A Second-order message passing

Assume we have a factor tree with variable nodes $1, 2, \ldots, T$ and factor nodes F, where each factor $\alpha \in F$ is a set of variable indices. Let $F(t)$ be the set of factors in which variable t participates. Let y_t be an assignment to variable t, let y_α be an assignment to all of the variables in factor α, and let y be an assignment to all of the variables in the graph. The notation $y_\alpha \sim y_t$ means that y_α is consistent with y_t, in the sense that it assigns the same value to variable t. Suppose each factor has a real-valued weight function $w_\alpha(y_\alpha)$. The belief propagation algorithm defines the following message functions:

- From a variable t to a factor α:
$$m_{t\to\alpha}(y_t) = \prod_{\alpha' \in F(t) \setminus \{\alpha\}} m_{\alpha' \to t}(y_t) \tag{1}$$
- From a factor α to a variable t:
$$m_{\alpha\to t}(y_t) = \sum_{y_\alpha \sim y_t} \left[w_\alpha(y_\alpha) \prod_{t' \in \alpha \setminus \{t\}} m_{t' \to \alpha}(y_{t'}) \right] \tag{2}$$

In the forward pass, these messages are passed up the tree, from the leaves to the root. In the backward pass, the messages are passed back down, from the root to the leaves. Upon completion of the second pass, we have, for every y_t:

$$\prod_{\alpha \in F(t)} m_{\alpha\to t}(y_t) = \sum_{y \sim y_t} \prod_\alpha w_\alpha(y_\alpha) \tag{3}$$

If we think of the w_α as potential functions, then Equation (3) gives the (unnormalized) marginal probability of the assignment y_t under a Markov random field.

A.1 Semirings

The belief propagation algorithm is easily generalized to arbitrary semirings. Let $\langle W, \oplus, \otimes, \mathbf{0}, \mathbf{1} \rangle$ be a semiring with elements W, addition operator $\oplus$, multiplication operator $\otimes$, additive identity $\mathbf{0}$, and multiplicative identity $\mathbf{1}$. Then for $w_\alpha(y_\alpha) \in W$, belief propagation defines the following message functions:

$$m_{t\to\alpha}(y_t) = \bigotimes_{\alpha' \in F(t) \setminus \alpha} m_{\alpha' \to t}(y_t)$$
$$m_{\alpha\to t}(y_t) = \bigoplus_{y_\alpha \sim y_t} \left[w_\alpha(y_\alpha) \otimes \bigotimes_{t' \in \alpha \setminus t} m_{t' \to \alpha}(y_{t'}) \right] \tag{4}$$

After passing messages forward and backward, we have the following analog of Equation (3):

$$\bigotimes_{\alpha \in F(t)} m_{\alpha\to t}(y_t) = \bigoplus_{y \sim y_t} \bigotimes_\alpha w_\alpha(y_\alpha) \tag{5}$$

Under the max-product semiring, Equation (5) gives the so-called max-marginal—the maximum (unnormalized) probability of any total assignment consistent with y_t.

A.2 Second-order semiring

Li and Eisner proposed the following second-order semiring over four-tuples $(q, \phi, \psi, c) \in \mathbb{R}^4$:

$$\mathbf{0} = (0, 0, 0, 0) \tag{6}$$
$$\mathbf{1} = (1, 0, 0, 0) \tag{7}$$
$$(q_1, \phi_1, \psi_1, c_1) \oplus (q_2, \phi_2, \psi_2, c_2) = (q_1 + q_2,\ \phi_1 + \phi_2,\ \psi_1 + \psi_2,\ c_1 + c_2) \tag{8}$$
$$\begin{aligned}(q_1, \phi_1, \psi_1, c_1) \otimes (q_2, \phi_2, \psi_2, c_2) = (&q_1 q_2,\ q_1\phi_2 + q_2\phi_1,\ q_1\psi_2 + q_2\psi_1, \\ &q_1 c_2 + q_2 c_1 + \phi_1\psi_2 + \phi_2\psi_1)\end{aligned} \tag{9}$$

In our setting, the weight function for a factor α is given by

$$w_\alpha(y_\alpha) = (q^2(y_\alpha),\ q^2(y_\alpha)\phi_r(y_\alpha),\ q^2(y_\alpha)\phi_l(y_\alpha),\ q^2(y_\alpha)\phi_r(y_\alpha)\phi_l(y_\alpha)) \ , \tag{10}$$

where q are the quality scores and ϕ are the similarity features. In this case, the fourth component of $w_\alpha(y_\alpha) \otimes w_{\alpha'}(y_{\alpha'})$ is

$$\begin{aligned} q^2(y_\alpha)\left[q^2(y_{\alpha'})\phi_r(y_{\alpha'})\phi_l(y_{\alpha'})\right] + q^2(y_{\alpha'})\left[q^2(y_\alpha)\phi_r(y_\alpha)\phi_l(y_\alpha)\right] & \\ + \left[q^2(y_\alpha)\phi_r(y_\alpha)\right]\left[q^2(y_{\alpha'})\phi_l(y_{\alpha'})\right] + \left[q^2(y_{\alpha'})\phi_r(y_{\alpha'})\right]\left[q^2(y_\alpha)\phi_l(y_\alpha)\right] & \\ = q^2(y_\alpha)q^2(y_{\alpha'})\left[\phi_r(y_\alpha) + \phi_r(y_{\alpha'})\right]\left[\phi_l(y_\alpha) + \phi_l(y_{\alpha'})\right] \ . & \end{aligned} \tag{11}$$

By induction, it is possible to show that the fourth component of $\bigotimes_\alpha w_\alpha(y_\alpha)$ is

$$\left(\prod_\alpha q^2(y_\alpha)\right)\left(\sum_\alpha \phi_r(y_\alpha)\right)\left(\sum_\alpha \phi_l(y_\alpha)\right) \ . \tag{12}$$

Thus, by Equation (5) and the definition of $\oplus$, belief propagation with the second-order semiring yields messages that satisfy

$$\left[\bigotimes_{\alpha\in F(t)} m_{\alpha\to t}(y_t)\right]_4 = \sum_{y\sim y_t}\left(\prod_\alpha q^2(y_\alpha)\right)\left(\sum_\alpha \phi_r(y_\alpha)\right)\left(\sum_\alpha \phi_l(y_\alpha)\right) \ . \tag{13}$$

We can simply run the algorithm D^2 times for all pairs (r, l) (or just vectorize the semiring) to obtain C. In fact, since we only need complete messages at a single variable node, we can use the root node and avoid making the backward pass.

B Structured sampling

It is possible to fix a variable t' to a specific assignment $y_{t'}$ by creating a new singleton factor containing only that variable, and setting its weight to $\mathbf{1}$ for $y_{t'}$ and $\mathbf{0}$ otherwise. Then it is easy to see that Equation (5) becomes

$$\bigotimes_{\alpha\in F(t)} m_{\alpha\to t}(y_t) = \bigoplus_{y\sim y_t, y_{t'}} \bigotimes_\alpha w_\alpha(y_\alpha) \ , \tag{14}$$

where the sum is now doubly constrained, since any assignment y that is not consistent with $y_{t'}$ will introduce a $\mathbf{0}$ term into the product. If $\bigotimes_\alpha w_\alpha(y_\alpha)$ gives rise to a probability measure over labelings y, and $\oplus$ adds those probabilities, then Equation (14) yields the unnormalized *conditional* marginal probability of the assignment y_t given $y_{t'}$. In practice, we do not need to actually create a new factor; we can simply set outgoing messages from variable t' to $\mathbf{0}$ for all but the desired assignment $y_{t'}$.

For SDPP sampling, we have $p(y) = q^2(y)(v^\top\phi(y))^2$, where for simplicity we have assumed that $V = \{v\}$ contains only a single basis vector. (In general we can simply run the algorithm $|V|$ times, or vectorize.) We use the second-order semiring with

$$w_\alpha(y_\alpha) = (q^2(y_\alpha),\ q^2(y_\alpha)(v^\top\phi(y_\alpha)),\ q^2(y_\alpha)(v^\top\phi(y_\alpha)),\ q^2(y_\alpha)(v^\top\phi(y_\alpha))^2) \ . \tag{15}$$

Then the fourth component of Equation (14) is proportional to $p(y_t|y_{t'})$ by the same reasoning as Equation (13).

This observation gives rise to a naive algorithm for sampling a structure according to $p(y)$:

- Initialize a set of assignments $S = \emptyset$
- For $t = 1, \dots, T$
  - Run belief propagation with the assignments in S held fixed.
  - Sample y_t from the conditional marginal distribution $p(y_t|S)$.
  - Add y_t to S.
- Return the collected assignments S.

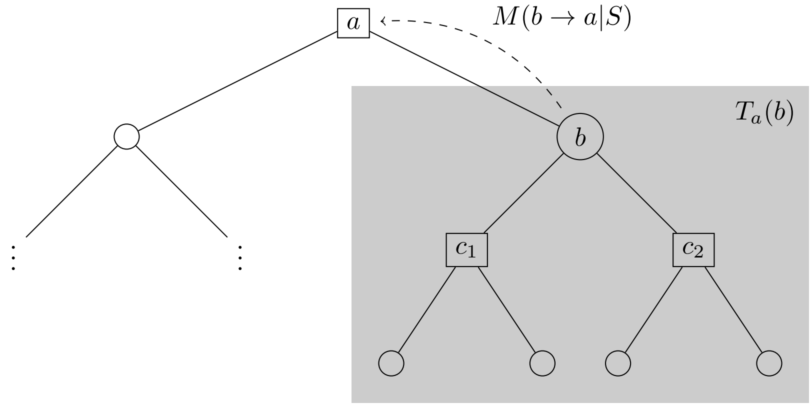

Figure 1: Illustration of $M(b \to a|S)$ and $T_a(b)$ when a is a (square) factor node and b is a (round) variable node. The same definitions apply when a is a variable and b is a factor.

B.1 Linear graphs

Unfortunately, this algorithm requires running belief propagation T times. Suppose, however, that the factor graph is a linear chain arranged from left to right. Then each node in the graph has at most two neighbors—one to the left, and one to the right. Assume the forward pass proceeds from left to right, and the backward pass from right to left. To send a message to the right, a node needs only to receive its message from the left. Conversely, to send a message to the left, only the message from the right is needed. Thus, the forward and backward passes can be performed independently.

Assume the variable nodes are numbered in decreasing order from left to right, so the variable sampled in the first iteration is the rightmost variable node. On iteration t, we do not actually need to run belief propagation to completion; we need only compute the forward messages sent from the left of variable t, and the backward messages from the right. These suffice to perform the sampling of variable t. Call this set of messages $R(t)$. Clearly $R(1)$ is just a full, unconstrained forward pass, which can be computed in time $O(T)$.

Now compare $R(t)$ to $R(t-1)$. Between iteration $t-1$ and t, the only change to S is that variable $t-1$, to the right of variable t, has been assigned. Therefore the forward messages in $R(t)$, which come from the left, do not need to be recomputed, as they are a subset of the forward messages in $R(t-1)$. Likewise, the backward messages sent from the right of variable $t-1$ are unchanged, so they do not need to be recomputed. The only new messages in $R(t)$ are those backward messages traveling from $t-1$ to t. These can be computed, using the sampled assignment y_{t-1}, in constant time.

Since the initial iteration takes $O(T)$ time and each of the subsequent $T-1$ iterations takes $O(1)$ time, we can sample from $p(y)$ over a linear graph in linear time.

B.2 Trees

In fact, the algorithm for linear graphs can be generalized to arbitrary factor trees. Let $M(b \to a|S)$ be the message function sent from node b to node a during a run of belief propagation where the assignments in S have been held fixed. Imagine that we re-root the factor tree with a as the root; then define $T_a(b)$ to be the subtree rooted at b (see Figure 1).

Several useful observations follow.

Lemma 1. *If b_1 and b_2 are distinct neighbors of a, then $T_a(b_1)$ and $T_a(b_2)$ are disjoint.*

Proof. The claim is immediate, since the underlying graph is a tree. □

Lemma 2. *$M(b \to a|S)$ can be computed given only the messages $M(c \to b|S)$ for all neighbors $c \neq a$ of b, the weight function w_b (if b is a factor node), and the assignment to b in S, if one exists.*

Proof. Follows from the message definitions in Equation (4). □

Lemma 3. *$M(b \to a|S)$ depends only on the assignments in S that give values to variables in $T_a(b)$.*

Proof. If b is a leaf (that is, its only neighbor is a), the lemma holds trivially. If b is not a leaf, then assume inductively that incoming messages $M(c \to b|S)$, $c \neq a$, depend only on assignments to variables in $T_b(c)$. By Lemma 2, the message $M(b \to a|S)$ depends only on those messages and (possibly) the assignment to b in S. Since b and $T_b(c)$ are subgraphs of $T_a(b)$, the claim follows. □

Suppose that we set $S_0 = \emptyset$ and initialize current messages $\hat{M}(b \to a) = M(b \to a|S_0)$ for all neighbor pairs (a, b). This can be done in time $O(T)$ via belief propagation.

Now we walk the graph, sampling assignments and updating the current messages $\hat{M}(b \to a)$ as we go. Step i from node b to a proceeds in three parts as follows:

1. Check whether b is a variable node without an assignment in S_{i-1}. If so, sample an assignment y_b using the current incoming messages $\hat{M}(c \to b)$, and set $S_i = S_{i-1} \cup \{y_b\}$. Otherwise set $S_i = S_{i-1}$.
2. Recompute and update $\hat{M}(b \to a)$ using the current messages and Equation (4), taking into account any assignment to b in S_i.
3. Advance to node a.

This simple algorithm has the following useful invariant.

Theorem 1. *Following step i in the walk, if our current location is a, then for every neighbor b of a we have $\hat{M}(b \to a) = M(b \to a|S_i)$.*

Proof. By design, the theorem holds at the outset of the walk. Suppose inductively that the claim is true for steps $1, \ldots, i-1$. Let i' be the most recent step prior to i at which we visited a, or 0 if step i was our first visit to a. Since the graph is a tree, we know that between steps i' and i the walk remained entirely within $T_a(b)$. Hence the only assignments in $S_i - S_{i'}$ are to variables in $T_a(b)$. Thus for all neighbors $d \neq b$ of a, we have $\hat{M}(d \to a) = M(d \to a|S_{i'}) = M(d \to a|S_i)$ by inductive assumption, Lemma 1, and Lemma 3.

It remains to show that $\hat{M}(b \to a) = M(b \to a|S_i)$. For all neighbors $c \neq a$ of b, we know that $\hat{M}(c \to b) = M(c \to b|S_{i-1}) = M(c \to b|S_i)$ due to the inductive hypothesis and Lemma 3 (since b is not in $T_b(c)$). By Lemma 2, then, we have $\hat{M}(b \to a) = M(b \to a|S_i)$. □

Theorem 1 guarantees that whenever we sample an assignment for the current variable node in the first part of step i, we sample from the conditional marginal distribution $p(y_b|S_{i-1})$. Therefore, we can sample a complete structure from the distribution $p(y)$ if we walk the entire tree. This can be done, for example, by starting at the root and proceeding in depth-first order. Such a walk takes $O(T)$ steps, and each step requires computing only a single message. Thus the algorithm runs in time $O(T)$.