

Message-Passing for Approximate MAP Inference with Latent Variables

Authored by:

Piyush Rai
Hal Daume
Jiarong Jiang

Abstract

We consider a general inference setting for discrete probabilistic graphical models where we seek maximum a posteriori (MAP) estimates for a subset of the random variables (max nodes), marginalizing over the rest (sum nodes). We present a hybrid message-passing algorithm to accomplish this. The hybrid algorithm passes a mix of sum and max messages depending on the type of source node (sum or max). We derive our algorithm by showing that it falls out as the solution of a particular relaxation of a variational framework. We further show that the Expectation Maximization algorithm can be seen as an approximation to our algorithm. Experimental results on synthetic and real-world datasets, against several baselines, demonstrate the efficacy of our proposed algorithm.

1 Paper Body

Probabilistic graphical models provide a compact and principled representation for capturing complex statistical dependencies among a set of random variables. In this paper, we consider the general maximum a posteriori (MAP) problem in which we want to maximize over a subset of the variables (max nodes, denoted X), marginalizing the rest (sum nodes, denoted Z). This problem is termed as the Marginal-MAP problem. A typical example is the minimum Bayes risk (MBR) problem [1] where the goal is to find an assignment x^* which optimizes a loss $\ell(x^*, x)$ with regard to some usually unknown truth x . Since x is latent, we need to marginalize it before optimizing with respect to x^* . Although the specific problems of estimating marginals and estimating MAP individually have been studied extensively [2, 3, 4], similar developments for the more general problem of simultaneous marginal and MAP estimation are lacking. More recently, [5] proposed a method based optimizing a variational objective on specific graph structures, and is a simultaneous development as the method we propose in this paper (please refer to the supplementary material

for further details and other related work). This problem is fundamentally difficult. As mentioned in [6, 7], even for a tree-structured model, we cannot solve the Marginal-MAP problem exactly in poly-time unless $P = NP$. Moreover, it has been shown [8] that even if a joint distribution $p(x, z)$ belongs to the exponential family, the P corresponding marginal distribution $p(x) = \sum_z p(x, z)$ is in general not exponential family (with a very short list of exceptions, such as Gaussian random fields). This means that we cannot directly apply algorithms for MAP inference to our task. Motivated by this problem, we propose a hybrid message passing algorithm which is both intuitive and justified according to variational principles. Our hybrid message passing algorithm uses a mix of sum and max messages with the message type depending on the source node type. Experimental results on chain and grid structured synthetic data sets and another real-world dataset show that our hybrid message-passing algorithm works favorably compared to standard sumproduct, standard max-product, or the Expectation-Maximization algorithm which iteratively provides MAP and marginal estimates. Our estimates can be further improved by a few steps of local

search [6]. Therefore, using the solution found by our hybrid algorithm to initialize some local search algorithms largely improves the performance on both accuracy and convergence speed, compared to the greedy stochastic search method described in [6]. We also give an example in Sec. 5 of how our algorithm can also be used to solve other practical problem which can be cast under the Marginal-MAP framework. In particular, the Minimum Bayes Risk [9] problem for decomposable loss-functions can be readily solved under this framework.

2

Problem Setting

In our setting, the nodes in a graphical model with discrete random variables are divided into two sets: max and sum nodes. We denote a graph $G = (V, E)$, $V = X \cup Z$ where X is the set of nodes for which we want to compute the MAP assignment (max nodes), and Z is the set of nodes for which we need the marginals (sum nodes). Let $x = \{x_1, \dots, x_m\}$ ($x_s \in X_s$), $z = \{z_1, \dots, z_n\}$ ($z_s \in Z_s$) be the random variables associated with the nodes in X and Z respectively. The exponential family distribution p over these random variables is defined as follows: $p(x, z) = \exp[h(\theta, \phi(x, z))] / A(\theta)$ where $\phi(x, z)$ is the sufficient statistics of the enumeration of all node assignments, and θ is the vector of canonical or exponential parameters. $A(\theta) = \sum_{x, z} \exp[h(\theta, \phi(x, z))]$ is the log-partition function. In this paper, we consider only pairwise node interactions and use standard overcomplete representation of the sufficient statistics [10] (defined by indicator function I later). The general MAP problem can be formalized as the following maximization problem:

$$\begin{aligned} (1) \quad & \max_x p(x, z) \\ & \text{with corresponding marginal probabilities of the } z \text{ nodes, given } x. \\ (2) \quad & p(z_s = x_s) = p(z_s = x_s), s = 1, \dots, n \end{aligned}$$

$$Z\{z_s\}$$

Before proceeding, we introduce some notations for clarity of exposition: Subscripts s, u, t , etc. denote nodes in the graphical model. z_s, x_s are sum and max random variables respectively, associated with some node s . v_s can be either a sum (z_s) or a max random (x_s) variable, associated with some node s . $N(s)$ is the set of neighbors of node s . X_s, Z_s, V_s are the state spaces from which x_s, z_s, v_s take values. 2.1

Message Passing Algorithms

The sum-product and max-product algorithms are standard message-passing algorithms for inferring marginal and MAP estimates respectively in probabilistic graphical models. Their idea is to store a belief state associated with each node, and iteratively passing messages between adjacent nodes, which are used to update the belief states. It is known [11] that these algorithms are guaranteed to converge to the exact solution on trees or polytrees. On loopy graphs, they are no longer guaranteed to converge, but they can still provide good estimates when converged [12]. In the standard sum product algorithm, the message M_{ts} passed from node s to one of its neighbors t is as follows: $M_{ts}(v_t) = \sum_{v_s} \psi(v_s, v_t) \prod_{u \in N(s) \setminus t} M_{su}(v_s)$ (3) $M_{ts}(v_t) = \max_{v_s} \psi(v_s, v_t) \prod_{u \in N(s) \setminus t} M_{su}(v_s)$

$$\frac{1}{Z} \sum_{v_s} \psi(v_s, v_t) \prod_{u \in N(s) \setminus t} M_{su}(v_s)$$

where Z is a normalization constant. When the messages converge, i.e. $\{M_{ts}, M_{st}\}$ does not change for every pair of nodes s and t , the belief (psuedomarginal distribution) for the node s is given by $\psi_s(v_s) = \frac{1}{Z} \sum_{v_t} \psi(v_s, v_t) \prod_{u \in N(s) \setminus t} M_{su}(v_s)$. The outgoing messages for max product algorithm have the same form but with a maximization instead of a summation in Eq. (3). After convergence, the MAP assignment for each node is the assignment with the highest max-marginal probability. On loopy graphs, the tree-weighted sum and max product [13, 14] can help find the upper bound of the marginal or MAP problem. They decompose the loopy graph into several spanning trees and reweight the messages by the edge appearance probability. 2

2.2

Local Search Algorithm

Eq (1) can be viewed as doing a variable elimination for z nodes first, followed by a maximization over x . Its maximization step may be performed using heuristic search techniques [7, 6]. Eq (2) can be computed by running standard sum-product over z , given the MAP x assignments. In [6], the assignment for the MAP nodes are found by greedily searching the best neighboring assignments which only differs on one node. However, the hybrid algorithm we propose allows simultaneously approximating both Eq (1) and Eq (2).

3

HYBRID MESSAGE PASSING

In our setting, we wish to compute MAP estimates for one set of nodes and marginals for the rest. One possible approach is to run standard sum/max product algorithms over the graph, and find the most-likely assignment for each max node according to the maximum of sum or max marginals¹. These naive approaches have their own shortcomings; for example, although using standard maxproduct may perform reasonably when there are many max nodes,

it inevitably ignores the effect of sum nodes which should ideally be summed over. This is analogous to the difference between EM for Gaussian mixture models and K-means. (See Sec. 6) 3.1

ALGORITHM

We now present a hybrid message-passing algorithm which passes sum-style or max-style messages based on the type of nodes from which the message originates. In the hybrid message-passing algorithm, a sum node sends sum messages to its neighbors and a max node sends max messages. The type of message passed depends on the type of source node, not the destination node. More specifically, the outgoing messages from a source node are as follows: ? Message from sum node t to any neighbor s : $\prod_{t \in N(s)} \exp[\psi_{st}(v_s, z_t)] \prod_{t \in N(s)} \psi_{st}(v_s, z_t)$

(4)

$\prod_{t \in N(s)} \psi_{st}(v_s, z_t)$

? Message from max node t to any neighbor s : $\prod_{t \in N(s)} \exp[\psi_{st}(v_s, x_t)] \prod_{t \in N(s)} \max_{x_t} \psi_{st}(v_s, x_t)$

Y

$\prod_{t \in N(s)} \psi_{st}(v_s, x_t)$

$\prod_{t \in N(s)} \max_{x_t} \psi_{st}(v_s, x_t)$

? ?

(5)

?

and ψ_1, ψ_2 are normalization constants. Algo 1 shows the procedure to do hybrid message-passing. Algorithm 1 Hybrid Message-Passing Algorithm Inputs: Graph $G = (V, E)$, $V = X \cup Z$, potentials $\psi_s, \psi_{st}, \psi_{st}(s, t) \in \mathbb{R}$. 1. Initialize the messages to some arbitrary value. 2. For each node $s \in V$ in G , do the following until messages converge (or maximum number of iterations reached) ? If $s \in X$, update messages by Eq.(5). ? If $s \in Z$, update messages by Eq.(4). 3. Compute the local belief Q for each node s . $Q_s(x_s) = \exp\{\psi_s(x_s)\} \prod_{t \in N(s)} \psi_{st}(x_s, z_t)$ 4. For all $x_s \in X$, return $\arg \max_{x_s} Q_s(x_s)$ 5. For all $z_s \in Z$, return $Q_s(z_s)$. When there is only a single type of node in the graph, the hybrid algorithm reduces to the standard max or sum-product algorithm. Otherwise, it passes different messages simultaneously and gives an approximation to the MAP assignment on max nodes as well as the marginals on sum nodes. On the loopy graphs, we can also apply this scheme to pass hybrid tree-reweighted messages between nodes to obtain marginal and MAP estimates. (See Appendix C of the supplementary material) 1 Running the standard sum-product algorithm and choosing the maximum likelihood assignment for the max nodes is also called maximum marginal decoding [15, 16].

3

3.2

VARIATIONAL DERIVATION

In this section, we show that the Marginal-MAP problem can be framed under a variational framework, and the hybrid message passing algorithm turns out to be a solution of it. (a detailed derivation is in Appendix A of the supplementary material). To see this, we construct a new graph G_x with $x_s \in \mathcal{X}$

$\mathbf{x} = \mathbf{x}_1 \dots \mathbf{x}_m$, so the log-partition function $A(\mathbf{x})$ of the graph assignments fixed to be \mathbf{x} is $A(\mathbf{x}) = \log p(\mathbf{x}, \mathbf{z}) + \log A(\mathbf{z}) = \log p(\mathbf{x}) + \text{const}$ (6)

As the constant only depends on the log-partition function of the original graph and does not vary with different assignments of MAP nodes, $A(\mathbf{x})$ exactly estimates the log-likelihood of assignment \mathbf{x} . Therefore $\arg\max_{\mathbf{x}} A(\mathbf{x}) = \arg\max_{\mathbf{x}} \log p(\mathbf{x})$. Moreover, $A(\mathbf{x})$ can be approximated by the following [10]: (7) $A(\mathbf{x}) \approx \sum_i h_i + \text{HBethe}(\mathbf{z}) - \sum_{\mathbf{x}} M(\mathbf{x})$

where $M(\mathbf{x})$ is the following marginal polytope of graph $G_{\mathbf{x}} : \mathbf{z} \rightarrow \mathbf{x}$ fixed to its assignment \mathbf{x} with $\mathbf{z} = \mathbf{z}_s, \mathbf{z}_t$ (vs, vt): marginals

$$M(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x}_s = \mathbf{x}_t \\ 0 & \text{else} \end{cases}$$

(8)

Recall, vs stands for \mathbf{z}_s or \mathbf{z}_t . $\text{HBethe}(\mathbf{z})$ is the Bethe energy of the graph: $\text{HBethe}(\mathbf{z}) =$

$$\begin{aligned} & \sum_s H_s(\mathbf{z}_s) - \sum_{\mathbf{z}} \sum_{\mathbf{z}'} \text{Ist}(\mathbf{z}, \mathbf{z}'), H_s(\mathbf{z}_s) = \sum_{\mathbf{z}} \sum_{\mathbf{z}'} \sum_{\mathbf{z}''} \sum_{\mathbf{z}'''} \log \sum_{\mathbf{z}'''} \sum_{\mathbf{z}''} \sum_{\mathbf{z}} \sum_{\mathbf{z}'} \text{Ist}(\mathbf{z}, \mathbf{z}') \log \sum_{\mathbf{z}'} \sum_{\mathbf{z}''} \sum_{\mathbf{z}'''} \sum_{\mathbf{z}''} \sum_{\mathbf{z}} \sum_{\mathbf{z}'} \text{Ist}(\mathbf{z}, \mathbf{z}') \log \sum_{\mathbf{z}'} \sum_{\mathbf{z}''} \sum_{\mathbf{z}'''} \sum_{\mathbf{z}''} \sum_{\mathbf{z}} \sum_{\mathbf{z}'} \text{Ist}(\mathbf{z}, \mathbf{z}') \end{aligned}$$

(9)

For readability, we use \sum, \max to subsume the node and pairwise marginals for sum/max nodes and $\sum_{\mathbf{z}}, \max_{\mathbf{z}}$, $\sum_{\mathbf{z}, \mathbf{z}'}, \max_{\mathbf{z}, \mathbf{z}'}$ are the pairwise marginals for edges between different types of nodes. The direction here is used to be consistent with the distinction of the constraints as well as the messages. Solving the Marginal-MAP problem is therefore equivalent to solving the following optimization problem: \max

$$\begin{aligned} & \sum_i h_i + \text{HBethe}(\mathbf{z}) - \sum_{\mathbf{x}} M(\mathbf{x}) \\ & \sum_{\mathbf{z}} \sum_{\mathbf{z}'} \text{Ist}(\mathbf{z}, \mathbf{z}') \log \sum_{\mathbf{z}'} \sum_{\mathbf{z}''} \sum_{\mathbf{z}'''} \sum_{\mathbf{z}''} \sum_{\mathbf{z}} \sum_{\mathbf{z}'} \text{Ist}(\mathbf{z}, \mathbf{z}') \log \sum_{\mathbf{z}'} \sum_{\mathbf{z}''} \sum_{\mathbf{z}'''} \sum_{\mathbf{z}''} \sum_{\mathbf{z}} \sum_{\mathbf{z}'} \text{Ist}(\mathbf{z}, \mathbf{z}') \end{aligned}$$

(10)

$\mathbf{z}_{\text{other}}$ contains all other node/pairwise marginals except \mathbf{z}_{max} . The Bethe entropy terms can be written as (H is the entropy and I is mutual information) $\text{HBethe}(\mathbf{z}) = H(\mathbf{z}_{\text{max}}) + H(\mathbf{z}_{\text{sum}}) - I(\mathbf{z}_{\text{max}}; \mathbf{z}_{\text{sum}})$

Another plausible approach to solve the Marginal MAP problem is by the Expectation Maximization (EM) algorithm [17], typically used for maximum likelihood parameter estimation in latent variable models. In our setting, the variables Z correspond to the latent variables. We now show one way of approaching this problem by applying the sum-product and max-product algorithms in the E and M step respectively. To see this, let us first define $F(p, x) = E_p[\log p(x, z)] + H(p(z))$ (12) where $H(p) = -E_p[\log p(z)]$. Then EM can be interpreted as a joint maximization of the function F [18]: At iteration t , for the E-step, $p^t(z)$ is set to be the p that maximizes $F(p, x^{(t-1)})$ and for the M-step, $x^{(t)}$ is the x that maximizes $F(p^t, x)$. Given F , the following two properties show P that jointly maximizing function F is equivalent to maximizing the objective function $p(x) = \sum_z p(x, z)$. 1. With the value of x fixed in function F , the unique solution to maximizing $F(p, x)$ is given by $p^*(z) = p(z|x)$. 2. If $p^*(z) = p(z|x)$, then $F(p, x) = \log p(x) = \log \sum_z p(x, z)$.

4.1

Expectation Maximization via Message Passing

Now we can derive the EM algorithm for solving the Marginal-MAP problem by jointly maximizing function F . In the E-step, we need to estimate $p^*(z) = p(z|x)$ given x . This can be done by fixing x values at their MAP assignments and running the sum-product algorithm over the resulting graph: The M-step works by maximizing $E_p[(z - x^*) \log p(x, z)]$, where x^* is the assignment given by the previous M-step. This is equivalent to maximizing $E_z[p^*(z - x^*) \log p(x - z)]$, as P the $\log p^*(z)$ term in the maximization is independent of x . $\max_x E_z[p^*(z - x^*) \log p(x - z)] = \max_x \sum_z p(z - x^*) \log p(x, z)$, which in the overcomplete representation [10] can be approximated by $\sum_i X_i$

$$\begin{aligned} & \sum_i X_i \\ & \sum_i s_i + \\ & X \\ & \sum_j Z_j \\ & \sum_j t_j \sum_{ij} \theta_{st;ij} \theta_{Is;i} (x_s) + \\ & X \\ & X \\ & \sum_{st;ij} \theta_{st;ij} \theta_{Is;i} (x_s, x_t) + C \\ & (s, t) \in E, s, t \in X(i, j) \end{aligned}$$

where C subsumes the terms irrelevant to the maximization over x , t is the pseudo-marginal of node t given x . Then, the M-step amounts to running the max product algorithm with potentials on x nodes modified according to Eq. (13). Summarizing, the EM algorithm for solving marginal-MAP estimation can be interpreted as follows: E-step: Fix x_s to be the MAP assignment value from iteration $(k - 1)$ and run sum product to get beliefs on sum nodes z_s , say $t_s, t_s \in Z$.

By directly applying Jensen's inequality to the objective function $\max_x \log \sum_z p(x, z)$ The proofs are straightforward following Lemma 1 and 2 in [18] page 4-5. More details are in Appendix B of the supplementary material A detailed derivation is in Appendix B.4 of the supplementary material

$\mathcal{V} = (\mathcal{V}, \mathcal{E})$ only containing the max nodes. $\mathcal{V} = \mathcal{X}$ and $\mathcal{E} = \emptyset$. M-step: Build a new graph $G = \{(s, t) \mid (s, t) \in \mathcal{E}, s, t \in \mathcal{X}\}$. For each max node s in the graph, set its potential as $P = \sum_{t \in \mathcal{Z}} \sum_{j \in \mathcal{Z}} \psi_{st;j}(s, t)$, where $t \in \mathcal{Z}$ and $(s, t) \in \mathcal{E}$. Run $\psi_{st;j} = \psi_{st;j}(s, t)$. max product over this new graph and update the MAP assignment. 4.2

Relationship with the Hybrid Algorithm

Apart from the fact that the hybrid algorithm passes different messages simultaneously and EM does it iteratively, to see the connection with the hybrid algorithm, let us first consider the message passed in the E-step at iteration k . x_s are fixed at the last assignment which maximizes the message $\psi_{st;j}(k)$ at iteration $k \geq 1$, denoted as x_s^k here. The $\mu_{st;j}$ are the messages computed at iteration $k \geq 1$. $Y(k) = \sum_{t \in \mathcal{Z}} \sum_{j \in \mathcal{Z}} \psi_{st;j}(k) \mu_{st;j}(k)$. Eq (13) becomes

Now assume there exists an iterative algorithm which, at each iteration, computes the messages $\mu_{st;j}^k$. Eq (13) then used in both steps of the message-passing variant of the EM algorithm, denoted M becomes $Y(k) = \sum_{t \in \mathcal{Z}} \sum_{j \in \mathcal{Z}} \psi_{st;j}(k) \mu_{st;j}^k$. ut $M(z_s) = \sum_{t \in \mathcal{Z}} \sum_{j \in \mathcal{Z}} \exp[\psi_{st;j}(z_s, x_s^k) + \mu_{st;j}^k] M(x_s^k)$

So the max nodes (x_s) should pass the max messages to its neighbors (z_s), which is what the hybrid message-passing algorithm does. In the M-step for EM (as discussed in Sec. 4), all the sum nodes t are removed P from the graph and the parameters of the adjacent max nodes are modified as: $\psi_{st;i} = \psi_{st;i} + \sum_{j \in \mathcal{Z}} \psi_{st;j} \mu_{st;j}^k$. $\mu_{st;j}$ is computed by the sum product at the E-step of iteration k , and these sum messages are used (in form of the marginals $\mu_{st;j}^k$) in the subsequent M-step (with the sum nodes removed). However, a max node may prefer different assignments according to different neighboring nodes. With such uncertainties, especially during the first a few iterations, it is very likely that making hard decisions will directly lead to the bad local optima. In comparison, the hybrid message passing algorithm passes mixed messages instead of making deterministic assignments in each iteration.

5

MBR Decoding

Most work on finding "best" solutions in graphical models focuses on the MAP estimation problem: find the x that maximizes $p(x)$. In many practical applications, one wishes to find an x that minimizes some risk, parameterized by a given loss function. This is the minimum Bayes risk (MBR) setting, which has proven useful in a number of domains, such as speech recognition [9], natural language parsing [19, 20], and machine translation [1]. We are given a loss function $\ell(x, x^*)$ which measures the loss of x assuming x^* is the truth. We assume losses are non-negative. Given this loss function, the minimum Bayes risk solution is the minimizer of Eq (14): $X_{MBR} = \arg \min_x \mathbb{E} \ell(x, x^*)$

x^*

x
 We φ) = P now assume that φ decomposes over the structure of x . In particular, suppose that: $\varphi(x, x' \varphi(x, x'))$, where C is some set of cliques in x , and x denotes the variables associated with $c \in C$ that clique. For example, for Hamming loss, the cliques are simply the set of pairs of vertices of the form (x_i, x'_i) , and the loss simply counts the number of disagreements. Such compositionality is widely assumed in structured prediction algorithms [21, 22]. Assume $l_c(x, x') = L_c(x, x')$. Therefore $l(x, x') = \sum_{c \in C} L_c$. We can then expand Eq (14) into the following:

$$\begin{aligned}
 &= \arg \max_x \varphi \\
 &P(x) \left(\sum_{c \in C} L_c(x, x') \right) \\
 &P(x) \varphi(x, x') = \arg \max \\
 &MBR \varphi = \arg \min \\
 &x' \\
 &x \\
 &X x \\
 &'' \\
 &\exp h \varphi, x_i + \log \\
 &x \\
 &X c \\
 &[L \varphi \varphi(x_c, x'_c)] \varphi A(\varphi) \\
 &\#
 \end{aligned}$$

This resulting expression has exactly the same form as the MAP-with-marginal problem, where φ being the variable being maximized. Fig. 1 shows a x is the variable being marginalized and x simple example of transforming a MAP lattice problem into an MBR problem under Hamming loss. Therefore, we can apply our hybrid algorithm to solve the MBR problem. 6

Average KL-divergence on sum nodes 0.05
 Max Product 0.045
 Sum Product Hybrid Message Passing
 0.04
 KL-divergence
 0.035 0.03 0.025 0.02 0.015 0.01 0.005 0
 10
 20
 30
 40
 50
 60
 70
 80
 90
 % of sum nodes

Figure 1: The Augmented Model For Solving Figure 2: Comparison of Various Algorithms For The MBR Problem Under Hamming Loss over Marginals on 10-Node Chain Graph a 6-node simple lattice

EXPERIMENTS

We perform the experiments on synthetic datasets as well as a real-world protein side-chain prediction dataset [23], and compare our hybrid message-passing algorithm (both its standard belief propagation and the tree-reweighted belief propagation (TRBP) versions) against a number of baselines such as the standard sum/max product based MAP estimates, EM, TRBP, and the greedy local search algorithm proposed in [6].

6.1 Synthetic Data For synthetic data, we first take a 10-node chain graph with varying splits of sum vs max nodes, and random potentials. Each node can take one of the two states (0/1). The node and the edge potentials are drawn from $U \text{ NIFORM}(0,1)$ and we randomly pick nodes in the graph to be sum or max nodes. For this small graph, the true assignment is computable by explicitly maximizing $\prod_{i=1}^n p(x_i) \prod_{(i,j) \in E} p(x_i, x_j)$, where Z is some normalization $z = \sum_{x \in V} \exp(\sum_{(i,j) \in E} \psi_{ij}(x_i, x_j))$ constant and $\psi_{ij}(x_i, x_j) = \exp(\sum_{(i,j) \in E} \psi_{ij}(x_i, x_j))$. First, we compare the various algorithms on the MAP assignments. Assume that the aforementioned maximization gives assignment $x^* = (x_1^*, \dots, x_n^*)$ and some algorithm gives the approximate assignment $x = (x_1, \dots, x_n)$. The metrics we use here are the 0/1 loss and the Hamming loss.

0/1 Loss on a 10-node chain graph

0.7

0.25 max sum hybrid EM max+local search sum+local search hybrid+local search

0.6

0.2

0.4

Error rate

Error rate

0.5

0.3

max sum hybrid EM max+local search sum+local search hybrid+local search

0.15

0.1

0.2 0.05 0.1

0 0.1

0.2

0.3

0.4 0.5 0.6 % of sum nodes

0.7

0.8

0.9

0 0.1

0.2

0.3

0.4 0.5 0.6 % of sum nodes

0.7

0.8
0.9

Figure 3: Comparison of Various Algorithms For MAP Estimates on 10-Node Chain Graph: 0-1 Loss (Left), Hamming Loss (Right) Fig. 3 shows the loss on the assignment of the max nodes. In the figure, as the number of sum nodes goes up, the accuracy of the standard sum-product based estimation (sum) gets better, whereas the accuracy of standard max-product based estimation (max) worsens. However, our hybrid messagepassing algorithm (hybrid), on an average, results in the lowest loss compared to the other baselines, with running times similar to the sum/max product algorithms. We also compare a stochastic greedy search approach described in [6] initialized by the results of sum/max/hybrid algorithm (sum/max/hybrid+local search). As shown in [6], local search with sum product initialization empirically performs better than with max product, so later on, we only compare the results with local search using sum product initialization (LS). Best of the three initialization methods, by starting at the hybrid algorithm results, the search algorithm in very few steps can find 7

log?likelihood of the assignments normalized by hybrid algo
log?likelihood of the assignment normalized by hybrid algorithm
1.001
1.005
1 0.999 0.998
Relative Likelihood
Relative Likelihood
1
0.995
0.99 max
0.997 0.996 0.995 0.994 TR?max TR?sum TR?hybrid LS TR?hybrid+LS
sum 0.985
0.993
hybrid LS
0.992
hybrid+LS 0.98 10%
20%
30%
40%
50%
60%
70%
80%
0.991 0.1
90%
% of sum node
0.2
0.3
0.4 0.5 0.6 % of sum nodes

0.7
0.8
0.9

Figure 4: Approximate Log-Partition Function Scores on a 50-Node Tree (Left) and an 8*10 Grid (Right) Graph Normalized By the Result of Hybrid Algorithm

the local optimum, which often happened to be the global optimum as well. In particular, it only takes 1 or 2 steps of search in the 10-node chain case and 1 to 3 steps in the 50-node tree case. Next, we experiment with the marginals estimation. Fig 2 shows the mean of KL-divergence on the marginals for the three message-passing algorithms (each averaged over 100 random experiments) compared to the true marginals of $p(z|x)$. Greedy search of [6] is not included since it only provides MAP, not marginals. The x-axis shows the percentage of sum nodes in the graph. Just like in the MAP case, our hybrid method consistently produces the smallest KL-divergence compared to others. When the computation of the truth is intractable, the loglikelihood of the assignment can be approximated by the log-partition function with Bethe approximation according to Sec. 3.2. Note that this is exact on trees. Here, we use a 50-node tree with binary node states and 8 * 10 grid with various states $1 \rightarrow Ys \rightarrow 20$. On the grid graph, we apply tree-reweighted sum or max product [14, 13], and our hybrid version based on TRBP. For the edge appearance probability in TRBP, we apply a common approach that use a greedy algorithm to finding the spanning trees with as many uncovered edges as possible until all the edges in the graph are covered at least once. Even if the messagepassing algorithms are not guaranteed to converge on loopy graphs, we can still compare the best result they provide after a certain number of iterations Fig. 4 presents the results. In the tree case, as expected, using hybrid message-passing algorithms's result to initialize the local search algorithm performs the best. On the grid graph, the local search algorithm initialized by the sum product results works well when there are few max nodes, but as the search space grows exponentially with the number of max nodes, so it takes hundreds of steps to find the optimum. On the other hand, because the hybrid TRBP starts in a good area, it consistently achieves the highest likelihood among all four algorithms with fewer extra steps.

6.2 Real-world Data We then experiment with the protein side-chain prediction dataset [23, 24] which consists a set of protein structures for which we need to find lowest energy assignment for rotamer residues. There are two sets of residues: core residues and surface residues. max product 0.7900 0.7564 0.8325 The core residues are the residues which are connected to more than 19 other residues, and the surface ones are the others. Since the MAP results are usually lower on the surface residues than the core sum product 0.6482 0.6069 0.7005 residues nodes [24], we choose the surface residues max product 0.6512 0.6064 0.7078 hybrid 0.6485 0.6051 0.7033 to be max nodes and the core nodes to be the sum TRBP 0.6592 0.6112 0.7174 nodes. The ground truth is given by

the maximum hybrid TRBP 0.6597 0.6140 0.7186 likelihood assignment of the residues, so we do not expect to have a better results on the core nodes, but we hope that any improvement on the accuracy of the surface nodes can make up the loss on the core nodes and thus give a better performance overall. As shown in Table 6.2, the improvements of the hybrid methods on the surface nodes are more than the loss the the core nodes, thus improving the overall performance.

8

2 References

- [1] Shankar Kumar, William Byrne, and Speech Processing. Minimum bayes-risk decoding for statistical machine translation. In HLT-NAACL, 2004. [2] David Sontag and Tommi Jaakkola. New outer bounds on the marginal polytope. In In Advances in Neural Information Processing Systems, 2007. [3] Amir Globerson and Tommi Jaakkola. Fixing max-product: Convergent message passing algorithms for map lp-relaxations. In NIPS, 2007. [4] Pradeep Ravikumar, Alekh Agarwal, and Martin J. Wainwright. Message-passing for graph-structured linear programs: proximal projections, convergence and rounding schemes. In ICML, 2008. [5] Qiang Liu and Alexander Ihler. Variational algorithms for marginal map. In UAI, 2011. [6] James D. Park. MAP Complexity Results and Approximation Methods. In UAI, 2002. [7] D. Koller and N. Friedman. Probabilistic Graphical Models: Principles and Techniques. MIT Press, 2009. [8] Shaul K. Bar-Lev, Daoud Bshouty, Peter Enis, Gerard Letac, I-Li Lu, and Donald Richards. The diagonal multivariate natural exponential families and their classification. In Journal of Theoretical Probability, pages 883?929, 1994. [9] Vaibhava Goel and William J. Byrne. Minimum Bayes-risk automatic speech recognition. Computer Speech and Language, 14(2), 2000. [10] M. J. Wainwright and M. I. Jordan. Graphical Models, Exponential Families, and Variational Inference. Foundations and Trends in Machine Learning, 2008. [11] Judea Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. [12] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Generalized belief propagation. In NIPS, 2000. [13] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. Exact map estimates by tree agreement. In NIPS, 2002. [14] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. Tree-reweighted belief propagation algorithms and approximate ml estimation by pseudo-moment matching. In AISTATS, 2003. [15] Mark Johnson. Why doesnt em find good hmm pos-taggers. In EMNLP, pages 296?305, 2007. [16] Pradeep Ravikumar, Martin J. Wainwright, and Alekh Agarwal. Message-passing for graph-structured linear programs: Proximal methods and rounding schemes, 2008. [17] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM algorithm. Journal of The Royal Statistica Society, 1977. [18] Radford M. Neal and Geoffrey E. Hinton. A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants. In Learning in graphical models, pages 355?368, 1999. [19] Slav Petrov and Dan

Klein. Discriminative log-linear grammars with latent variables. In NIPS, 2008. [20] Ivan Titov and James Henderson. A latent variable model for generative dependency parsing. In IWPT, 2007. [21] Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. Learning structured prediction models: a large margin approach. 2004. [22] Ioannis Tsochantaridis, Google Inc, Thorsten Joachims, Thomas Hofmann, Yasemin Altun, and Yoram Singer. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453?1484, 2005. [23] Chen Yanover, Talya Meltzer, and Yair Weiss. Linear programming relaxations and belief propagation ? an empirical study. *Journal of Machine Learning Research*, 7:1887?1907, 2006. [24] Chen Yanover, Ora Schueler-furman, and Yair Weiss. Minimizing and learning energy functions for side-chain prediction. In RECOMB2007, 2007.