Approximate message-passing inference algorithm

Kyomin Jung
Mathematics, MIT
Cambridge, MA 02139
kmjung@mit.edu

Devavrat Shah
EECS, MIT
Cambridge, MA 02139
devavrat@mit.edu

***Abstract*— In a recent result, Weitz [13] established equivalence between the marginal distribution of a node, say v, in any binary pair-wise Markov Random Field (MRF), say G, with the marginal distribution of the root node in the self-avoid walk tree of the G starting at v. Analogous result for max-marginal distribution holds for the reason that addition and multiplication commute in the same way as addition and maximum. This remarkable connection suggests a message-passing algorithm for computing exact marginal and max-marginal in any binary MRF.**

In this paper, we exploit this property along with appropriate graph partitioning scheme to design approximate message passing algorithms for computing max-marginal of nodes or maximum a-posteriori assignment (MAP) in a binary MRF G. Our algorithm can provide provably arbitrarily small error for a large class of graphs including planar graphs.

Our algorithms are linear in number of nodes G with constant dependent on the approximation error. For precise evaluation of computation cost of algorithm, we obtain a novel tight characterization of the size of self-avoiding walk tree for any connected graph as a function of number of edges and nodes.

I. Introduction

Markov Random Field (MRF) [7] based exponential family of distribution allows for representing distributions in an intuitive parametric form. Therefore, it has been successful for modeling in many applications [10]. Specifically, consider an exponential family on n random variables $\mathbf{X} = (X_1, \ldots, X_n)$ represented by a pair-wise (undirected) MRF with graph structure $G = (V, E)$, where vertices $V = \{1, \ldots, n\}$ and edge set $E \subset V \times V$. Each X_i takes value in a finite set Σ (e.g. $\Sigma = \{0, 1\}$). The joint distribution of $\mathbf{X} = (X_i)$: for $\mathbf{x} = (x_i) \in \Sigma^n$,

$$\mathbb{P}[\mathbf{X} = \mathbf{x}] \quad \propto \quad \prod_{i \in V} \phi_i(x_i) \prod_{(i,j) \in E} \psi_{ij}(x_i, x_j). \qquad (1)$$

Here, functions $\phi_i : \Sigma \to [1, \infty)$, and $\psi_{ij} : \Sigma^2 \to [1, \infty)$ are assumed to be arbitrary non-negative (real-valued) functions. An important computational question of interest is *finding maximum a-posteriori (MAP) assignment* $\mathbf{x}^*$, where $\mathbf{x}^* = \arg\max_{\mathbf{x} \in \Sigma^n} \mathbb{P}[\mathbf{X} = \mathbf{x}]$. MAP is equivalent to a *minimal energy assignment* (or ground state) where energy, $\mathcal{E}(\mathbf{x})$, of state $\mathbf{x} \in \Sigma^n$ is defined as $\mathcal{E}(\mathbf{x}) = -\mathcal{H}(\mathbf{x}) + \mathsf{Constant}$, where $\mathcal{H}(\mathbf{x}) = \sum_{i \in V} \log \phi_i(x_i) + \sum_{(i,j) \in E} \log \psi_{ij}(x_i, x_j)$. For ease of implementation and scalability, message passing algorithms have emerged as canonical solutions. In this paper, we present message passing algorithm that will find ε-approximation solution of MAP for a class of graphs.

Previous Work. The question of finding MAP (or ground state) comes up in many important application areas such as coding theory, discrete optimization, image denoising.This problem is NP-hard for exact and even (constant) approximate computation for arbitrary graph G. However, applications require solving this problem using very simple algorithms. A plausible approach is as follows. First, identify wide class of graphs that have simple algorithms for computing MAP and log-partition function. Then, try to build system (e.g. codes) so that such good graph structure emerges and use the simple algorithm or else use the algorithm as a heuristic.

Such an approach has resulted in many interesting recent results starting the Belief Propagation (BP) algorithm designed for Tree graph [7]. Since there a vast literature on this topic, we will recall only few results. Two important algorithms are the generalized belief propagation (BP) [14] and the tree-reweighted algorithm (TRW) [11], [12]. Key properties of interest for these iterative procedures are the correctness of fixed points and convergence. Many results characterizing properties of the fixed points are known starting from [14].Various sufficient conditions for their convergence are known starting [9]. However, simultaneous convergence and correctness of such algorithms are established for only specific problems, e.g. [1]. We take note of recent advances in the context of TRW algorithm by by Kolmogorov [3], Kolmogorov and Wainwright [4] where they made a predicted connection between TRW for MAP estimation and specific Linear Programming (LP) relaxation of the problem [12] precise.

Contribution. We propose a novel message passing algorithm for approximate computation of MAP. For any $\varepsilon > 0$, our algorithm can produce an ε-approximate solution for MAP for *arbitrary* binary MRF G as long as G admits a *good partitioning* property (defined precisely later in the paper). Class of graphs that admit this property as well as allow for message passing algorithm for finding such partition include include those that exclude a finite graph as a minor: planar graph is special case of such graphs.

The running time of the algorithm is $\Theta(n)$, with constant dependent on ε and the maximum vertex degree of G. In order to evaluate this constant for our message passing algorithm, we need to evaluate size of the self-avoiding walk tree of a graph. We show that the size of self-avoiding walk tree for arbitrary connected graph with n nodes and $n+k-1$ edges is no more than $(n+k-1)2^{k+1}$. Using this, in the specific case of Planar

1-4244-1564-0/07/$25.00 ©2007 IEEE

graph with bounded degree we show that algorithm performs $\leq C(\varepsilon)n$ operations to find an ε-approximate solution with $\log\log C(\varepsilon) = O(1/\varepsilon)$.

It is worth noting that, algorithm will work for any binary G with quantifiable error bound. We also note that, MAP computation for arbitrary pair-wise finite valued exponential family is equivalent to computing MAP for a specific binary MRF. Thus, in principle our algorithm extends for arbitrary finite valued exponential family MAP estimation.

Techniques. Our algorithm is primarily based on the following idea: First, decompose G into small-size connected components say $G_1, \ldots, G_k$ by removing few edges of G. Second, compute exact MAP in each of G_i separately. This computation is performed through a message passing algorithm using an adaptation of result of Weitz [13] for MAP. Third, combine these estimates to produce a global estimate while *taking care* of the effect induced by removed edges. This can be done in a local manner.

The error produced by the above method is primarily due to the edges removed and the computation time depends on the size of the components. For small error, we need to remove appropriately selected edges while for small computation time we need small size of components. The graphs with good partition property (defined later) posses both of these properties.

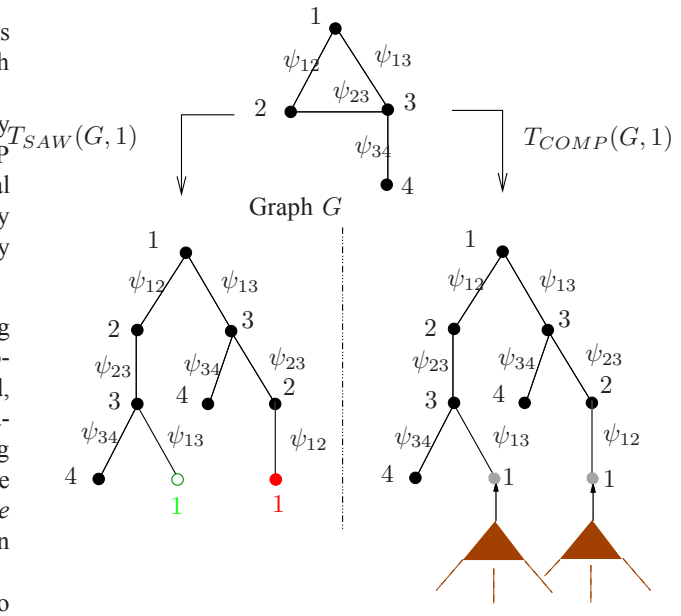

Fig. 1. A graph G of 4 nodes with one loop is given. On left, we have the self-avoiding walk tree of G for node 1, i.e. $T_{SAW}(G,1)$ with green and red being special nodes. On right, we have computation tree $T_{COMP}(G,1)$ for node 1's computation under Belief Propagation (or Max-Product) algorithm. The grey nodes of $T_{COMP}(G,1)$ correspond to green and red node of $T_{SAW}(G,1)$ on the left.

II. Preliminaries

This section contains two main ingredients for the results of this paper. The first result is about equivalence of max-marginal of a node, say v, in G and max-marginal of root of self-avoiding walk tree with respect to v. This result follows by a direct adaption of result by Weitz [13]. The second result is about existence of good graph partitioning property for a class of graphs. Here we describe graph partitioning schemes for minor-excluded graphs (based on result of Klein, Plotkin and Rao [2]).

A. Equivalence: MRF and Self-Avoiding Walk Tree

Given binary pair-wise MRF G of n nodes, our interest is in finding

$$p_v^*(\gamma) = \max_{\sigma\in\{0,1\}^n:\sigma_v=\gamma} \mathbb{P}(\sigma), \text{ for } \gamma \in \{0,1\} \text{ for all } v.$$

Definition 1 (Self-Avoiding Walk Tree): Consider graph $G = (V, E)$ of pair-wise binary MRF. For $v \in V$, we define the self avoiding walk tree $T_{SAW}(G, v)$ as follows. First, for each $u \in V$, give an ordering of its neighbors $N(u)$. This ordering can be arbitrary but remains fixed forever. Given this, $T_{SAW}(G, v)$ is constructed by the breadth first search of nodes of G starting from v without backtracking. Then stop the bread-first search along a direction when an already visited vertex is encountered (but include it in $T_{SAW}(G, v)$ as a leaf). Say one such leaf be $\hat{w}$ of $T_{SAW}(G, v)$ and let it be a copy of a node w in G. We call such a leaf node of $T_{SAW}(G, v)$ as *Marked*. A marked leaf node is assigned color *Red* or *Green* according to the following condition: The leaf $\hat{w}$ is marked since we encountered node w of G twice along our bread-first search excursion. Let the (directed) path between these two encounters of w in G be given by $(w, v_1, \ldots, v_k, w)$. Naturally, $v_1, v_k \in N(w)$ in G. We mark the leaf node $\hat{w}$ as *Green* if according to the ordering done by node w in G of its neighbors, if v_k is given smaller number than that of v_1. Else, we mark it as *Red*. Let $\mathbf{V}_v$ and $\mathbf{E}_v$ denote the set of nodes and vertices of tree $T_{SAW}(G, v)$. With little abuse of notation, we will call root of $T_{SAW}(G, v)$ as v.

Given a $T_{SAW}(G, v)$ for a node $v \in V$ in G, an MRF is naturally induced on it as follows: all edges inherit the pair-wise compatibility function (i.e. $\psi_{..}(\cdot,\cdot)$) and all nodes inherit node-potentials (i.e. $\phi_.(\cdot)$) from those of MRF G in a natural manner. The only distinction is the modification of the node-potential of *marked* leaf nodes of $T_{SAW}(G, v)$ as follows. A marked leaf node, say $\hat{w}$ of $T_{SAW}(G, v)$ modifies its potentials as follows: if it is *Green* than it sets $\phi_{\hat{w}}(1) = \phi_w(1), \phi_{\hat{w}}(0) = 0$ but if it is *Red* leaf node then it sets $\phi_{\hat{w}}(0) = \phi_w(0), \phi_{\hat{w}}(1) = 0$.

Example 1 (Self-avoiding walk tree): Consider 4 node binary pair-wise MRF G in Figure 1. Let node 1 gives number a to node 2, number b to node 3 so that $a > b$. Given this numbering, the bottom left of Figure 1 represents $T_{SAW}(G, 1)$. The Green leaf node essentially means that we set its value permanently to 1.

With above description, $T_{SAW}(G, v)$ gives rise to a pair-wise binary MRF. Let $\mathbb{Q}_{G,v}$ denote the probability distribution induced by this MRF on boolean cube $\{0, 1\}^{|\mathbf{V}_v|}$. Our interest

will be in the max-marginal for root v or equivalently

$$q_v^*(\gamma) = \max_{\sigma\in\{0,1\}^{|\mathbf{V}_v|}:\sigma_v=\gamma} \mathbb{Q}_{G,v}(\sigma), \text{ where } \gamma \in \{0,1\}.$$

Here we present an equivalence between $p_v^*(\cdot)$ and $q_v^*(\cdot)$. This is a direct adaptation of result by Weitz [13].

Theorem 1: Consider any binary pair-wise MRF $G = (V, E)$. For any $v \in V$, let $p_v^*(\cdot)$ be as defined above with respect to $\mathbb{P}_G$. Let $T_{SAW}(G, v)$ be the self-avoiding walk tree MRF and let $q_v^*(\cdot)$ be as defined above for root node of $T_{SAW}(G, v)$ with respect to $\mathbb{Q}_{G,v}$. Then,

$$\frac{p_v^*(1)}{p_v^*(0)} = \frac{q_v^*(1)}{q_v^*(0)}. \quad (2)$$

Here we allow ratio to be $0, \infty$.

Proof: The proof follows by induction. As a part of the proof, we will come across graphs with some *fixed* vertices, where a vertex u is said to be fixed to 0 (resp. 1) if $\phi_u(0) > 0$, $\phi_u(1) = 0$ (resp. $\phi_u(1) > 0$, $\phi_u(0) = 0$). The induction is on the number of *unfixed* vertices of G. We essentially prove the following, which implies the statement of Lemma: given any pair-wise MRF on a graph G (with possibly some *fixed* vertices), construct corresponding $T_{SAW}(G, v)$ MRF for some node v. If the number of *unfixed* vertex of G is at most m, then the (2) holds. Next, inductive proof.

Initial condition. Trivially the desired statement holds for any graph with exactly one *unfixed* vertex, by definition of MRF, i.e. (1). The reason is that for such a graph, due to all but one node being fixed, the max-marginal of each node is purely determined by its immediate neighbors due to Markovian nature of MRF. The immediate neighborhood of v in $T_{SAW}(G, v)$ and G is the same.

Hypothesis. Assume that the statement is true for any graph with less than or equal to $m \in \mathbb{N}$ *unfixed* nodes.

Induction step. Without loss of generality, suppose that our graph of interest, G, has $m + 1$ *unfixed* vertices. If v is a *fixed* vertex, then (2) holds trivially. Let $v \in V$ be an unfixed vertex of G. Then we will show via inductive hypothesis that

$$\frac{q_v^*(1)}{q_v^*(0)} = \frac{p_v^*(1)}{p_v^*(0)}.$$

Let d be the degree of v; $v_1, v_2, \ldots, v_d$ be the neighbors of v where the order of neighbors is the same as that used in definition of $T_{SAW}(G, v)$. Let T_ℓ be the ℓth subtree of $T_{SAW}(G, i)$ having v_ℓ as its root and $Y(\ell)$ be the binary pair-wise MRF induced on T_ℓ by restriction of $T_{SAW}(G, v)$. Let $q_\ell^*(\sigma)$ be the max-marginal of vertex v_ℓ taking value $\sigma \in \Sigma = \{0, 1\}$ with respect to $Y(\ell)$. Note that when T_ℓ consists of a single vertex, then $q_\ell^*(\sigma) \propto \phi_{v_\ell}(\sigma)$. Let $\lambda_v = \frac{\phi_v(1)}{\phi_v(0)}$. Then from definition of pair-wise MRF and tree-structure,

$$\frac{q_v^*(1)}{q_v^*(0)} = \lambda_v \prod_{\ell=1}^{d} \frac{\max_{\sigma\in\Sigma} \psi_{v_\ell,v}(\sigma, 1) q_\ell^*(\sigma)}{\max_{\sigma\in\Sigma} \psi_{v_\ell,v}(\sigma, 0) q_\ell^*(\sigma)}. \quad (3)$$

Now to calculate $\frac{p_v^*(1)}{p_v^*(0)}$, we define a new graph G' and the corresponding pair-wise MRF X' as follows. Let G' be the same as G except that v is replaced by d vertices $v'_1, v'_2, \ldots, v'_d$; each v'_ℓ is connected only to v_ℓ, $1 \le \ell \le d$. The X' is defined same as X except that $\phi_{v'_\ell}(1) = \lambda_v^{1/d}\phi_v(1)$, $\phi_{v'_\ell}(0) = \phi_v(0)$ and $\psi_{v_\ell v'_\ell} = \psi_{v_\ell v}$. Then,

$$\begin{aligned}\frac{p_v^*(1)}{p_v^*(0)} &= \frac{\max_{\left\{X':X'_{v'_1}=1,X'_{v'_2}=1,\ldots,X'_{v'_d}=1\right\}} \mathbb{P}_{G'}(X')}{\max_{\left\{X':X'_{v'_1}=0,X'_{v'_2}=0,\ldots,X'_{v'_d}=0\right\}} \mathbb{P}_{G'}(X')} \\ &= \prod_{\ell=1}^{d} \frac{\mu_\ell(1)}{\mu_\ell(0)}, \end{aligned} \quad (4)$$

where define $\mu_\ell(\sigma) = \max_{\{X':X'_{v'_\ell}=\sigma\}} \mathbb{P}[X' | \ X'_{v'_1} = 0, \ldots, X'_{v'_{(\ell-1)}} = 0, X'_{v'_{(\ell+1)}} = 1, \ldots, X'_{v'_d} = 1]$. The second equality in (4) follows by standard trick of Telescoping multiplication and Lemma 2.

Now for $1 \le \ell \le d$, consider MRF $X'(\ell)$ induced on $G'(\ell) = G' - \{v'_\ell\}$ by fixing $\{v'_1, \ldots v'_d\} - \{v'_\ell\}$ as follows: let $(\phi_{v'_1}(0) = 1, \phi_{v'_1}(1) = 0); \ldots; (\phi_{v'_{\ell-1}}(0) = 1, \phi_{v'_{\ell-1}}(1) = 0); (\phi_{v'_{\ell+1}}(0) = 0, \phi_{v'_{\ell+1}}(1) = 1); \ldots; (\phi_{v'_d}(0) = 0, \phi_{v'_d}(1) = 1)$. Then let $\nu_\ell(\sigma), \sigma \in \Sigma$ denote the max-marginal of v_ℓ for taking value σ with respect to $X'(\ell)$. Given this, by definition of MRF X' as well $X'(\ell)$ and noting that v'_ℓ is a leaf (only connected to v_ℓ) with respect to graph G', we have

$$\frac{\mu_\ell(1)}{\mu_\ell(0)} = \lambda_v^{1/d} \frac{\max_{\sigma\in\Sigma} \psi_{v_\ell,v'_\ell}(\sigma, 1)\nu_\ell(\sigma)}{\max_{\sigma\in\Sigma} \psi_{v_\ell,v'_\ell}(\sigma, 0)\nu_\ell(\sigma)}. \quad (5)$$

From (3), (4) and (5) it is sufficient to show that

$$\frac{\nu_\ell(1)}{\nu_\ell(0)} = \frac{q_\ell^*(1)}{q_\ell^*(0)}, \quad 1 \le \ell \le d. \quad (6)$$

Now, note that T_ℓ is the same as $T_{SAW}(G(\ell))$ with respect to $X'(\ell)$. Because for each $\ell = 1, \ldots d$, $G'(\ell)$ has one less *unfixed* node than G, the desired result (6) follows by induction hypothesis. ∎

Lemma 2: Consider a distribution on $X = (X_1, \ldots, X_n)$ where X_i are binary variables. Let $p_s = \mathbb{P}[X = s], s \in \Sigma^n$. Let $p_{s|a_2,\ldots,a_d} = \mathbb{P}[X = s | X_2 = a_2, \ldots, X_d = a_d]$ for any $d \ge 1$. Let $S(a_1, \ldots, a_d) = \{s = (s_1, \ldots, s_n) \in \Sigma^n : s_1 = a_1, \ldots, s_d = a_d\}$. Then,

$$\frac{\max_{s\in S(a_1,a_2\ldots,a_d)} p_s}{\max_{s\in S(\hat{a}_1,a_2,\ldots,a_d)} p_s} = \frac{\max_{s\in S(a_1,a_2\ldots,a_d)} p_{s|a_2,\ldots,a_d}}{\max_{s\in S(\hat{a}_1,a_2,\ldots,a_d)} p_{s|a_2,\ldots,a_d}}.$$

Proof: Let $q = \mathbb{P}(X_2 = a_2, \ldots, X_d = a_d)$. Then, by definition of conditional probability for $s \in S(a_1, a_2, \ldots, a_d) \cup S(\hat{a}_1, a_2, \ldots, a_d)$, $p_s = p_{s|a_2,\ldots,a_d} q$. From this, Lemma follows immediately. ∎

B. Graphs with Good Partitioning Property

Here we discuss graph partitioning property and schemes for finding such partitions. First, a definition.

Definition 2 ((δ, Δ)-decomposition): Given graph $G = (V, E)$, a randomly chosen subset of edges $\mathcal{B} \subset E$ is called (δ, Δ) decomposition of G if the following holds: (a) For any edge $e \in E$, $\mathbb{P}(e \in \mathcal{B}) \le \delta$. (b) Let $S_1, \ldots, S_K$ be connected components of graph $G' = (V, E\backslash\mathcal{B})$ obtained by

removing edges of $\mathcal{B}$ from G. Then, for any such component $S_j, 1 \le j \le K$ and any $u, v \in S_j$ the shortest-path distance between (u, v) in the original graph G is at most Δ with probability 1.

We call a graph admits *good partitioning property* if there exists (δ, Δ)-decomposition for any $\delta > 0$ and Δ independent of n but with possible dependence on δ. The existence of (δ, Δ)-decomposition implies that it is possible to remove δ fraction of edges so that graph *decomposes* into connected components whose *diameter* is small.

We discuss two classes of graphs that posses this property: (1) minor-excluded graphs and (2) graphs with low doubling dimension. Next, we define these graph classes and quickly recall the schemes that provide such decomposition.

Minor-excluded graphs. First we present the definition and then a decomposition scheme for such graphs due to Klein, Plotkin, Rao [2] and Rao [8].

Definition 3 (Minor Exclusion): A graph H is called minor of G if we can transform G into H through an arbitrary sequence of the following two operations: (a) removal of an edge; (b) merge two connected vertices u, v: that is, remove edge (u, v) as well as vertices u and v; add a new vertex and make all edges incident on this new vertex that were incident on u or v. Now, if H is not a minor of G then we say that G excludes H as a minor.

MINOR(G, r, Δ)

(0) Input is graph $G = (V, E)$ and $r, \Delta \in \mathbb{N}$. Initially, $i = 0$, $G_0 = G$, $\mathcal{B} = \emptyset$.
(1) For $i = 0, \ldots, r-1$, do the following.
  (a) Let $S_1^i, \ldots, S_{k_i}^i$ be the connected components of G_i.
  (b) For each $S_j^i, 1 \le j \le k_i$, pick an arbitrary node $v_j \in S_j^i$.
    ◦ Create a breadth-first search tree $\mathcal{T}_j^i$ rooted at v_j in S_j^i.
    ◦ Choose a number L_j^i uniformly at random from $\{0, \ldots, \Delta - 1\}$.
    ◦ Let $\mathcal{B}_j^i$ be the set of edges at level $L_j^i, \Delta + L_j^i, 2\Delta + L_j^i, \ldots$ in $\mathcal{T}_j^i$.
    ◦ Update $\mathcal{B} = \mathcal{B} \cup_{j=1}^{k_i} \mathcal{B}_j^i$.
  (c) set $i = i + 1$.
(3) Output $\mathcal{B}$ and graph $G' = (V, E \backslash \mathcal{B})$.

As stated above, the basic idea is to use the following step recursively (upto depth r of recursion): in each connected component, say S, choose a node arbitrarily and create a breadth-first search tree, say $\mathcal{T}$. Choose a number, say L, uniformly at random from $\{0, \ldots, \Delta - 1\}$. Remove (add to $\mathcal{B}$) all edges that are at level $L + k\Delta, k \ge 0$ in $\mathcal{T}$. Clearly, the total running time of such an algorithm is $O(r(n + |E|))$ for a graph $G = (V, E)$ with $|V| = n$. For message passing implementation of this scheme, nodes can elect the arbitrary root with the help of randomization (e.g. nodes choose random number from large enough set and the one with maximum value becomes the root) and then the breadth-first search tree can be created trivially through spreading message iteratively. Therefore, it is easy to see that the algorithm MINOR can be made message passing for any G.

The algorithm MINOR(G, r, Δ) is designed to provide a good decomposition for class of graphs that exclude a connected graph S of r nodes. Figure 2 explains the algorithm for a line-graph of $n = 9$ nodes, which excludes $K_{2,2}$ as a minor. The example is about a sample run of MINOR$(G, 2, 3)$ (Figure 2 shows the first iteration of the algorithm).

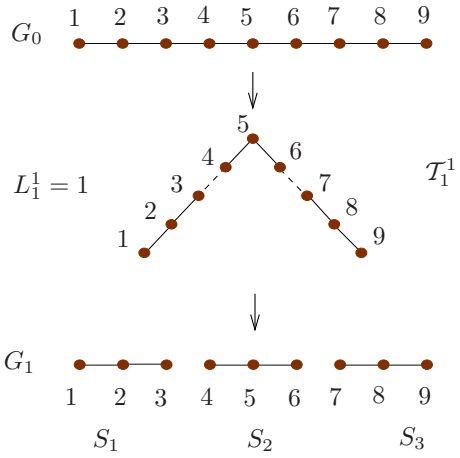

Fig. 2. The first of two iterations in execution of MINOR$(G, 2, 3)$ is shown.

Lemma 3: If G excludes a graph S with r nodes as a minor, then algorithm MINOR(G, r, Δ) outputs $\mathcal{B}$ which is $(r/\Delta, O(\Delta))$-decomposition of G.

It is known that Planar graph excludes $K_{3,3}$ as a minor. Hence, Lemma 3 implies the following.

Corollary 4: Given a planar graph G, the algorithm MINOR$(G, 3, \Delta)$ produces $(3/\Delta, O(\Delta))$-decomposition for any $\Delta \ge 1$.

III. APPROXIMATE MAP

Now, we describe algorithm to compute MAP approximately. Essentially, the algorithm does the following: given G, decompose it into (small) components $S_1, \ldots, S_K$ by removing (few) edges $\mathcal{B} \subset E$ (we use a term DECOMP to obtain such $\mathcal{B}$; for minor-excluded graph use MINORand LOWDDfor graphs with low doubling dimension). Then, compute an approximate MAP assignment by computing exact MAP restricted to the components. This exact computation for each component is performed through a message passing mechanism using the equivalence stated in Theorem 1: essentially, growing self-avoiding walk tree is just sending messages along a breadth-first search tree; computation over a self-avoiding walk tree is essentially standard max-product (message passing) algorithm. The computation time and performance of the algorithm depends on property of decomposition scheme. We describe algorithm for any graph G.

MODE(G)

(1) Use DECOMP(G) to obtain $\mathcal{B} \subset E$ such that
   (a) $G' = (V, E \backslash \mathcal{B})$ is made of connected components $S_1, \ldots, S_K$.
(2) For each connected component $S_j, 1 \leq j \leq K$, do the following:
   (a) Compute exact MAP $\mathbf{x}^{*,j}$ for component S_j, where $\mathbf{x}^{*,j} = (x_i^{*,j})_{i \in S_j}$.
   (b) Computation of $\mathbf{x}^{*,j}$ is performed by growing self-avoiding walk tree for each node in S_j restricted to induced graph by nodes of S_j using a message passing mechanism; then computing max-marginal on self-avoiding walk tree using message passing mechanism (i.e. standard max-product algorithm on self-avoiding walk tree).
(3) Produce output $\widehat{\mathbf{x}^*}$, which is obtained by assigning values to nodes using $\mathbf{x}^{*,j}, 1 \leq j \leq K$. This is clearly local operation.

A. Analysis of MODE

Here, we analyze performance of MODE for any G. Later, we will specialize our analysis for minor excluded G when it uses MINOR as the DECOMP algorithm.

Lemma 5: If G has maximum vertex degree D and the DECOMP(G) produces $\mathcal{B}$ that is (δ, Δ)-decomposition, then

$$\mathbb{E}\left[\mathcal{H}(\mathbf{x}^*) - \mathcal{H}(\widehat{\mathbf{x}^*})\right] \leq \delta(D+1)\mathcal{H}(\mathbf{x}^*),$$

where expectation is w.r.t. the randomness in $\mathcal{B}$. Further, MODE takes time $O(nD2^{D^{\Delta+1}}) + T_{\text{DECOMP}}$.

When G excludes minor, then we use MINOR as decomposition scheme. The above lemma implies the following result.

Theorem 6: Let G exclude a graph of r nodes as a minor and have D as the maximum vertex degree. Given $\varepsilon > 0$, use MODE algorithm with MINOR(G, r, Δ) where $\Delta = \lceil \frac{r(D+1)}{\varepsilon} \rceil$. Then,

$$(1-\varepsilon)\mathcal{H}(\mathbf{x}^*) \leq \mathbb{E}[\mathcal{H}(\widehat{\mathbf{x}^*})] \leq \mathcal{H}(\mathbf{x}^*).$$

Further, algorithm takes $n \cdot C(D, \varepsilon)$ time, where constant $C(D, \varepsilon) = D2^{D^{O(rD/\varepsilon)}}$.

Proof: The proof follows from Lemma 5, recalling that the total running time of MINOR, is of the order of number of edges which is $\leq 2Dn$. ∎

B. Proof of Lemma 5

The proof of Lemma 5 will use the Lemmas 7-9 stated below. We will complete the proof of Lemma 5 at the end of this section.

Lemma 7: Consider a connected graph $G = (V, E)$ with $|V| = n$ nodes and $|E| = n - 1 + k$ edges, $k \geq 0$. Then for any $v \in V$, $|T_{SAW}(G, v)| \leq (n + k - 1)2^{k+1}$. Further, there exists a graph with $n - 1 + k$ edges with $k < n/2$ so that for any node $v \in V$, $|T_{SAW}(G, v)| \geq n2^{k-2}$.

Proof: The proof is divided into two parts. We first provide the proof of lower bound. Consider a line graph of n nodes (with $n-1$ edges). Now add $k < n/2$ edges as follows. Add an edge between 1 and n. Remaining $k - 1$ edges are added between node pairs: $(2, 4), (4, 6), \ldots, (2(k-2), 2(k-1)), (2(k-1), 2k)$. Consider any node, say v. It is easy to see that there are at least 2^{k-2} different ways in which one can start walking on the graph from node v towards node 1, cross from 1 to n via edge $(1, n)$ and then come back to node v. Each of these different loops, starting from v and ending at v creates 2 distinct paths in the self-avoiding walk tree of length at least$\frac{n}{2}$. Thus, the size of self-avoiding walk tree of each node is at least $n2^{k-2}$ for each node. This completes the proof of lower bound.

Now, we prove the upper bound of $n2^{k+1}$ on the size of self-avoiding walk tree for each node $v \in V$. Given that G is connected, we can divide the edge set $E = E_T \cup E_k$ where $E_k = \{e_1, \ldots, e_k\}$ and $T = (V, E_T)$ forms a spanning tree of G. Let $\mathcal{S}$ be the set of all subsets of $E_k = \{e_1, \ldots, e_k\}$ (there are 2^k of them including empty set). Now fix a vertex $v \in V$ and we will concentrate on $T_{SAW}(G, v)$. Consider any $u \in V$ (can be v) and $S \in \mathcal{S}$. Next, we wish to count number of paths in $T_{SAW}(G, v)$ that end at (a copy of) u (however, u need not be a leaf), contain all edges in S but none from $E_k \backslash S$. We claim the following.

Claim. There can be at most one path of $T_{SAW}(G, v)$ from v to (a copy of) u and containing all edges from S but none from $E_k \backslash S$.

Proof: To prove the above claim, suppose it is not true. Then there are at least two distinct paths from v to u that contain all edges in S (but none from $E_k \backslash S$). Consider the symmetric difference of these two paths (in terms of edges). This symmetric difference must be a non-empty subset of E_T and also contain a loop (as the two paths have same starting and ending point). But this is not possible as $T = (V, E_T)$ is a tree and it does not contain a loop. This contradicts our assumption and proves the claim. ∎

Given the above claim, for any node u, clearly the number of distinct paths from node v to (a copy of) u in $T_{SAW}(G, v)$ are at most 2^k. Now each edge has two end points. For each appearance of an edge of G in $T_{SAW}(G, v)$, a distinct path from v to one of its end point must appear in $T_{SAW}(G, v)$. From above claim, this can happen at most $2 \times 2^k = 2^{k+1}$. There are $n + k - 1$ edges of G in total. Thus, net number of edges that can appear in $T_{SAW}(G, v)$ is at most $(n + k - 1)2^{k+1}$; thus completing the proof of Lemma 7. ∎

Lemma 8: Let $\psi_{ij}^U = \max_{(x,x') \in \{0,1\}^2} \psi_{ij}(x, x')$, $\psi_{ij}^L = \min_{(x,x') \in \{0,1\}^2} \psi_{ij}(x, x')$ and G have maximum vertex degree D. Then

$$\mathcal{H}(\mathbf{x}^*) \geq \frac{1}{D+1}\left[\sum_{(i,j) \in E} \log \psi_{ij}^U\right] \geq \frac{1}{D+1}\left[\sum_{(i,j) \in E} \log \psi_{ij}^U - \log \psi_{ij}^L\right].$$

Proof: Assign weight $w_{ij} = \log \psi_{ij}^U$ to an edge $(i, j) \in E$. Using Vizing's theorem and Pigeon hole principle, we obtain that there exists a matching $M \subset E$ such that

$$\sum_{(i,j) \in M} \log \psi_{ij}^U \geq \frac{1}{D+1}\left(\sum_{(i,j) \in E} \log \psi_{ij}^U\right).$$

Now, consider an assignment $\mathbf{x}^M$ as follows: for each $(i,j) \in M$ set $(x_i^M, x_j^M) = \arg\max_{(x,x') \in \{0,1\}^2} \psi_{ij}(x,x')$; for remaining $i \in V$, set x_i^M to some value in Σ arbitrarily. Note that for above assignment to be possible, we have used matching property of M. Therefore, we have

$$\begin{aligned}
\mathcal{H}(\mathbf{x}^M) &= \sum_{i \in V} \log \phi_i(x_i^M) + \sum_{(i,j) \in E} \log \psi_{ij}(x_i^M, x_j^M) \\
&= \sum_{i \in V} \log \phi_i(x_i^M) + \sum_{(i,j) \in E \setminus M} \log \psi_{ij}(x_i^M, x_j^M) \\
&\quad + \sum_{(i,j) \in M} \log \psi_{ij}(x_i^M, x_j^M) \overset{(a)}{\geq} \sum_{(i,j) \in M} \log \psi_{ij}(x_i^M, x_j^M) \\
&= \sum_{(i,j) \in M} \log \psi_{ij}^U \geq \frac{1}{D+1} \left[\sum_{(i,j) \in E} \log \psi_{ij}^U \right]. \qquad (7)
\end{aligned}$$

Here (a) follows because $\log \psi_{ij}, \log \phi_i$ are non-negative valued functions. Since $\mathcal{H}(\mathbf{x}^*) \geq \mathcal{H}(\mathbf{x}^M)$ and $\log \psi_{ij}^L \geq 0$ for all $(i,j) \in E$, we obtain the Lemma 8. ■

Lemma 9: Given an MRF G described by (1), the MODE algorithm produces outputs $\widehat{\mathbf{x}^*}$ such that $\mathcal{H}(\mathbf{x}^*) - \sum_{(i,j) \in \mathcal{B}} \left(\log \psi_{ij}^U - \log \psi_{ij}^L \right) \leq \mathcal{H}(\widehat{\mathbf{x}^*}) \leq \mathcal{H}(\mathbf{x}^*)$.

Proof: By definition of MAP $\mathbf{x}^*$, we have $\mathcal{H}(\widehat{\mathbf{x}^*}) \leq \mathcal{H}(\mathbf{x}^*)$. Now, consider the following: let $\Sigma = \{0,1\}$,

$$\begin{aligned}
\mathcal{H}(\mathbf{x}^*) &= \max_{\mathbf{x} \in \Sigma^n} \left[\sum_{i \in V} \log \phi_i(x_i) + \sum_{(i,j) \in E} \log \psi_{ij}(x_i, x_j) \right] \\
&= \max_{\mathbf{x} \in \Sigma^n} \left[\sum_{i \in V} \log \phi_i(x_i) + \sum_{(i,j) \in E \setminus \mathcal{B}} \log \psi_{ij}(x_i, x_j) \right. \\
&\quad \left. + \sum_{(i,j) \in \mathcal{B}} \log \psi_{ij}(x_i, x_j) \right] \\
&\overset{(a)}{\leq} \max_{\mathbf{x} \in \Sigma^n} \left[\sum_{i \in V} \log \phi_i(x_i) + \sum_{(i,j) \in E \setminus \mathcal{B}} \log \psi_{ij}(x_i, x_j) \right. \\
&\quad \left. + \sum_{(i,j) \in \mathcal{B}} \log \psi_{ij}^U \right] \\
&\overset{(b)}{=} \sum_{j=1}^{K} \left[\max_{\mathbf{x}^j \in \Sigma^{|S_j|}} \mathcal{H}(\mathbf{x}^j) \right] + \left[\sum_{(i,j) \in \mathcal{B}} \log \psi_{ij}^U \right] \\
&\overset{(c)}{=} \sum_{j=1}^{K} \mathcal{H}(\mathbf{x}^{*,j}) + \left[\sum_{(i,j) \in \mathcal{B}} \log \psi_{ij}^U \right] \\
&\overset{(d)}{\leq} \mathcal{H}(\widehat{\mathbf{x}^*}) + \left[\sum_{(i,j) \in \mathcal{B}} \log \psi_{ij}^U - \log \psi_{ij}^L \right]. \qquad (8)
\end{aligned}$$

We justify (a)-(d) as follows: (a) holds because for each edge $(i,j) \in \mathcal{B}$, we have replaced its effect by maximal value $\log \psi_{ij}^U$; (b) holds because by placing constant value $\log \psi_{ij}^U$ over $(i,j) \in \mathcal{B}$, the maximization over G decomposes into maximization over the connected components of $G' = (V, E \setminus \mathcal{B})$; (c) holds by definition of $\mathbf{x}^{*,j}$ and (d) holds because when we obtain global assignment $\widehat{\mathbf{x}^*}$ from $\mathbf{x}^{*,j}, 1 \leq j \leq K$ and compute its global value, the additional terms get added for each $(i,j) \in \mathcal{B}$ which add at least $\log \psi_{ij}^L$ amount. ■

Proof of Lemma 5. From Lemma 9, Lemma 8 and definition of (δ, Δ)-decomposition, we have the following.

$$\begin{aligned}
\mathbb{E}\left[\mathcal{H}(\mathbf{x}^*) - \mathcal{H}(\widehat{\mathbf{x}^*}) \right] &\leq \mathbb{E}\left[\sum_{(i,j) \in \mathcal{B}} (\log \psi_{ij}^U - \log \psi_{ij}^L) \right] \\
&= \sum_{(i,j) \in E} \mathbb{P}((i,j) \in \mathcal{B}) (\log \psi_{ij}^U - \log \psi_{ij}^L) \\
&\leq \delta \left[\sum_{(i,j) \in E} (\log \psi_{ij}^U - \log \psi_{ij}^L) \right] \leq \delta(D+1) \mathcal{H}(\mathbf{x}^*). \qquad (9)
\end{aligned}$$

The running time bound is implied as follows: the decomposition algorithm takes time T_{DECOMP}. The exact evaluation of MAP in each component takes time of the order of size of self-avoiding walk tree. Each component has at most D^Δ nodes and $D^{\Delta+1}$ edges. Therefore, Lemma 7 implies the desired bound since there are at most $O(n)$ components. □

IV. CONCLUSION

In this paper, we presented message passing approximate inference algorithm for computing MAP in arbitrary pair-wise binary MRF. The algorithm provides arbitrarily good approximation for minor-excluded graphs.

In principle, this algorithm extends for computing MAP for any pair-wise MRF representing exponential family since it can be reduced to computing Maximum Weighted Independent Set. It will be of interest to obtain efficient such reduction in order.

REFERENCES

[1] M. Bayati, D. Shah and M. Sharma, "Maximum Weight Matching via Max-Product Belief Propagation," *IEEE ISIT*, 2005.

[2] P. Klein, S. Plotkin and S. Rao, "Excluded minors, network decomposition, and multicommodity flow," *ACM STOC*, 1993.

[3] V. Kolmogorov, "Convergent Tree-reweighted Message Passing for Energy Minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[4] V. Kolmogorov and M. Wainwright, "On optimality of tree-reweighted max-product message-passing," *Uncertainty in Artificial Intelligence*, 2005.

[5] N. Madras and G. Slade, "The Self-Avoiding Walk," *Birkhauser, Boston*, 1993.

[6] C. Moallemi and B. Van Roy, "Convergence of the min-sum message passing algorithm for quadratic optimization. *Stanford University Technical report*, 2006.

[7] J. Pearl, "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference," San Francisco, CA: Morgan Kaufmann, 1988.

[8] S. Rao, "Small distortion and volume preserving embeddings for Planar and Euclidian metrics," *ACM SCG*, 1999.

[9] S. C. Tatikonda and M. I. Jordan, "Loopy Belief Propagation and Gibbs Measure," *Uncertainty in Artificial Intelligence*, 2002.

[10] M. Wainwright and M. Jordan, " Graphical models, exponential families, and variational inference," UC Berkeley, Dept. of Statistics, Technical Report 649, 2003.

[11] M. J. Wainwright, T. Jaakkola and A. S. Willsky, "Tree-based reparameterization framework for analysis of sum-product and related algorithms," *IEEE Transactions on Information Theory*, 2003.

[12] M. J. Wainwright, T. S. Jaakkola and A. S. Willsky, "MAP estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches," *IEEE Transactions on Information Theory*, 51(11), 2005.

[13] D. Weitz, "Counting independent sets up to the tree threshold," *ACM STOC*, 2006.

[14] J. Yedidia, W. Freeman and Y. Weiss, "Generalized Belief Propagation," *Mitsubishi Elect. Res. Lab.*, TR-2000-26, 2000.