Low-Complexity Stochastic Generalized Belief Propagation

Farzin Haddadpour
Electrical Engineering Department
Sharif University of Technology
Email: farzin_haddadpour@alum.sharif.edu

Mahdi Jafari Siavoshani
Computer Engineering Department
Sharif University of Technology
Email: mjafari@sharif.edu

Morteza Noshad
Electrical Engineering and
Computer Science Department
University of Michigan
Email: noshad@umich.edu

***Abstract*—The generalized belief propagation (GBP), introduced by Yedidia et al., is an extension of the belief propagation (BP) algorithm, which is widely used in different problems involved in calculating exact or approximate marginals of probability distributions. In many problems, it has been observed that the accuracy of GBP outperforms that of BP considerably. However, due to its generally higher complexity compared to BP, its application is limited in practice.**

In this paper, we introduce a stochastic version of GBP called *stochastic generalized belief propagation* (SGBP) that can be considered as an extension to the stochastic BP (SBP) algorithm introduced by Noorshams et al. They have shown that SBP reduces the complexity per iteration of BP by an order of magnitude in alphabet size. In contrast to SBP, SGBP can reduce the computation complexity if certain topological conditions are met by the region graph associated to a graphical model. However, this reduction can be larger than only one order of magnitude in alphabet size. In this paper, we characterize these conditions and the amount of complexity gain that one can obtain by using SGBP. Finally, using similar proof techniques employed by Noorshams et al., for general graphical models satisfy contraction conditions, we prove the asymptotic convergence of SGBP to the unique GBP fixed point, as well as providing non-asymptotic upper bounds on the mean square error and on the high probability error.

I. Introduction

Graphical models and corresponding message-passing algorithms have attracted a great amount of attention due to their wide-spreading application in many fields, including signal processing, machine learning, channel and source coding, computer vision, decision making, and game theory (e.g., see [1], [2]).

Finding marginal and mode of a probability distribution are two basic problems encountered in the field of graphical models. Taking the rudimentary approach, the marginalization problem has exponentially growing complexity in alphabet size. However, using BP algorithm (firstly introduced in [3]) to solve this problem either exactly or approximately, we can reduce the computational complexity to a significant degree. It has been proved that applying BP on graphical models without cycles provides exact solution to the marginalization problem. Furthermore, it has been observed that for general graphs, BP can find good approximations for marginalization (or finding mode) problems, [1], [2].

Although BP has many favourable properties, it suffers from some limiting drawbacks. First, in complex and densely interconnected graphs, BP may not be able to produce accurate results; and even worse, it may not converge at all. Second, since in many applications (e.g., decoding of error-correcting codes) messages are of high dimensions, the computational complexity of BP algorithm will highly increase which leads to slow convergence rates.

To deal with the first drawback, some works have been done to propose alternative algorithms (e.g., see [4], [5], [6], [7]). Specifically, to improve the accuracy of estimated marginal distribution, a generalization algorithm to BP has been introduced by Yedidia et al. [8], known as Generalized Belief Propagation (GBP) algorithm. In their proposed algorithm, local computation is performed by a group of nodes instead of a single node as in BP. According to many empirical observations, GBP outperforms BP in many situations; [9], [10], [11], [12]. However, although GBP algorithm provides accurate results in terms of marginal distribution, it suffers from high order of computation complexity, specially in case of large alphabet size.

To overcome the second aforementioned deficiency of BP, lots of research have been conducted to reduce BP complexity for different applications (e.g., refer to [13], [14], [15], [16], [17], [18], [19], [20]). In a recent work by Noorshams et al. [21], to tackle with the challenge of high complexity in the case of large alphabet size, they introduce an alternative stochastic version of BP algorithm with potentially lower complexity. The main idea behind their work is that each node sends a randomly sampled message taken from a properly chosen probability distribution instead of computing the exact message update rule in each iteration.

In this work, motivated by [21] and in order to mitigate the computational complexity of GBP, we extend GBP and propose stochastic GBP (SGBP) algorithm. SGBP has the advantage of reducing the complexity, while increasing the accuracy of estimation. In contrast to SBP, SGBP algorithm can reduce the computational complexity only if certain topological conditions are met by the region graph (defined later) associated to a graphical model. However, the complexity gain can be larger than only one order of magnitude in alphabet size. We characterize these conditions and the amount of computational gain that we can obtain by performing SGBP instead of GBP. Determining these criteria, we hope that they provide some useful guidelines on how to choose the regions

978-1-5090-1806-2/16/$31.00 ©2016 IEEE

and construct the region graph in a way that results to a lower complexity algorithm with good accuracy.

The rest of the paper is organized as follows. First, §II introduces our problem statement. In §III, we present the proposed stochastic GBP and then derive the topological conditions that guarantee SGBP has lower complexity than GBP. Moreover, theoretical convergence results have been provided as well. Finally, to validate our theoretical results, considering a specific graphical model, SGBP is simulated and the results are presented. For an extended version of this work please refer to [22].

II. Problem Statement

A. Notation

In the following, we introduce the notation that will be used in the paper. The random variables are represented by upper case letters and their values by lower case letters. Vectors and matrices are determined by bold letters. Sometimes, we use calligraphic letters to denote sets. When we have a set of random variables $X_1, \dots, X_n$, we write $\boldsymbol{X}_{\mathcal{A}}$ to denote $(X_i, i \in \mathcal{A})$. An undirected graph $G = (\mathcal{V}, \mathcal{E})$ is defined by a set of nodes $\mathcal{V} = \{1, 2, \dots, n\}$ and a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, where $(u, v) \in \mathcal{E}$ if and only if nodes u and v are connected. Similarly, we can define a directed graph.

For every function $f(x_1, x_2, \dots, x_n)$ where $f : \mathcal{X}^n \mapsto \mathbb{R}$, we define the operator $\mathcal{L}$ as a map that turns this function to a vector $\mathcal{L}(f) \in \mathbb{R}_{|\mathcal{X}|^n \times 1}$ by evaluating f at every input point.

B. Graphical Model

Undirected graphical models, also known as Markov random fields (MRF), is a way to represent the probabilistic dependencies among a set of random variables having Markov properties using an undirected graph. More precisely, we say that a set of random variables $X_1, \dots, X_n$ form an MRF if there exists a graph $G = (\mathcal{V}, \mathcal{E})$, where each X_i is associated to the node $i \in \mathcal{V} = \{1, \dots, n\}$, and edges of the graph G encode Markov properties of the random vector $\boldsymbol{X} = (X_1, \dots, X_n)$. These Markov properties are equivalent to a factorization of the joint distribution of random vector $\boldsymbol{X}$ over the cliques of graph G [23]. In this paper, we focus on discrete random variables case where for all $j \in \mathcal{V}$ we have $X_j \in \mathcal{X} \triangleq \{1, 2, \dots, d\}$. Moreover, we assume that the distribution of $\boldsymbol{X}$ is factorized according to

$$p(\boldsymbol{x}) = \frac{1}{Z} \prod_{a \in \mathcal{F}} \phi_a(\boldsymbol{x}_a)$$

where $\mathcal{F}$ is a collection of subsets of $\mathcal{V}$ and Z is a constant called the partition function. For the factor functions ϕ_a, we have also $\phi_a \geq 0$. This factorization can be represented by using a bipartite graph $G_f = (\mathcal{V}, \mathcal{F}, \mathcal{E}_f)$ called factor graph. In this representation, the variable nodes $\mathcal{V}$ correspond to random variables X_i's and factor nodes $\mathcal{F}$ determine the factor functions ϕ_a's. Moreover, there exists an edge $(i, a) \in \mathcal{E}_f$ between a variable node i and a factor node a if the variable x_i appears in the factor ϕ_a (for more information on factor graphs refer to [1]).

C. Region Graph

In order to present the Yedidia's parent-to-child algorithm [8] as well as introducing our stochastic GBP algorithm, we need to state some definitions as follows.

Definition 1 (see [8])**.** *A region graph $G_r = (\mathcal{R}, \mathcal{E}_r)$ defined over a factor graph $G_f = (\mathcal{V}, \mathcal{F}, \mathcal{E}_f)$ is a directed graph in which for each vertex $v \in \mathcal{R}$ (corresponding to a region) we have $v \subseteq \mathcal{V} \cup \mathcal{F}$. Each region v has this property that if a factor node $a \in \mathcal{F}$ belongs to v then all of its neighbouring variable nodes have to also belong to v. A directed edge $(v_p \to v_c) \in \mathcal{E}_r$ may exist if $v_c \subset v_p$. If such an edge exists, v_p is a* parent *of v_c, or equivalently, v_c is a* child *of v_p. If there exists a directed path from v_a to v_d on G_r, we say that v_a is an* ancestor *of v_d and v_d is a* descendant *of v_a.*

Now, for each $R \in \mathcal{R}$, we let $\mathcal{P}(R)$ denotes for the set of all parents of R, $\mathcal{A}(R)$ denotes for the set of all ancestors of R and $\mathcal{D}(R)$ denotes for the set of all descendants of R. Moreover, we define $\mathcal{E}(R) \triangleq R \cup \mathcal{D}(R)$. Finally, for $R \in \mathcal{R}$, we use $|R|$ to denote for the number of variable nodes in R.

D. Parent-to-child GBP algorithm

Yedidia et al., generalize the idea behind BP in [8], proposing an algorithm called parents-to-child GBP algorithm[1]. As explained in [8], in the parent-to-child algorithm, we have only one kind of message $m_{P \to R}(\boldsymbol{x}_R)$ from a parent region P to a child region R. Then, for the belief of region $R \in \mathcal{R}$ we have

$$\begin{aligned} b_R(\boldsymbol{x}_R) \propto \Phi_R(\boldsymbol{x}_R) \times &\prod_{P \in \mathcal{P}(R)} m_{P \to R}(\boldsymbol{x}_R) \\ &\times \prod_{D \in \mathcal{D}(R)} \prod_{P' \in \mathcal{P}(D) \setminus \mathcal{E}(\mathcal{R})} m_{P' \to D}(\boldsymbol{x}_D) \qquad (1) \end{aligned}$$

where $\Phi_R(\boldsymbol{x}_R) \triangleq \prod_{a \in R} \phi_a(\boldsymbol{x}_a)$ (with an abuse of notation when we product over $a \in R$ we mean to product only over the factor indexes of R). Moreover, the message update rule over each edge $(P, R) \in \mathcal{E}_r$ is given by

$$\begin{aligned} m_{P \to R}(\boldsymbol{x}_R) &= \frac{\sum_{\boldsymbol{x}_{P \setminus R}} \Phi_{P \setminus R}(\boldsymbol{x}_{P'}) \prod_{(I,J) \in N(P,R)} m_{I \to J}(\boldsymbol{x}_J)}{\prod_{(I,J) \in D(P,R)} m_{I \to J}(\boldsymbol{x}_J)} \\ &= \sum_{\boldsymbol{x}_{P \setminus R}} \Phi_{P \setminus R}(\boldsymbol{x}_{P'}) \hat{M}(\boldsymbol{x}_{T_{PR}}) \qquad (2) \end{aligned}$$

where $\Phi_{P \setminus R}(\boldsymbol{x}_{P'}) \triangleq \frac{\Phi_P}{\Phi_R}(\boldsymbol{x}_{P'})$ and P' is the set of all variables appear in $\frac{\Phi_P}{\Phi_R}(\boldsymbol{x}_{P'})$. In addition, we have also

$$N(P, R) \triangleq \Big\{(I, J) | (I, J) \in \mathcal{E}_r, I \notin \mathcal{E}(P), J \in \mathcal{E}(P) \setminus \mathcal{E}(R)\Big\}$$

and

$$D(P, R) \triangleq \Big\{(I, J) | (I, J) \in \mathcal{E}_r, I \in \mathcal{D}(P) \setminus \mathcal{E}(R), J \in \mathcal{E}(R)\Big\}.$$

Notice that the sets $N(P, R)$ and $D(P, R)$ can be calculated in advance. Also, $\hat{M}(\boldsymbol{x}_{T_{PR}})$ in (2) is defined as follows

$$\hat{M}(\boldsymbol{x}_{T_{PR}}) \triangleq \frac{\prod_{(I,J) \in N(P,R)} m_{I \to J}(\boldsymbol{x}_J)}{\prod_{(I,J) \in D(P,R)} m_{I \to J}(\boldsymbol{x}_J)},$$

[1] More precisely, they have proposed different variation of GBP but here we only focus on parent-to-child algorithm.

where T_{PR} is the set of all variables that appear in the above ratio.

Remark 1. *In the parent-to-child algorithm, the message transmitted over each edge $(P, R) \in \mathcal{E}_r$ can be considered as a vector by applying the operator $\mathcal{L}(\cdot)$. Namely, by concatenating all possible messages, we define $\boldsymbol{m}_{P\to R} \triangleq \mathcal{L}(m_{P\to R})$ where $\boldsymbol{m}_{P\to R} \in \mathbb{R}^{d^{|R|}}$. Moreover, concatenating all the messages over all edges of the region graph, we define $\boldsymbol{m} \triangleq \{\boldsymbol{m}_{P\to R}\}_{(P,R)\in\mathcal{E}_r} \in \mathbb{R}^{\Delta}$ where $\Delta = \sum_{(P,R)\in\mathcal{E}_r} d^{|R|}$.* ■

Now, we can state the complexity of the parent-to-child GBP algorithm as stated in the following lemma.

Lemma 1. *The computation complexity of the message update rule associated with each edge in the parent-to-child GBP algorithm, computed according to* (2), *is $\mathcal{O}(d^{|P|})$.*

Proof: For each fixed vector $\boldsymbol{x}_R$, the calculation of $m_{P\to R}(\boldsymbol{x}_R) = \sum_{\boldsymbol{x}_{P\setminus R}} \Phi_{P\setminus R}(\boldsymbol{x}_{P'})\hat{M}(\boldsymbol{x}_{T_{PR}})$ requires $d^{|P\setminus R|}$ operations. Moreover, to determine $m_{P\to R}(\cdot)$ completely, one needs to evaluate the above summation $\mathcal{O}(d^{|R|})$ times. Consequently, the overall complexity of calculating $m_{P\to R}(\boldsymbol{x}_R)$ is of order $\mathcal{O}(d^{|R|} \times d^{|P\setminus R|}) = \mathcal{O}(d^{|P|})$. ■

At each round of the parent-to-child algorithm, $t = 1, 2, \ldots$, every parent node P of R in the region graph calculates a message $m^{(t+1)}_{P\to R}$ and sends it to node R. Mathematically, this can be written as (see [8])

$$\begin{aligned} m^{(t+1)}_{P\to R}(\boldsymbol{x}_R) &= \Big[\Upsilon_{P\to R}(m^{(t)})\Big](\boldsymbol{x}_R) \\ &= \sum_{\boldsymbol{x}_{P\setminus R}} \Phi_{P\setminus R}(\boldsymbol{x}_{P'})\hat{M}^{(t)}(\boldsymbol{x}_{T_{PR}}). \end{aligned}$$

Now, this expression can be expanded as

$$\begin{aligned} m^{(t+1)}_{P\to R}(\boldsymbol{x}_R) &= \sum_{\boldsymbol{x}_{(P\setminus R)\setminus T_{PR}}} \sum_{\boldsymbol{x}_{(P\setminus R)\cap T_{PR}}} \Phi_{P\setminus R}(\boldsymbol{x}_{P'})\hat{M}^{(t)}(\boldsymbol{x}_{T_{PR}}) \\ &= k^{(t)}_{PR}\left(\boldsymbol{x}_{T_{PR}\setminus(P\setminus R)}\right) \sum_{\boldsymbol{x}_{(P\setminus R)\setminus T_{PR}}} \sum_{\boldsymbol{x}_{(P\setminus R)\cap T_{PR}}} \Big[\Phi_{P\setminus R}(\boldsymbol{x}_{P'}) \\ &\quad \times Q^{(t)}(\boldsymbol{x}_{T_{PR}\cap(P\setminus R)}|\boldsymbol{x}_{T_{PR}\setminus(P\setminus R)})\Big], \end{aligned} \tag{3}$$

where

$$Q^{(t)}(\boldsymbol{x}_{T_{PR}\cap(P\setminus R)}|\boldsymbol{x}_{T_{PR}\setminus(P\setminus R)}) \triangleq \frac{\hat{M}^{(t)}\left(\boldsymbol{x}_{T_{PR}\cap(P\setminus R)}, \boldsymbol{x}_{T_{PR}\setminus(P\setminus R)}\right)}{\sum_{\boldsymbol{x}'_{T_{PR}\cap(P\setminus R)}} \hat{M}^{(t)}\left(\boldsymbol{x}'_{T_{PR}\cap(P\setminus R)}, \boldsymbol{x}_{T_{PR}\setminus(P\setminus R)}\right)} \tag{4}$$

is a conditional distribution. Moreover,

$$k^{(t)}_{PR}\left(\boldsymbol{x}_{T_{PR}\setminus(P\setminus R)}\right) \triangleq \sum_{\boldsymbol{x}'_{T_{PR}\cap(P\setminus R)}} \hat{M}^{(t)}\left(\boldsymbol{x}'_{T_{PR}\cap(P\setminus R)}, \boldsymbol{x}_{T_{PR}\setminus(P\setminus R)}\right).$$

Hence, for the update rule we can write

$$m^{(t+1)}_{P\to R}(\boldsymbol{x}_R) = k^{(t)}_{PR} \sum_{\boldsymbol{x}_{(P\setminus R)\setminus T_{PR}}} \mathbb{E}_{[\boldsymbol{X}_{(P\setminus R)\cap T_{PR}}\sim Q^{(t)}]}\left[\Phi_{P\setminus R}(\boldsymbol{X}_{P'})\right] \tag{5}$$

Here and in the following, for brevity and clarity of notation, we will omit the dependence of $k^{(t)}_{PR}$ to the variables $\boldsymbol{x}_{T_{PR}\setminus(P\setminus R)}$.

Now, notice that we can decompose the set P' as follows

$$P' = [(P \setminus R) \cap T_{PR}] \cup [(P \setminus R) \setminus T_{PR}] \cup [P' \setminus (P \setminus R)]$$

because we always have $P \setminus R \subseteq P'$. By using this relation, we can rewrite (5) as

$$\begin{aligned} m^{(t+1)}_{P\to R}(\boldsymbol{x}_R) = k^{(t)}_{PR} &\sum_{\boldsymbol{x}_{(P\setminus R)\setminus T_{PR}}} \mathbb{E}_{[\boldsymbol{X}_{(P\setminus R)\cap T_{PR}}\sim Q^{(t)}]}\Bigg[\\ &\Phi_{P\setminus R}\Big(\boldsymbol{X}_{(P\setminus R)\cap T_{PR}}, \boldsymbol{x}_{(P\setminus R)\setminus T_{PR}}, \boldsymbol{x}_{P'\setminus(P\setminus R)}\Big)\Bigg]. \end{aligned} \tag{6}$$

In (3), $\Upsilon_{P\to R} : \mathbb{R}^{\Delta} \mapsto \mathbb{R}^{d^{|R|}}$ is the local update function of the directed edge $(P, R) \in \mathcal{E}_r$. By concatenating all of the local update functions over the edges of the region graph, we can define the global update function as

$$\Upsilon(\boldsymbol{m}) = \Big[\Upsilon_{P\to R}(\boldsymbol{m}) : (P, R) \in \mathcal{E}_r\Big] \tag{7}$$

where $\Upsilon : \mathbb{R}^{\Delta} \mapsto \mathbb{R}^{\Delta}$. The goal of the (parent-to-child) GBP algorithm is to find a fixed point $\boldsymbol{m}^*$ that satisfies $\Upsilon(\boldsymbol{m}^*) = \boldsymbol{m}^*$. If a fixed point $\boldsymbol{m}^*$ is found, then the beliefs of random variables in a region $R \in \mathcal{R}$ is computed by applying (1).

III. Stochastic Generalized Belief Propagation Algorithm

In this section, first we introduce our stochastic extension to the parent-to-child GBP algorithm, and then present a result on the criteria where this algorithm is able to mitigate the computation complexity of GBP.

Based on (6), we introduce our algorithm as stated in Algorithm 1. The main idea of the algorithm is that under proper conditions (that will be stated in Theorem 1), some parts of the message update rule (2) for each edge of the region graph can be written as an expectation as stated in (6).

Algorithm 1 Stochastic Generalized Belief Propagation (SGBP) algorithm.

```
Initialize the messages.
for $t \in \{1, 2, \ldots\}$ and each directed edge $(P, R) \in \mathcal{E}_r$ do
    Choose a random vector $\boldsymbol{J}^{(t+1)}_{PR} \in \mathcal{X}^{|T_{PR}\cap(P\setminus R)|}$ according to the conditional distribution $Q^{(t)}(\boldsymbol{x}_{T_{PR}\cap(P\setminus R)}|\boldsymbol{x}_{T_{PR}\setminus(P\setminus R)})$ defined in (4).
    Update the message $m^{(t+1)}_{P\to R}$ with the appropriately tuned step size $\alpha^{(t)} = \mathcal{O}(\frac{1}{t})$ according to
        $m^{(t+1)}_{P\to R}(\boldsymbol{x}_R) = (1 - \alpha^{(t)})m^{(t)}_{P\to R}(\boldsymbol{x}_R)$
        $+ \alpha^{(t)}k^{(t)}_{PR} \sum_{\boldsymbol{x}_{(P\setminus R)\setminus T_{PR}}} \Phi_{P\setminus R}\left(\boldsymbol{J}^{(t+1)}_{PR}, \boldsymbol{x}_{(P\setminus R)\setminus T_{PR}}, \boldsymbol{x}_{P'\setminus(P\setminus R)}\right)$   (8)
    $t = t + 1$
end for
```

Remark 2. *Note that when $(P \setminus R) \cap T_{PR} = \varnothing$, the update rule* (8) *becomes deterministic as stated in the following*

$$m_{P\to R}^{(t+1)}(\boldsymbol{x}_R) = (1-\alpha^{(t)})m_{P\to R}^{(t)}(\boldsymbol{x}_R) \\ + \alpha^{(t)} k_{PR}^{(t)}(\boldsymbol{x}_{T_{PR}}) \Big[\sum_{\boldsymbol{x}_{(P\setminus R)}} \Phi_{P\setminus R}\left(\boldsymbol{x}_{(P\setminus R)}, \boldsymbol{x}_{P'\setminus(P\setminus R)}\right) \Big].$$

It is shown in [22] that this condition can only happen in update rules corresponding to the highest-level ancestors regions. ■

In contrast to SPB studied in [21], the stochastic version of GBP does not always reduce the computational complexity in each iteration. Theorem 1 describes the topological and regional conditions for which the complexity of SGBP is less than GBP for a specific edge of the region graph.

Theorem 1. *Our proposed algorithm that runs over a region graph G_r reduces the computation complexity of each message $m_{P\to R}$ (compared to GBP) if and only if the following conditions hold*

(i) $(P \setminus R) \cap T_{PR} \neq \varnothing$,
(ii) $(P \setminus R) \not\subseteq T_{PR}$.

Proof: The main idea of the proof lies in the fact that whether or not (2) can be written in the form of an expected value of *potential functions* as stated in (6). If this happens, as presented in Algorithm 1, the complexity of update rules can be reduced. To be able to have an expectation operation in (6), we should have $(P \setminus R) \cap T_{PR} \neq \varnothing$.

Now, assuming condition (i) holds, we find the complexity of Algorithm 1's update rule over every edge $(P, R) \in \mathcal{E}_r$ in each iteration. First, let us fix $\boldsymbol{x}_R$. To find the PMF of the random vector $\boldsymbol{J}_{PR}$ which is given by (4), we need $\mathcal{O}(d^{|\{P\setminus R\}\cap T_{PR}|} \times d^{|T_{PR}\setminus\{P\setminus R\}|}) = \mathcal{O}(d^{|T_{PR}|})$ operations. Notice that since we have $T_{PR} \setminus (P \setminus R) \subseteq R$ and $[(P \setminus R) \cap T_{PR}] \cap [(P \setminus R) \setminus T_{PR}] = \varnothing$, for every fixed $\boldsymbol{x}_R$, the PMF of $\boldsymbol{J}_{PR}$ does not depend on the vector $\boldsymbol{x}_{(P\setminus R)\setminus T_{PR}}$. This means that for a fixed $\boldsymbol{x}_R$, to find the summation in (8), the PMF of $\boldsymbol{J}_{PR}$ should only computed once.

Hence, the overall complexity of update rule (8) becomes

$$\mathcal{O}\left(d^{|T_{PR}|} + d^{|R|}\big[d^{|(P\setminus R)\cap T_{PR}|} + d^{|(P\setminus R)\setminus T_{PR}|} + d^{|(P\setminus R)\cap T_{PR}|}\big]\right)$$

where the terms in the brackets count for a fixed $\boldsymbol{x}_R$ the computation complexity of $k(\boldsymbol{x}_{T_{PR}\setminus(P\setminus R)})$, of the summation in (8), and of taking a sample vector $\boldsymbol{J}_{PR}$ from the above PMF, respectively. The above relation can be rewritten as follows

$$\mathcal{O}\left(\max\left[d^{|T_{PR}|}, d^{|R|+|(P\setminus R)\setminus T_{PR}|}, d^{|R|+|(P\setminus R)\cap T_{PR}|}\right]\right).$$

Now, we can conclude that if $T_{PR} \neq \varnothing$ and $(P\setminus R) \not\subseteq T_{PR}$ then we have

$$\mathcal{O}\left(\max\left[d^{|T_{PR}|}, d^{|R|+|(P\setminus R)\setminus T_{PR}|}, d^{|R|+|(P\setminus R)\cap T_{PR}|}\right]\right) < \mathcal{O}(d^{|P|}),$$

where the right hand side is the computation complexity of the parent-to-child GBP algorithm derived in Lemma 1. This completes the proof of theorem. ■

Corollary 1. *Assuming that the conditions of Theorem 1 hold and denoting*

$$\eta_{PR} \triangleq \max\Big[|T_{PR}|, |R| + |(P\setminus R)\setminus T_{PR}|, |R| + |(P\setminus R)\cap T_{PR}|\Big],$$

Algorithm 1 reduces the computation complexity of message $m_{P\to R}$ of the order $\mathcal{O}(d^{|P|-\eta_{PR}}) = \mathcal{O}(d^{I_{PR}})$ where $I_{PR} \triangleq |P| - \eta_{PR}$. Notice that I_{PR} can be larger than 1.

Corollary 2. *The complexity of the parent-to-child GBP algorithm is dominated by the computation complexity of message update rule of the highest-level regions in the region graph G_r. As a result, if the dominant message update rule that belongs to the highest-level ancestor regions with the largest size, satisfies the conditions of Theorem 1, then no matter what are the complexity of other edges, Algorithm 1 will reduce the overall computation complexity of the parent-to-child GBP.*

A. Convergence Rate of SGBP Algorithm

In this section, we extend the convergence guarantees of [21] to SGBP. Our convergence theorem (Theorem 2) is based on imposing a sufficient condition similar to [21] that guarantees uniqueness and convergence of the parent-to-child GBP message updates. More precisely we assume that the global update function $\Upsilon(\cdot)$, defined in (7), is contractive, namely $\exists \nu, 0 < \nu < 2$ such that

$$\|\Upsilon(\boldsymbol{m}) - \Upsilon(\boldsymbol{m}')\|_2 \leq \left(1 - \frac{\nu}{2}\right)\|\boldsymbol{m} - \boldsymbol{m}'\|_2. \tag{9}$$

Following similar proof technique to [21], with some appropriate modifications, we can obtain the following results.

Theorem 2. *Assume that, for a given region graph, the global update function Υ is contractive with parameter $1 - \frac{\nu}{2}$ as defined in* (9). *Then, parent-to-child GBP has a unique fixed point $\boldsymbol{m}^*$ and the message sequence $\{\boldsymbol{m}_{P\to R}^{(t)}\}_{t=1}^{\infty}$ generated by the SGBP algorithm has the following properties:*

i) *The result of SGBP is consistent with GBP, namely we have $\boldsymbol{m}^{(t)} \xrightarrow{\text{a.s.}} \boldsymbol{m}^*$ as $t \longrightarrow \infty$.*
ii) *Bounds on mean-squared error: Let us divide the fixed point message $\boldsymbol{m}^*$ into two parts, $\boldsymbol{m}^* = (\boldsymbol{m}^*_{\mathcal{E}_1}, \boldsymbol{m}^*_{\mathcal{E}_{\sim 1}})$, where $\boldsymbol{m}^*_{\mathcal{E}_1}$ corresponds to those edges of the region graph that perform constant update rule (see [22] for more details), while $\boldsymbol{m}^*_{\mathcal{E}_{\sim 1}}$ corresponds to the rest of edges. Choosing step size $\alpha^{(t)} = \frac{\alpha}{\nu(t+2)}$ for some fixed $1 < \alpha < 2$ and defining $\delta_i^{(t)} \triangleq \frac{\boldsymbol{m}_i^{(t)} - \boldsymbol{m}_i^*}{\|\boldsymbol{m}_i^*\|^2}$ for each $i \in \{1, \sim 1\}$, we have*

$$\frac{\mathbb{E}[\|\delta^{(t)}\|_2^2]}{\|\boldsymbol{m}^*\|_2^2} \leq \left(\frac{3^\alpha \alpha^2 \Lambda(\Phi', k_{lu})}{2^\alpha(\alpha-1)\nu^2}\right)\frac{1}{t} + \frac{\mathbb{E}[\|\delta^{(0)}_{\mathcal{E}_{\sim 1}}\|_2^2]}{\|\boldsymbol{m}^*_{\mathcal{E}_{\sim 1}}\|_2^2}\left(\frac{2}{t}\right)^{\alpha}$$

for all iteration $t = 1, 2, 3, \ldots$ where $\Lambda(\Phi', k_{lu})$ is a constant which depends on some factor functions (through Φ') and some variable nodes (through k_{lu}). For more details refer to [22].

iii) *High probability bounds on error: With step size* $\alpha^{(t)} = \frac{1}{\nu(t+1)}$, *for any* $1 > \epsilon > 0$ *and* $\forall t = 1, 2, \ldots$, *we have*

$$\delta^{(t+1)} \leq \frac{\Lambda(\Phi', k_{lu})}{\nu^2} \frac{1 + \log(t+1)}{t+1} + \frac{4Q(\Phi', k_{lu})}{\nu^2\sqrt{\epsilon}} \frac{\sqrt{(1+\log(t+1))^2 + 4}}{t+1}$$

with probability at least $1 - \epsilon$.

The proof of Theorem 2 and more discussion about the overall complexity of SGBP versus GBP can be found in [22].

B. Simulation Results

In this section, considering a pairwise MRF, we present some simulation results to study the impact of our algorithm along verifying our theoretical results. We choose the so-called Potts model (which is a generalization to Ising model; see [13]) of size 3×3 for our simulation purpose (i.e., a 3×3 lattice). We have the following potentials assigned to each of the edges $(u, v) \in \mathcal{E}$

$$\psi_{uv}(i, j) = \begin{cases} 1 & \text{if } i = j, \\ \gamma & \text{Otherwise.} \end{cases}$$

where $0 < \gamma < 1$. For the nodes' potential we have

$$\phi_u(i) = \begin{cases} 1 & \text{if } i = 1, \\ \mu + \sigma Y & \text{Otherwise.} \end{cases}$$

in which σ and μ satisfy the conditions $0 < \sigma \leq \mu$ and $\sigma + \mu < 1$ and Y should have the uniform distribution over the interval $(-1, 1)$ in addition to being independent from other parameters. We take the following steps to run our simulation. First, setting $\sigma = \mu = \gamma = 0.1$, we run parent-to-child algorithm with region size of 2×2 to get the asymptotic $\boldsymbol{m}^*$. Second, with the same parameters and taking $\alpha^{(t)} = \frac{2}{(1+t)}$ for $d \in \{4, 8, 16, 32\}$, we perform Algorithm 1 for the same region graph. It is worth noting that to calculate $\frac{\mathbb{E}[\|\delta^{(t)}\|_2^2]}{\|\boldsymbol{m}^*\|_2^2}$, we run algorithm 20 times and then average over error corresponding to each simulation. As it is illustrated in the simulation result of Figure 1, this result is in consistency with Theorem 2.

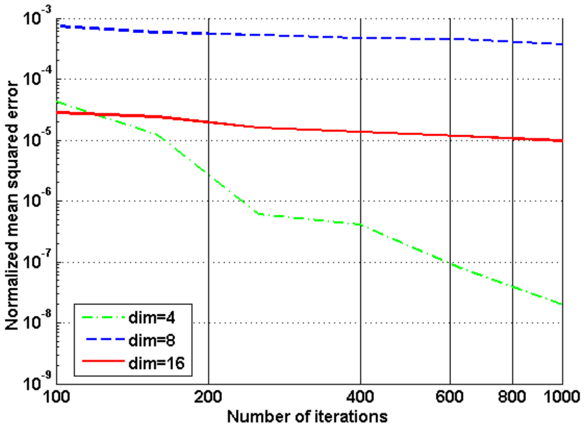

Fig. 1. The normalized mean-squared error of SGBP versus number of iterations for a Potts models of size 3×3.

References

[1] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 498–519, 2001.

[2] H.-A. Loeliger, "An introduction to factor graphs," *Signal Processing Magazine, IEEE*, vol. 21, no. 1, pp. 28–41, 2004.

[3] J. Pearl, "Probabilistic reasoning in intelligent systems; networks of plausible inference," 1988.

[4] A. L. Yuille, "Cccp algorithms to minimize the bethe and kikuchi free energies: Convergent alternatives to belief propagation," *Neural computation*, vol. 14, no. 7, pp. 1691–1722, 2002.

[5] M. Welling and Y. W. Teh, "Belief optimization for binary networks: A stable alternative to loopy belief propagation," in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2001, pp. 554–561.

[6] T. Heskes, "Stable fixed points of loopy belief propagation are local minima of the bethe free energy," in *Advances in neural information processing systems*, 2002, pp. 343–350.

[7] P. Pakzad and V. Anantharam, "Estimation and Marginalization Using the Kikuchi Approximation Methods," *Neural Computation*, vol. 17, no. 8, pp. 1836–1873, Aug. 2005.

[8] J. S. Yedidia, W. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2282–2312, Jul. 2005.

[9] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Characterization of belief propagation and its generalizations," *IT-IEEE*, vol. 51, pp. 2282–2312, 2001.

[10] J. Harel, "Poset belief propagation: Experimental results," Ph.D. dissertation, CaliFornia Institute oF Technology, 2003.

[11] M. Welling, "On the choice of regions for generalized belief propagation," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 2004, pp. 585–592.

[12] J.-C. Sibel, S. Reynal, and D. Declercq, "An application of generalized belief propagation: splitting trapping sets in ldpc codes," in *Information Theory (ISIT), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 706–710.

[13] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *International journal of computer vision*, vol. 70, no. 1, pp. 41–54, 2006.

[14] E. B. Sudderth, A. T. Ihler, M. Isard, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," *Communications of the ACM*, vol. 53, no. 10, pp. 95–103, 2010.

[15] J. J. Mcauley and T. S. Caetano, "Faster algorithms for max-product message-passing," *The Journal of Machine Learning Research*, vol. 12, pp. 1349–1388, 2011.

[16] M. Isard, J. MacCormick, and K. Achan, "Continuously-adaptive discretization for message-passing algorithms," in *Advances in Neural Information Processing Systems*, 2009, pp. 737–744.

[17] K. Kersting, B. Ahmadi, and S. Natarajan, "Counting belief propagation," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2009, pp. 277–284.

[18] J. Coughlan and H. Shen, "Dynamic quantization for belief propagation in sparse spaces," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 47–58, 2007.

[19] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174–188, 2002.

[20] A. Smith, A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*. Springer Science & Business Media, 2013.

[21] N. Noorshams and M. Wainwright, "Stochastic Belief Propagation: A Low-Complexity Alternative to the Sum-Product Algorithm," *IEEE Transactions on Information Theory*, vol. 59, no. 4, pp. 1981–2000, Apr. 2013.

[22] F. Haddadpour, M. Jafari Siavoshani, and M. Noshad, "Low-complexity stochastic generalized belief propagation," 2016. [Online]. Available: http://arxiv.org/abs/1605.02046

[23] G. Grimmett, *Probability on Graphs: Random Processes on Graphs and Lattices*, ser. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2010.