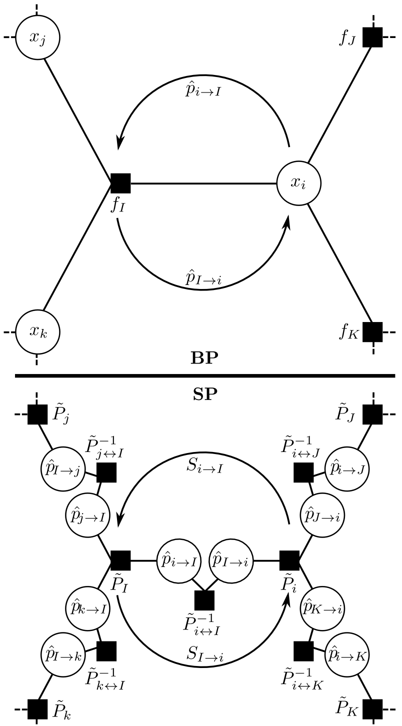


Figure 1: BP vs. SP

The SP marginal at a variable-type factor is given by

$$S_i(\hat{p}_i) \propto \sum_{\underline{\hat{p}}_{\partial i \to i}} \Bigg(\mathbf{1}\left(\hat{p}_i = P_i(\underline{\hat{p}}_{\partial i \to i})\right) P_i(\underline{\hat{p}}_{\partial i \to i})(\emptyset)$$

$$\prod_{I \in \partial i} S_{I \to i}\ (\hat{p}_{I \to i}) \Bigg) \tag{6}$$

and at a factor-type factor by

$$S_I(\hat{p}_I) \propto \sum_{\underline{\hat{p}}_{\partial I \to I}} \Bigg(\mathbf{1}\left(\hat{p}_I = P_I(\underline{\hat{p}}_{\partial I \to I})\right) P_I(\underline{\hat{p}}_{\partial I \to I})(\emptyset)$$

$$\prod_{i \in \partial I} S_{i \to I}\ (\hat{p}_{i \to I}) \Bigg). \tag{7}$$

Note that each of these SP marginals is a distribution over all possible values for a BP marginal at either a variable or factor node of the original factor graph, respectively. Looking at these expressions, the probability of each possible message or marginal value is proportional to the number of "locally consistent" BP fixed points in which it takes part, where each message combination is weighted by its contribution to the corresponding Bethe partition function G.

3 Approximation Scheme

We now describe an efficient SP implementation for marginalization on binary pairwise factor graphs. For marginalization, if there are no hard constraints, any factor graph can be transformed to a pairwise binary model (Eaton and Ghahramani, 2013). The factors in any such factor graph can then be transformed to so-called Ising factors. The Ising model defines $p(\underline{x}) \propto \prod_{ij} e^{x_i\, J_{ij}\, x_j} \prod_i e^{x_i\, \theta_i}$, where $x_i \in \{-1, 1\}$. This joint form as a product of two types of factors: local factors $f_i(x_i) = e^{x_i \theta_i}$, and pairwise factors $f_{ij}(x_i, x_j) = e^{x_i x_j J_{ij}}$.

Figure 2 shows a portion of the BP and SP factor graphs for the Ising model. For an N-variable Ising model where the degree of each variable node is D, one iteration of LBP is $\mathcal{O}(ND)$ or the number of edges in the Ising model.

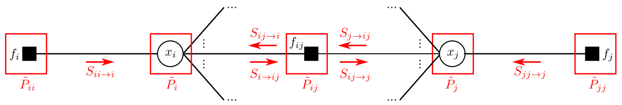


Figure 2: Portion of BP and SP factor graphs for the Ising model: BP factor graph is denoted in black and corresponding SP factors and messages are in red.

We now highlight the challenges in deriving efficient SP for marginalization on the Ising model when starting from the generic SP messages in Section 2. (In the following, we refer to variable-type factors in the SP factor graph as variables and factor-type-factors as factors)

Quantization. Since the variables are binary, each BP message can be represented by a scalar, the ratio $p(x_i = 1)/p(x_i = -1)$. However, this ratio is real-valued which implies that the SP messages would be continuous distributions over the domain $[0, +\infty)$. A first attempt is to quantize this continuous space (where ∞ is replaced by a sufficiently large number) into K linearly spaced bins where each bin index k represents the ratio $b_k = p(x_i = 1)/p(x_i = -1)$ of BP message values for a particular BP message.

Representing SP message. Sending out an SP message from a variable with degree D involves the combination of $(D-1)+1$ (*i.e.*, the $+1$ coming from the local factor) incoming SP messages where each message has length K. Calculating eq. (4) is therefore $\mathcal{O}(K^D)$, as one has to consider all combination of bins (BP messages) from all incoming SP messages. Repeating this for all D outgoing SP messages from each variable is $\mathcal{O}(DK^D)$.

Logarithmic quantization & convolution. To improve this exponential time-complexity, we change the quantization such that each bin index k now represents the log-ratio $b_k = \ln(p(x_i = 1)/p(x_i = -1))$ of BP message values. Therefore, $p(x_i = -1) = 1 - \sigma(k)$

and $p(x_i = 1) = \sigma(k)$ where $\sigma(k) = e^{b_k}/(e^{b_k}+1)$, and the center bin represents a log-ratio of 0. By working in the log-message space for BP, computing BP marginals and passing BP messages through variables now involves "summation" of the log-messages rather than "multiplication" of the original ones. For SP, this enables convolution of the SP messages at the variables.

Using Fast Fourier Transform (FFT). Rather than convolving SP messages at the variables, we go one step further and track their frequency transforms and multiply those. At every iteration, we send out messages from all variables by *multiplying* the frequency transforms of all incoming SP messages at each variable, which yields the *SP marginals* in the frequency domain, and then *divide* out the incoming frequency domain SP message on the edge for which we want to send out the message. Sending out a message from a variable with degree D involves the convolution of $(D-1)+1$ incoming SP messages of length K. In the frequency domain this implies that the transforms of the SP messages must have minimum length of $D(K-1)+1$ to avoid *aliasing.* Thus, sending out all messages from a single variable in the frequency domain involves multiplying $D+1$ incoming messages of length $D(K-1)+1$ then dividing out D of them, one at a time, resulting in a complexity of $\mathcal{O}(D^2K)$. Sending out messages for all variables is thus $\mathcal{O}(ND^2K)$.

Variable-to-factor message. Algorithm 1 summarizes the variable to factor subroutine. Here, the tuple $\underline{S}_{ij\to i} = (S_{ij\to i}(b_1), \ldots, S_{ij\to i}(b_K))$ denotes a SP message of length K and the tuple $\underline{\sigma} = (\sigma(1), \ldots, \sigma(K))$ contains the normalized BP probabilities for all K bins. Here, the product of two tuples (*e.g.*, $\underline{\sigma}\,\underline{S}_{ij\to i}$) is their element-wise product. $\mathcal{F}(.)$ and $\mathcal{F}^{-1}(.)$ represent taking *frequency* and *inverse frequency* transforms respectively. The result of $\mathcal{F}^{-1}(.)$ has length $D(K-1)+1$ since the frequency transforms of the SP messages also have the same length. We reduce it to K bins by making an approximation where quantities assigned beyond the two most extreme bins are grouped into those two bins.

Factor-to-variable message. The first step in passing a $D(K-1)+1$-length frequency domain SP message through a factor is taking its inverse frequency transform which is $\mathcal{O}(DK\log(DK))$ using FFT. The main procedure involves a one-to-one mapping of the bin indices of the incoming SP message (of length K) onto bin indices of the SP message leaving the factor, and therefore requires K steps. We then take the frequency transform of the result which also has complexity $\mathcal{O}(DK\log(DK))$ to get a frequency domain result of length $D(K-1)+1$. Combining these steps, the complexity of passing SP messages through all $\mathcal{O}(ND)$ pairwise factors is $\mathcal{O}(ND^2K\log(DK))$.

Algorithm 1: SP variable to factor subroutine

```
Input: incoming messages $\{\underline{S}_{ij\to i}\}_{j\in\partial i}$ to node $i$
Output: outgoing messages $\{\underline{S}_{i\to ij}\}_{j\in\partial i}$ from node $i$
for $ij \in \partial i$ do
    $\underline{S}^0_{ij\to i} \leftarrow \mathcal{F}\left((1-\underline{\sigma})\underline{S}_{ij\to i}\right)$
    $\underline{S}^1_{ij\to i} \leftarrow \mathcal{F}\left(\underline{\sigma}\,\underline{S}_{ij\to i}\right)$
end
$\underline{S}^0_i \leftarrow \prod_{ij\in\partial i} \underline{S}^0_{ij\to i}$
$\underline{S}^1_i \leftarrow \prod_{ij\in\partial i} \underline{S}^1_{ij\to i}$
for $ij \in \partial i$ do
    $\underline{S}_{i\to ij} \leftarrow \mathcal{F}^{-1}\left(\underline{S}^0_i/\underline{S}^0_{ij\to i} + \underline{S}^1_i/\underline{S}^1_{ij\to i}\right)$
end
```

Algorithm 2 shows the resulting factor to variable subroutine. Here, the operation $2\tanh^{-1}\Big(\tanh(J_{ij})\tanh\left(\frac{b_{k_{i\to ij}}}{2}\right)\Big)$ is the BP factor to variable update for the Ising model, when using log-ratios. The function bin : $\Re \to \{1, \ldots, K\}$ finds the nearest bin index k for a given log-ratio.

Algorithm 2: SP factor to variable subroutine

```
Input: incoming message $\underline{S}_{i\to ij}$ to the factor $ij$
Output: outgoing messages $\underline{S}_{ij\to j}$ from the factor $ij$
for $k \in \{1, \ldots, K\}$ do
    $S_{ij\to j}(k) \leftarrow 0$
    for $k' \in \{1, \ldots, K\}$ do
        if
        $k = \mathrm{bin}\Big(2\tanh^{-1}\Big(\tanh(J_{ij})\tanh\left(\frac{b_{k'}}{2}\right)\Big)\Big)$
        then
            $S_{ij\to j}(k) \leftarrow S_{ij\to j}(k) + S_{i\to ij}(k')$
        end
    end
end
```

General complexity. Adding the complexity of sending all SP messages from the variables to the complexity of then passing them through all factors, we see that the overall algorithm complexity is $\mathcal{O}(ND^2K\log(DK))$ – *i.e.*, polynomial in D, N, and K. This shows that SP is scalable to large instances in the number of variables, bins, or the maximum degree of a variable.

Algorithm 3 summarizes this general algorithm. Note that in the initialization step of this algorithm, $2\theta_i$ is the log-ratio of $h_i(x_i) = e^{\theta_i x_i}$, representing the local factor to variable BP update. Since this quantity remains static throughout the iterations of LBP, so will its corresponding SP message during SP iterations.

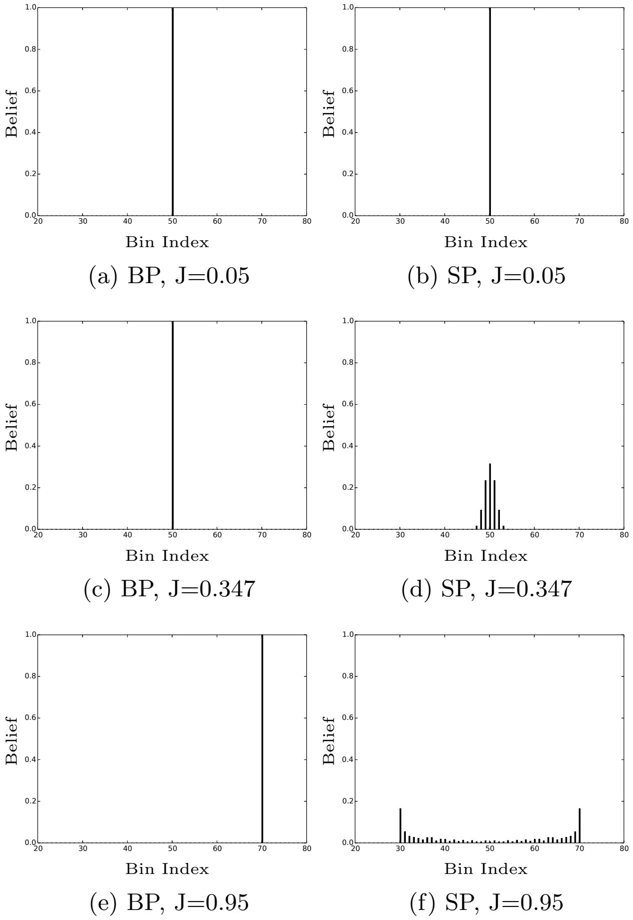
```
Algorithm 3: Approximate SP for the Ising Model
Input: The Ising model $\{J_{i,j}\}_{i,j}$, $\{h_i\}_i$; fidelity $K$
Output: Approximate SP marginals over BP
        marginals $\underline{S}_i$
Initialize $\underline{S}_{ij\to j}$   $\forall ij$
for $i \in \{1,\ldots,N\}$ do
    for $k \in \{1,\ldots,K\}$ do
        Fix $S_{ii\to i}(k) \leftarrow \mathbf{1}\big(k = \mathrm{bin}(2\theta_i)\big)$
    end
    Fix $\underline{S}^0_{ii\to i} \leftarrow \mathcal{F}\left((1-\boldsymbol{\sigma})\underline{S}_{ii\to i}\right)$
    Fix $\underline{S}^1_{ii\to i} \leftarrow \mathcal{F}\left(\boldsymbol{\sigma}\underline{S}_{ii\to i}\right)$
end
while not converged do
    for $i \in \{1,\ldots,N\}$ do
        Run SP variable-to-factor algorithm 1
    end
    for $i, j \neq i$ with $J_{ij} \neq 0$ do
        Run SP factor-to-variable algorithm 2
    end
end
for $i \in \{1,\ldots,N\}$ do
    $\underline{S}_i \leftarrow \mathcal{F}^{-1}\left(\underline{S}^0_i + \underline{S}^1_i\right)$
end
```

3.1 Tracking Many Fixed Points

To show the capabilities of SP, we test our algorithm on the attractive homogeneous Ising model where local fields θ_i are set to zero for all variables and the coupling strengths J_{ij} are all set to the same value $J > 0$.

When running LBP on this model, a phase transition exists as J increases (Mezard and Montanari, 2009). Before this phase transition, LBP has a single fixed point. After the phase transition, it has two and picks the one closest to the BP messages at initialization.

For a D-regular Ising model with $N = 1000$ variables and degree $D = 4$, the phase transition occurs roughly at $J = 0.347$. We run LBP and our approximate SP algorithm on this model for this critical value of J as well as values below and above it. For SP we use 101 bins where the middle bin represents a log-ratio of zero. Here, leftmost bin 1 has a log-ratio of 0.0098 and rightmost bin 101 has a log-ratio of 0.9902. We plot one of the SP and BP messages for each value of J. Each BP message is expressed in the SP message space by taking its value and binning it into one of the 101 SP bins. From fig. 3, we see that before the phase transition the SP message also consists of a delta function, only one fixed point exists, and *SP emulates BP*. Here, the value on which the delta is placed corresponds to the fixed-point message produced by BP. SP begins to provide useful information when its message hedges its bets on more than one value, as can be seen at the phase transition. This indicates that more than one BP fixed point exists and that SP is accounting for them all. This is clearer after the transition, where the SP message captures both fixed points. Note that the non-zero bins between the two fixed points is because the fixed points overlap to a certain degree. The assumption with SP is that the fixed points are disjoint which could only be valid if the number of variables tends to infinity. This overlap (i.e. the residue) diminishes as we increase the size.



(a) BP, J=0.05

(b) SP, J=0.05

(c) BP, J=0.347

(d) SP, J=0.347

(e) BP, J=0.95

(f) SP, J=0.95

Figure 3: SP and BP messages for attractive homogeneous Ising model (N = 1000), before (1st row), during (middle row) and after (bottom row) phase transition.

4 Experiments

We can use SP marginals, defined over BP marginals, to obtain an average marginal at each variable i as $\mathbb{E}[p(x_i = 1)] = \sum_{k=1}^{K} \sigma(k) S_i(k)$. This average is implicitly weighted by the sum of Bethe partition function of BP fixed points with a particular marginal $\sigma(k)$.

We compare these averaged SP marginals to that of BP and Gibbs sampling for many graph types. For each type, we test on three types of pairwise couplings: attractive, mixed, and repulsive (attractive and repulsive coupling results are presented in the supplementary material).

All couplings are sampled from a uniform distribution. Attractive couplings are sampled in the range $[0, \beta]$, mixed from the range $[-\beta, \beta]$, and repulsive from the range $[-\beta, 0]$ where β ranges from 0 to 2. Local fields θ_i are always sampled uniformly in the range $[-0.05, 0.05]$. For each graph and coupling type, we run three versions of SP: SP with 11 bins, 101 bins, and 1001 bins. BP messages are initialized to be uniform while SP messages are initialized randomly. All message passing algorithms do parallel message updates and run for a maximum 1000 iterations or until convergence, whichever comes first. Gibbs sampling is run for 100,000 iterations where a sample is recorded after each 100 iterations. We repeat the experiments for 10 trials, each on a new instance, at each β value.

We limit the graphs' sizes so that exact marginals can be obtained via the junction tree algorithm (Lauritzen and Spiegelhalter, 1988). However, as per our earlier analysis, this inference scheme can easily scale up, possibly to graphs up to millions of edges. At each value of β, each **column** of fig. 4 shows: *1st)* the mean absolute error of the marginals for all methods; *2nd)* the mean SP marginal entropy for the SP algorithms; *3rd)* the mean number of iterations for BP and SP. libDAI (Mooij, 2010) is used for BP, Gibbs sampling, and the junction tree method.

Each **row** in fig. 4 is a graph type: *1st)* 10 variable, fully connected; *2nd)* 40 variable, bipartite with 20 variables blocks; *3rd)* 100 variable 3-regular; *4th)* 100 variable, 10x10 grid with periodic boundary; *5th)* 1000 variable, 10x100 grid without periodic boundary.

Analysis. In all experiments, SP outperforms BP at non-trivial temperatures (see fig. 4, first column and supplementary material). SP achieves this accuracy, by converging after only a few iterations (fig. 4, last column). In fact SP's average marginal error curves resemble that of Gibbs sampling using a large number of (*i.e.*, 10^5) iterations. The average SP marginal entropy (fig. 4, 2nd column) starts at zero at trivial temperatures, suggesting that BP indeed has a single fixed point and increases for lower temperature (larger β) values.

For the fully connected and bipartite graphs (first two rows of fig. 4), the entropy points at a phase transition where it rises and then drops. After it drops, at β=2, the SP marginals show two distinct modes per marginal. Here, in contrast to the attractive homogeneous instance of section 3.1 with two fixed points, we hypothesize that these two modes identify a large number of BP fixed points, where the BP joint form of eq. (3) is effectively choosing one of the two modes at each variable, producing incorrect marginals. While BP still converges to one of its large number of modes for these instances (see final column), for the 3-regular graph and both grid graphs (rows 3-5) BP has trouble converging at all. For these graphs (*i.e.*, the 3-regular and both grid graphs), SP beliefs are spread across many SP bins and the entropy climbs as β increases.

5 Discussion and Future work

We introduce the first SP approximation scheme for inference beyond CSPs. Our scheme uses FFTs to scale gracefully with the number of variables and edges in the model. Our extensive experiments show that it converges to accurate posterior marginals within few iterations. We believe this opens the door for its application and extension in a variety of settings.

Central to the application of inference in machine learning is its role in maximum likelihood learning. However, due to non-convergence and inaccuracy of message passing techniques, Markov Chain Monte Carlo (MCMC) methods such as Gibbs sampling are the prominent methods in training models such as Boltzmann Machines. We hope that our scheme could become a fast alternative to MCMC for such models.

Another direction is in modeling and marginalization for ensembles. For example, if we wish to de-noise over the set of all possible corruptions of an image (Bishop, 2006) by a particular noise distribution (where the noise distribution may be different for each pixel), we could do so with SP by placing a prior over each local factor representing the noise distribution. This idea is closely related to an algorithm called *density evolution* used for error-correcting codes (Richardson and Urbanke, 2001). Density evolution operates by leveraging the symmetry in the error correcting codes to show that the density evolution variable-to-factor message updates are the same for all variables and, likewise, for all factor-to-variable messages, providing a closed form for the ensemble. SP can be thought of as a generalization of density evolution where the Bethe partition function of each fixed point is tracked. This is important if the goal is to infer marginals. Another method along the same lines is density propagation of Ermon et al. (2012) that uses message passing with convolution to estimate the number of configurations $\underline{x}$ with a probability $p(\underline{x})$.

Lastly, we are studying extensions of our scheme to perform more accurate max-sum inference. Chieu et al. (2007) propose a method that uses SP for Max-

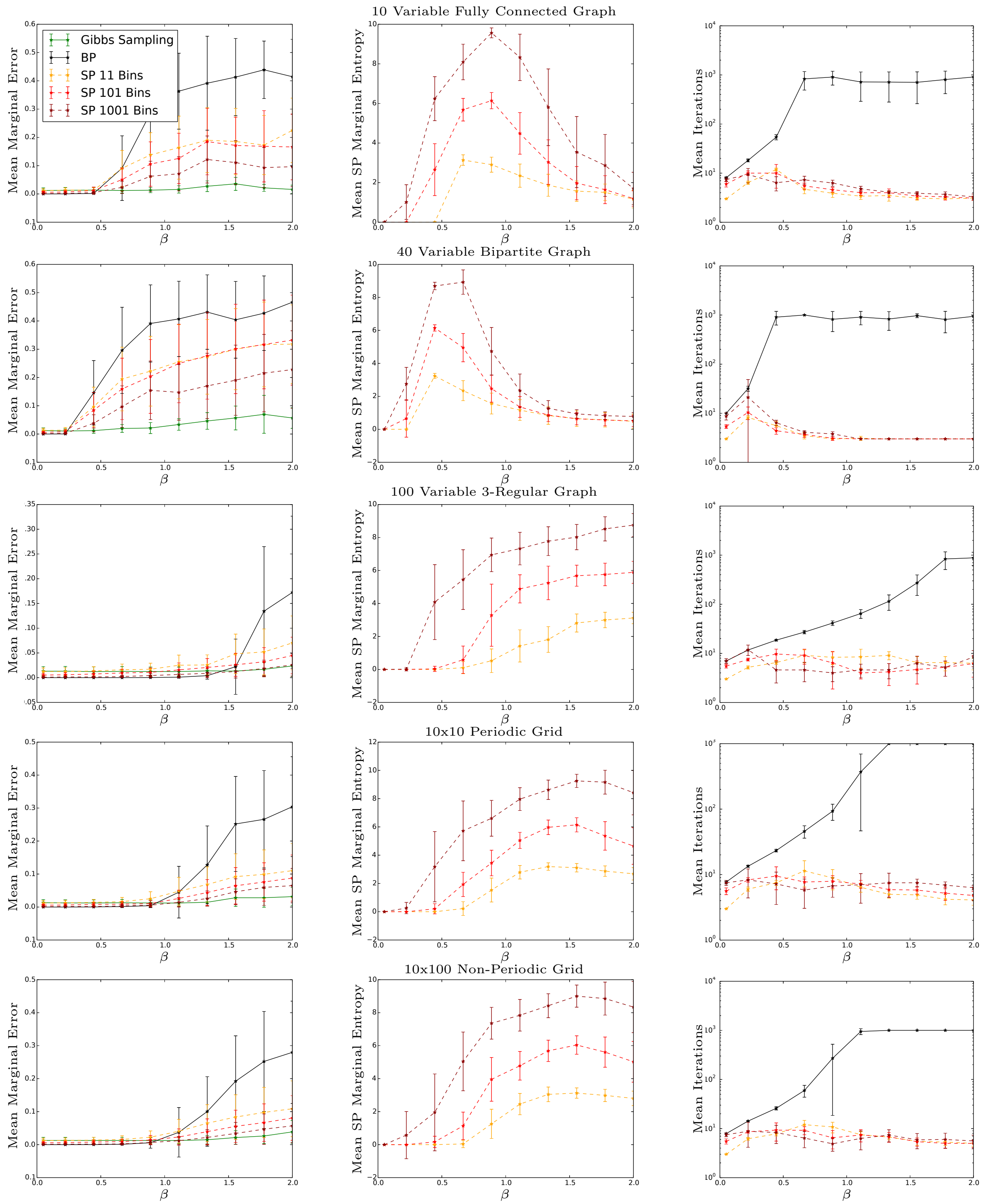


Figure 4: Results for mixed coupling. Green: Gibbs sampling, black: BP, dashed orange: SP 11 bins, dashed red: SP 101 bins, and dashed dark red: SP 1001 bins.

imum A Posteriori (MAP) inference. Their method relies on transferring the problem to a CSP and applying a parameterized version of SP (Maneva et al., 2007) to perform inference by proxy. However, direct use of SP for MAP, similar to our scheme is feasible, where the convolution is replaced with max-convolution, the zero temperature Bethe partition function represents the value of MAP assignment, and max-product SP tries to find the BP fixed point with the maximum approximate MAP value.

References

Srinivas M. Aji and Robert J. McEliece. The generalized distributive law. *IEEE Transactions on Information Theory*, 46:325–343, 2000.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., 2006.

Alfredo Braunstein, Roberto Mulet, Andrea Pagnan, Martin Weigt, and Riccardo Zecchina. Polynomial iterative algorithms for coloring and analyzing random graphs. *Physical Review E*, 68:36702, 2003.

Alfredo Braunstein, Marc Mézard, Martin Weigt, and Riccardo Zecchina. Constraint satisfaction by survey propagation. *Computational Complexity and Statistical Physics*, page 107, 2005a.

Alfredo Braunstein, Marc Mezard, and Riccardo Zecchina. Survey propagation: An algorithm for satisfiability. *Random Structures and Algorithms*, 27:201–226, 2005b.

Michael Chertkov and Vladimir Y Chernyak. Loop series for discrete statistical models on graphs. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(06):P06009, 2006.

Hai L Chieu, Wee S Lee, and Yee W Teh. Cooled and relaxed survey propagation for MRFs. In *Advances in Neural Information Processing Systems*, pages 297–304, 2007.

Adnan Darwiche. Recursive conditioning. *Artificial Intelligence*, 126(1):5–41, 2001.

Frederik Eaton and Zoubin Ghahramani. Model reductions for inference: Generality of pairwise, binary, and planar factor graphs. *Neural Computation*, 25: 1213–1260, 2013.

Stefano Ermon, Ashish Sabharwal, Bart Selman, and Carla P Gomes. Density propagation and improved bounds on the partition function. In *Advances in Neural Information Processing Systems*, pages 2762–2770, 2012.

Brendan J Frey and David J C MacKay. A revolution: Belief propagation in graphs with cycles. In *Advances in Neural Information Processing Systems (NIPS)*, 1997.

Hans-Otto Georgii. *Gibbs measures and phase transitions*, volume 9. Walter de Gruyter, 2011.

Vicenç Gómez, Joris M Mooij, and Hilbert J Kappen. Truncating the loop series expansion for belief propagation. *arXiv preprint cs/0612109*, 2006.

Tom Heskes. Stable fixed points of loopy belief propagation are minima of the bethe free energy. In *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.

Tom Heskes. Convexity arguments for efficient minimization of the bethe and kikuchi free energies. *Journal of Artificial Intelligence Research*, 26:153–190, 2006.

Lukas Kroc, Ashish Sabharwal, and Bart Selman. Survey propagation revisited. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, 2007.

Frank R. Kschischang, Brendan J. Frey, and Hans A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498 –519, 2001.

Steffen L Lauritzen and David J Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 157–224, 1988.

Elitza N. Maneva, Elchanan Mossel, and Martin J. Wainwright. A new look at survey propagation and its generalizations. *Journal of ACM*, 54:2–41, 2007.

Marc Mezard and Andrea Montanari. *Information, Physics, and Computation*. Oxford University Press, 2009.

Marc Mezard, Giorgio Parisi, and Riccardo Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297:812–815, 2002.

Andrea Montanari and Tommaso Rizzo. How to compute loop corrections to the bethe approximation. *Journal of Statistical Mechanics: Theory and Experiment*, 2005:16, 2005.

Joris M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173, 2010.

Joris M. Mooij and Hilbert J. Kappen. Loop corrections for approximate inference on factor graphs. *Journal of Machine Learning Research*, 8:1113–1143, 2007.

Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, 1999.

Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 2014.

Alessandro Pelizzola. Cluster variation method in statistical physics and probabilistic graphical models. *Journal of Physics A: Mathematical and General*, 38:36, 2005.

Siamak Ravanbakhsh and Russell Greiner. Revisiting algebra and complexity of inference in graphical models. *arXiv.org preprint*, arXiv:1409.7410, 2014.

Siamak Ravanbakhsh and Russell Greiner. Perturbed message passing for constraint satisfaction problems. *JMLR 16(Jul):1249-1274*, 2015.

Thomas J. Richardson and Rudiger L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47:599–618, 2001.

Martin J Wainwright, Tommi S. Jakkolla, and Alan S. Willsky. Map estimation via agreement on trees: Message-passing and linear programming. *IEEE Transactions on Information Theory*, 51:3697–3717, 2005.

Yair Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12:1–41, 2000.

Jonathan S. Yedida, William T. Freeman, and Yair Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51:2282–2312, 2005.

Jonathan S Yedidia, William T Freeman, Yair Weiss, et al. Generalized belief propagation. In *NIPS*, volume 13, pages 689–695, 2000.

Alan L. Yuille. CCCP algorithms to minimize the bethe and kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14:1691–1722, 2002.