16 Tree Re-weighted Variational Inference & the Bethe Approximation

In the previous lecture we developed the mean-field variational method, motivated by approximation of the partition function. Expressing the partition function as an optimization problem led naturally to a method for approximate inference. In this lecture we will revisit loopy belief propagation, and see that it is closely related to a different way of approximating the variational problem known as the Bethe approximation.

We first recall the loopy BP (parallel sum-product) algorithm. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph over random variables $\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_N \in \mathcal{X}$ distributed as

$$p_{\mathbf{x}}(\mathbf{x}) \propto \prod_{i \in \mathcal{V}} \exp\left(\tilde{\phi}_i(x_i)\right) \prod_{(i,j) \in \mathcal{E}} \exp\left(\tilde{\psi}_{ij}(x_i, x_j)\right). \tag{1}$$

The messages are initialized for $(i, j) \in \mathcal{E}$ and $x_i, x_j \in \mathcal{X}$ according to

$$m^0_{i \to j}(x_j) \propto 1 \propto m^0_{j \to i}(x_i).$$

Then, at each $t = 0, 1, 2, \ldots$ at each node $i \in \mathcal{V}$ a message is sent to neighbors $j \in \mathcal{N}(i)$,

$$m^{t+1}_{i \to j}(x_j) \propto \sum_{x_i \in \mathcal{X}} \exp\left(\tilde{\phi}_i(x_i)\right) \exp\left(\tilde{\psi}_{ij}(x_i, x_j)\right) \prod_{k \in \mathcal{N}(i) \setminus j} m^t_{k \to i}(x_i), \tag{2}$$

where we may normalize our messages: $\sum_{x_j \in \mathcal{X}} m^t_{i \to j}(x_j) = 1$.

The belief at node i is computed as

$$b^t_i(x_i) \propto \exp\left(\tilde{\phi}_i(x_i)\right) \prod_{k \in \mathcal{N}(i)} m^t_{k \to i}(x_i), \tag{3}$$

We may also form a belief for *edge marginals* from the messages:

$$b^t_{ij}(x_i, x_j) \propto \exp\left(\tilde{\phi}_i(x_i) + \tilde{\phi}_j(x_j) + \tilde{\psi}_{ij}(x_i, x_j)\right) \prod_{k \in \mathcal{N}(i) \setminus j} m^t_{k \to i}(x_i) \prod_{\ell \in \mathcal{N}(j) \setminus i} m^t_{\ell \to j}(x_j). \tag{4}$$

It is a good exercise to check that in a tree, $b^t_{ij}(x_i, x_j) = p_{\mathsf{x}_i, \mathsf{x}_j}(x_i, x_j)$ as long as t is larger than the diameter of the tree.

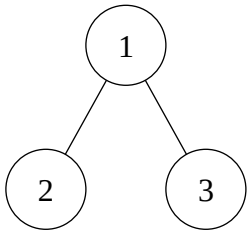


Figure 1

16.1 Variational Form of the Log-Partition Function

The guiding principle behind the variational approach to inference is to write the partition function as an optimization function. We rewrite (1) in the form of a Boltzmann distribution as

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_{i \in \mathcal{V}} \tilde{\phi}_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \tilde{\psi}_{ij}(x_i, x_j) \right) = \frac{1}{Z} e^{-E(\mathbf{x})}, \tag{5}$$

where Z is the partition function and $E(\mathbf{x}) \triangleq -\sum_{i \in \mathcal{V}} \tilde{\phi}_i(x_i) - \sum_{(i,j) \in \mathcal{E}} \tilde{\psi}_{ij}(x_i, x_j)$ refers to energy.

The variational characterization of the log partition function is then

$$\log Z = -D(q \parallel p_{\mathbf{x}}) + \varphi(q) = \sup_{q \in \mathcal{P}} \varphi(q) = \sup_{q \in \mathcal{P}} \left\{ -\mathbb{E}_q \left[E(\mathbf{x}) \right] + H(q) \right\}, \tag{6}$$

where $\mathcal{P}$ is the set of all distributions over $\mathcal{X}^N$ and $H(q) = \mathbb{E}_q \left[-\log q(\mathbf{x}) \right]$. As discussed before, the first equality indicates that the optimal choice of q in the optimization is $p_{\mathbf{x}}$, since in this case the divergence $D(q \parallel p_{\mathbf{x}}) = 0$.

This optimization problem is intractable in general because each of the two expectations in the objective entails a summation over exponentially many terms. We saw that the optimization massively simplified if we took q to have a product form, which resulted in the mean-field approximation. We next consider a more sophisticated way to relax this problem, by letting q factorize as a tree.

16.2 Variational Inference: Tree re-weighted

The guiding intuition we use for relaxing the log partition optimization is to look at what happens when $p_{\mathbf{x}}$ has a tree graph. By what we argued earlier, the solution to the log partition optimization is to set $q \equiv p_{\mathbf{x}}$, so if $p_{\mathbf{x}}$ has a tree graph, then q must factorize as a tree as well. Hence, we need not optimize over all possible distributions in $\mathcal{X}^N$ for q; we only need to look at distributions with tree graphs!

With this motivation, we look at how to parameterize q to simplify our optimization problem, which involves imposing constraints on q. We'll motivate the contraints

we're about to place by way of example. Supposing that $p_{\mathbf{x}}$ has the tree graph in Figure 1, then $p_{\mathbf{x}}$ factorizes as:

$$\begin{aligned} p_{\mathbf{x}}(\mathbf{x}) &= p_{x_1}(x_1)p_{x_2|x_1}(x_2|x_1)p_{x_3|x_1}(x_3|x_1) \\ &= p_{x_1}(x_1)\frac{p_{x_1,x_2}(x_1,x_2)}{p_{x_1}(x_1)}\frac{p_{x_1,x_3}(x_1,x_3)}{p_{x_1}(x_1)} \\ &= p_{x_1}(x_1)p_{x_2}(x_2)p_{x_3}(x_3)\frac{p_{x_1,x_2}(x_1,x_2)}{p_{x_1}(x_1)p_{x_2}(x_2)}\frac{p_{x_1,x_3}(x_1,x_3)}{p_{x_1}(x_1)p_{x_3}(x_3)}. \end{aligned}$$

A pattern emerges in the last step, which in fact holds in general: if $p_{\mathbf{x}}$ has tree graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, then its factorization is given by

$$p_{\mathbf{x}}(\mathbf{x}) = \prod_{i\in\mathcal{V}} p_{x_i}(x_i) \prod_{(i,j)\in\mathcal{E}} \frac{p_{x_i,x_j}(x_i,x_j)}{p_{x_i}(x_i)p_{x_j}(x_j)}.$$

Motivated by this representation of tree distributions, we consider the following parameterization of q. Let $p_{\mathbf{x}}$ have edge set $\mathcal{E}$ (not necessarily a tree graph). We pick some spanning tree $\mathcal{E}_T \subset \mathcal{E}$, and assume that q factors as

$$q(\mathbf{x}) = \prod_{i\in\mathcal{V}} q_i(x_i) \prod_{(i,j)\in\mathcal{E}_T} \frac{q_{ij}(x_i,x_j)}{q_i(x_i)q_j(x_j)}, \tag{7}$$

By forcing q to have the above factorization, we arrive at the following approximation for the log partition function:

$$\log Z_{\text{Tree}}(T) = \max_q \varphi(q) \tag{8}$$

subject to the constraints

$$\begin{aligned} q(\mathbf{x}) &= \prod_{i\in\mathcal{V}} q_i(x_i) \prod_{(i,j)\in\mathcal{E}_T} \frac{q_{ij}(x_i,x_j)}{q_i(x_i)q_j(x_j)} && \text{for all } \mathbf{x}\in\mathcal{X}^N \\ q_i(x_i) &\geq 0 && \text{for all } i\in\mathcal{V}, x_i\in\mathcal{X} \\ \sum_{x_i\in\mathcal{X}} q_i(x_i) &= 1 && \text{for all } i\in\mathcal{V} \\ q_{ij}(x_i,x_j) &\geq 0 && \text{for all } (i,j)\in\mathcal{E}_T, x_i, x_j\in\mathcal{X} \\ \sum_{x_j\in\mathcal{X}} q_{ij}(x_i,x_j) &= q_i(x_i) && \text{for all } (i,j)\in\mathcal{E}_T, x_i\in\mathcal{X} \\ \sum_{x_i\in\mathcal{X}} q_{ij}(x_i,x_j) &= q_j(x_j) && \text{for all } (i,j)\in\mathcal{E}_T, x_j\in\mathcal{X}\,. \end{aligned}$$

We let the set of distributions q which satisfy these conditions be $\mathcal{Q}_T$.

The key takeaway here is that *if* $p_{\mathbf{x}}$ *has a tree graph*, then the optimal q will factor as a tree and so this Bethe variational problem is equivalent to our original log partition optimization problem and, furthermore, $\log Z = \log Z_{\text{Tree}}(T)$.

Let's simplify $\varphi(q)$ under this factorization assumption. We have that

$$\varphi(q) = -\mathbb{E}_q[E(\mathbf{x})] + H(q)$$

First,

$$-\mathbb{E}_q[E(\mathbf{x})] = \sum_i \mathbb{E}_{q_i}[\tilde{\phi}_i(x_i)] + \sum_{(i,j)\in\mathcal{E}} \mathbb{E}_{q_{ij}}[\tilde{\psi}_{ij}(x_i, x_j)]$$

Also,

$$\begin{aligned}
H(q) &= -\sum_x q(x) \log q(x) \\
&= -\sum_x q(x) \left(\sum_i \log q_i(x) + \sum_{(i,j)\in\mathcal{E}_T} \log \frac{q_{ij}(x_i, x_j)}{q_i(x_i) q_j(x_j)} \right) \\
&= \sum_i \left(-\sum_{x_i \in \mathcal{X}} q_i(x_i) \log q_i(x_i) \right) + \sum_{(i,j)\in\mathcal{E}_T} \left(-\sum_{(x,x')\in\mathcal{X}^2} q_{ij}(x, x') \log \frac{q_{ij}(x, x')}{q_i(x) q_j(x)} \right) \\
&= \sum_i H(q_i) - \sum_{(i,j)\in\mathcal{E}_T} I(q_{ij}),
\end{aligned}$$

where

$$I(q_{ij}) := \sum_{(x,x')\in\mathcal{X}^2} q_{ij}(x, x') \log \frac{q_{ij}(x, x')}{q_i(x) q_j(x)}$$

is known as the *mutual information*. Therefore the optimization problem becomes

$$\max \sum_i \Big(\mathbb{E}_{q_i}[\tilde{\phi}_i(x_i)] + H(q_i) \Big) + \sum_{(i,j)\in\mathcal{E}} \mathbb{E}_{q_{ij}}[\tilde{\psi}_{ij}(x_i, x_j)] - \sum_{(i,j)\in\mathcal{E}_T} I(q_{ij})$$

subject to $q \in \mathcal{Q}_T$

The optimal objective $\log Z_{\text{Tree}}(T)$ is a lower bound to the log-partition function $\log Z$. Recall that we chose T to be an arbitrary spanning tree of $\mathcal{E}$. Conducting this analysis for each spanning tree $T \in \mathcal{T}(\mathcal{G})$, the set of spanning trees of $\mathcal{G}$, we obtain that

$$\max_{T\in\mathcal{T}(\mathcal{G})} \log Z_{\text{Tree}}(T) \leq \log Z$$

Furthermore, if we let ρ be a distribution over $\mathcal{T}(\mathcal{G})$, then we obtain another lower bound:

$$\sum_{T \in \mathcal{T}(\mathcal{G})} \rho(T) \log Z_{\text{Tree}}(T) \leq \log Z.$$

We will see that this interpretation is useful when we try to upper bound the log-partition function.

16.3 The Bethe Approximation

In the above tree approximation, the set of edges over which q factorizes, $\mathcal{E}_T$, is not the same as the set of edges over which p factorizes, $\mathcal{E}$. Let us relax our approximation further, and instead assume that q factors as

$$q(\mathbf{x}) = \prod_{i \in \mathcal{V}} q_i(x_i) \prod_{(i,j) \in \mathcal{E}} \frac{q_{ij}(x_i, x_j)}{q_i(x_i) q_j(x_j)}. \tag{9}$$

This is known as the *Bethe Approximation*. While seemingly ad-hoc, there is a powerful connection between the Bethe Approximation and Loopy Belief Propagation, as we will soon observe.

Crucially, q need not be an actual probability distribution: If the graph were a tree, then q would be a valid distribution, but we can still consider q of this form when the graph has cycles.

By forcing q to have the above factorization, we arrive at the *Bethe variational problem* for the log partition function:

$$\log Z_{\text{Bethe}} = \max_q \sum_i \Big(\mathbb{E}_{q_i}[\tilde{\phi}_i(x_i)] + H(q_i) \Big) + \sum_{(i,j) \in \mathcal{E}} \Big(\mathbb{E}_{q_{ij}}[\tilde{\psi}_{ij}(x_i, x_j)] - I(q_{ij}) \Big) \tag{10}$$

subject to the constraints

$$\begin{aligned}
q(\mathbf{x}) &= \prod_{i \in \mathcal{V}} q_i(x_i) \prod_{(i,j) \in \mathcal{E}} \frac{q_{ij}(x_i, x_j)}{q_i(x_i) q_j(x_j)} && \text{for all } \mathbf{x} \in \mathcal{X}^N \\
q_i(x_i) &\geq 0 && \text{for all } i \in \mathcal{V}, x_i \in \mathcal{X} \\
\sum_{x_i \in \mathcal{X}} q_i(x_i) &= 1 && \text{for all } i \in \mathcal{V} \\
q_{ij}(x_i, x_j) &\geq 0 && \text{for all } (i,j) \in \mathcal{E}, x_i, x_j \in \mathcal{X} \\
\sum_{x_j \in \mathcal{X}} q_{ij}(x_i, x_j) &= q_i(x_i) && \text{for all } (i,j) \in \mathcal{E}, x_i \in \mathcal{X} \\
\sum_{x_i \in \mathcal{X}} q_{ij}(x_i, x_j) &= q_j(x_j) && \text{for all } (i,j) \in \mathcal{E}, x_j \in \mathcal{X}\,.
\end{aligned}$$

Since q is not necessarily a probability distribution, we don't know whether Z_{Bethe} is a lower bound or upper bound for $\log Z$; it is, however, a valid approximation.

16.4 Loopy BP and the Bethe Approximation

We're now ready to state a landmark result on the connection between Loopy Belief Propagation and the Bethe Approximation.

Theorem 1. *(Yedidia, Freeman, Weiss 2001) The fixed points of the belief propagation message updates result in node and edge marginals that are local extrema of the Bethe variational problem* (10)*.*

We set the stage for the proof by first massaging the objective function a bit to make it clear exactly what we're maximizing once we plug in the equality constraint for q factorizing as q_i's and q_{ij}'s. This entails a fair bit of algebra. We'll start by computing simplified expressions for $\log p_{\mathbf{x}}$ and $\log q$, which we'll then use to derive reasonably simple expressions for the average energy and entropy terms of objective function φ.

Let

$$\varphi_{\text{Bethe}}(q) := \sum_i \Big(\mathbb{E}_{q_i}[\tilde{\phi}_i(x_i)] + H(q_i)\Big) + \sum_{(i,j)\in\mathcal{E}} \Big(\mathbb{E}_{q_{ij}}[\tilde{\psi}_{ij}(x_i,x_j)] - I(q_{ij})\Big)$$

be the function we want to maximize. We call φ_{Bethe} the *negative Bethe free energy.* We observe that

$$\varphi_{\text{Bethe}}(q) = \sum_i \mathbb{E}_{q_i}[\tilde{\phi}_i(x_i) - \log q_i(x_i)] + \sum_{(i,j)\in\mathcal{E}} \mathbb{E}_{q_{ij}}[\tilde{\psi}_{ij}(x_i,x_j) - \log q_{ij}(x_i,x_j) + \log q_i(x_i) + \log q_i(x_j)]$$

We add $\mathbb{E}_{q_{ij}}[\tilde{\phi}_i(x_i) + \tilde{\phi}_j(x_j)]$ to each term in the second sum, adding the negation of this to the first sum. This is equivalent to subtracting $d_i \mathbb{E}_{q_i}[\tilde{\phi}_i(x_i)]$ from each term in the first sum, where d_i denotes the degree of node i. Therefore

$$\begin{aligned}
\varphi_{\text{Bethe}}(q) &= \sum_{i\in\mathcal{V}} (1-d_i)\, \mathbb{E}_{q_i}\Big[\tilde{\phi}_i(\mathsf{x}_i) - \log q_i(\mathsf{x}_i)\Big] \\
&\quad + \sum_{(i,j)\in\mathcal{E}} \mathbb{E}_{q_{ij}}\Big[\tilde{\psi}_{ij}(\mathsf{x}_i,\mathsf{x}_j) + \tilde{\phi}_i(\mathsf{x}_i) + \tilde{\phi}_j(\mathsf{x}_j) - \log q_{ij}(\mathsf{x}_i,\mathsf{x}_j)\Big] \\
&= \sum_{i\in\mathcal{V}} (1-d_i) \sum_{x_i\in\mathcal{X}} q_i(x_i)[\tilde{\phi}_i(x_i) - \log q_i(x_i)] \\
&\quad + \sum_{(i,j)\in\mathcal{E}} \sum_{x_i,x_j\in\mathcal{X}} q_{ij}(x_i,x_j)[\tilde{\psi}_{ij}(x_i,x_j) + \tilde{\phi}_i(x_i) + \tilde{\phi}_j(x_j) - \log q_{ij}(x_i,x_j)].
\end{aligned} \tag{11}$$

This way of expressing φ_{Bethe} is useful because it shows explicitly the dependence on the variables $q_i(x_i)$ and $q_{ij}(x_i,x_j)$.

We can now rewrite the Bethe variational problem (10) as

$$\max_q \varphi_{\text{Bethe}}(q)$$

subject to the constraints:

$$\begin{aligned}
q_i(x_i) &\geq 0 && \text{for all } i \in \mathcal{V}, x_i \in \mathcal{X} \\
\sum_{x_i \in \mathcal{X}} q_i(x_i) &= 1 && \text{for all } i \in \mathcal{V} \\
q_{ij}(x_i, x_j) &\geq 0 && \text{for all } (i,j) \in \mathcal{E}, x_i, x_j \in \mathcal{X} \\
\sum_{x_j \in \mathcal{X}} q_{ij}(x_i, x_j) &= q_i(x_i) && \text{for all } (i,j) \in \mathcal{E}, x_i \in \mathcal{X} \\
\sum_{x_i \in \mathcal{X}} q_{ij}(x_i, x_j) &= q_j(x_j) && \text{for all } (i,j) \in \mathcal{E}, x_j \in \mathcal{X}
\end{aligned}$$

As a result, the stage is now set for us to prove Theorem 1.

Proof. Our plan of attack is to introduce Lagrange multipliers to solve the now simplified Bethe variational problem. We'll look at where the gradient of the Lagrangian is zero, corresponding to local extrema, and we'll see that at these local extrema, Lagrange multipliers relate to sum-product messages that have reached a fixed-point and the q_i's and q_{ij}'s correspond exactly to sum-product's node and edge marginals.

So first, we introduce Lagrange multipliers. We won't introduce any for the non-negativity constraints because it'll turn out that these constraints aren't active; i.e., the other constraints will already yield local extrema points that always have non-negative q_i and q_{ij}. Let's assign Lagrange multipliers to the rest of the constraints:

Constraints	**Lagrange multipliers**
$\sum_{x_i \in \mathcal{X}} q_i(x_i) = 1$	λ_i
$\sum_{x_j \in \mathcal{X}} q_{ij}(x_i, x_j) = q_i(x_i)$	$\lambda_{j \to i}(x_i)$
$\sum_{x_i \in \mathcal{X}} q_{ij}(x_i, x_j) = q_j(x_j)$	$\lambda_{i \to j}(x_j)$

Collectively calling all the Lagrange multipliers λ, the Lagrangian is thus

$$\begin{aligned}
\tilde{\varphi}(q, \lambda) = \varphi_{\text{Bethe}}(q) &+ \sum_{i \in \mathcal{V}} \lambda_i \left(\sum_{x_i \in \mathcal{X}} q_i(x_i) - 1 \right) \\
&+ \sum_{(i,j) \in \mathcal{E}} \sum_{x_i \in \mathcal{X}} \lambda_{j \to i}(x_i) \left(\sum_{x_j \in \mathcal{X}} q_{ij}(x_i, x_j) - q_i(x_i) \right) \\
&+ \sum_{(i,j) \in \mathcal{E}} \sum_{x_j \in \mathcal{X}} \lambda_{i \to j}(x_j) \left(\sum_{x_i \in \mathcal{X}} q_{ij}(x_i, x_j) - q_j(x_j) \right).
\end{aligned}$$

The local extrema of φ_{Bethe} subject to the constraints we imposed on q are precisely the points where the gradient of $\tilde{\varphi}$ with respect to q is zero and all the equality constraints in the table above are met.

Next, we take derivatives. We'll work with the form of φ_{Bethe} given in equation (11), and we'll use the result that $\frac{d}{dx} x \log x = \log x + 1$. First, we differentiate with respect to $q_k(x_k)$:

$$\begin{aligned}\frac{\partial \tilde{\varphi}(q,\lambda)}{\partial q_k(x_k)} &= (1-d_k)(\tilde{\phi}_k(x_k) - \log q_k(x_k) - 1) + \lambda_k - \sum_{\ell \in \mathcal{N}(k)} \lambda_{\ell \to k}(x_k) \\ &= -(d_k - 1)\tilde{\phi}_k(x_k) + (d_k - 1)(\log q_k(x_k) + 1) + \lambda_k - \sum_{\ell \in \mathcal{N}(k)} \lambda_{\ell \to k}(x_k).\end{aligned}$$

For nodes k satisfying $d_k > 1$, setting this to 0 gives

$$\log q_k(x_k) = \tilde{\phi}_k(x_k) + \frac{1}{d_k - 1}\left(\sum_{\ell \in \mathcal{N}(k)} \lambda_{\ell \to k}(x_k) - \lambda_k\right) - 1,$$

so

$$q_k(x_k) \propto \exp\left\{\tilde{\phi}_k(x_k) + \frac{1}{d_k - 1}\sum_{\ell \in \mathcal{N}(k)} \lambda_{\ell \to k}(x_k)\right\}. \tag{12}$$

For nodes k with $d_k = 1$, we get

$$\lambda_{\ell \to k}(x_k) = \lambda_k \tag{13}$$

for all values of x_k, where ℓ is the (only) neighbor of k.

Next, we differentiate with respect to $q_{k\ell}(x_k, x_\ell)$:

$$\frac{\partial \tilde{\varphi}(q,\lambda)}{\partial q_{k\ell}(x_k, x_\ell)} = \tilde{\psi}_{k\ell}(x_k, x_\ell) + \tilde{\phi}_k(x_k) + \tilde{\phi}_\ell(x_\ell) - \log q_{k\ell}(x_k, x_\ell) - 1 + \lambda_{\ell \to k}(x_k) + \lambda_{k \to \ell}(x_\ell).$$

Setting this to 0 gives:

$$q_{k\ell}(x_k, x_\ell) \propto \exp\left\{\tilde{\psi}_{k\ell}(x_k, x_\ell) + \tilde{\phi}_k(x_k) + \tilde{\phi}_\ell(x_\ell) + \lambda_{\ell \to k}(x_k) + \lambda_{k \to \ell}(x_\ell)\right\}. \tag{14}$$

We now introduce variables $m_{i \to j}(x_j)$ for $i \in \mathcal{V}$ and $j \in \mathcal{N}(i)$, which are conveniently named as they will turn out to be the same as sum-product messages, but right now, we do not make such an assumption! The idea is that we'll write our $\lambda_{i \to j}$ multipliers in terms of $m_{i \to j}$'s. But how do we do this? Pattern matching between equation (14) and the edge marginal equation (4) suggests that we should set, for any $j \in \mathcal{V}$ satisfying $|\mathcal{N}(j)| = d_j > 1$, and any $i \in \mathcal{N}(j), x_j \in \mathcal{X}$,

$$\lambda_{i \to j}(x_j) = \sum_{k \in \mathcal{N}(j) \setminus i} \log m_{k \to j}(x_j). \tag{15}$$

This can be thought as a "change of variable." For example, if $j = 4$ and $\mathcal{N}(j) = \{1, 2, 3\}$, the assignment (15) corresponds to

$$\begin{bmatrix} \lambda_{1\to 4}(x_4) \\ \lambda_{2\to 4}(x_4) \\ \lambda_{3\to 4}(x_4) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} \log m_{1\to 4}(x_4) \\ \log m_{2\to 4}(x_4) \\ \log m_{3\to 4}(x_4) \end{bmatrix}.$$

Since the matrix $\mathbf{M}$ is an invertible matrix if $|\mathcal{N}(j)| > 1$, this change of variable is one-to-one. For $j \in \mathcal{V}$ satisfying $d_j = 1$, from (13) we already have know that $\lambda_{i\to j}(x_j)$ is constant over all values of x_j.

With this substitutions (15) and (13), equation (14) becomes

$$q_{k\ell}(x_k, x_\ell) \propto \exp\left(\tilde{\psi}_{k\ell}(x_k, x_\ell) + \tilde{\phi}_k(x_k) + \tilde{\phi}_\ell(x_\ell)\right) \prod_{i\in\mathcal{N}(k)\setminus\ell} m_{i\to k}(x_k) \prod_{j\in\mathcal{N}(\ell)\setminus k} m_{j\to\ell}(x_\ell), \tag{16}$$

which means that at a local extremum of φ_{Bethe}, the above equation is satisfied. Of course, this is the same as the edge marginal equation (4) for sum-product. We now verify a similar result for q_k, for $k \in \mathcal{V}$ with degree $d_k > 1$. The key observation is that

$$\sum_{\ell\in\mathcal{N}(k)} \lambda_{\ell\to k}(x_k) = \sum_{\ell\in\mathcal{N}(k)} \sum_{i\in\mathcal{N}(k)\setminus\ell} \log m_{i\to k}(x_k) = (d_k - 1) \sum_{j\in\mathcal{N}(k)} \log m_{j\to k}(x_k),$$

which follows from a counting argument. Plugging this result directly into equation (12), we obtain

$$\begin{aligned} q_k(x_k) &\propto \exp\left\{\tilde{\phi}_k(x_k) + \frac{1}{d_k - 1} \sum_{\ell\in\mathcal{N}(k)} \lambda_{\ell\to k}(x_k)\right\} \\ &= \exp\left\{\tilde{\phi}_k(x_k) + \frac{1}{d_k - 1}(d_k - 1) \sum_{j\in\mathcal{N}(k)} \log m_{j\to k}(x_k)\right\} \\ &= \exp(\tilde{\phi}_k(x_k)) \prod_{j\in\mathcal{N}(k)} m_{j\to k}(x_k), \end{aligned} \tag{17}$$

matching the sum-product node marginal equation (3).

What we've shown so far is that any local extremum q of φ_{Bethe} subject to the constraints we imposed must have q_i and q_{ij} take on the forms given in Equations (17) and (16), which just so happen to match node marginal equations (3) (for nodes with degree $d_k > 1$) and edge marginal equations (4) for sum-product. What we have left to show is the node marginal equations (3) for nodes with degree $d_k = 1$, and that the messages $m_{i\to j}$ themselves are at a fixed point of the message update equations (2).

To show this last step, we examine our equality constraints on q_{ij}. It suffices to just look at what happens for one of them:

$$\begin{aligned}
&\sum_{x_\ell \in \mathcal{X}} q_{k\ell}(x_k, x_\ell) \\
&\propto \sum_{x_\ell \in \mathcal{X}} \exp\left(\tilde{\psi}_{k\ell}(x_k, x_\ell) + \tilde{\phi}_k(x_k) + \tilde{\phi}_\ell(x_\ell)\right) \prod_{i \in \mathcal{N}(k)\setminus \ell} m_{i\to k}(x_k) \prod_{j \in \mathcal{N}(\ell)\setminus k} m_{j\to \ell}(x_\ell) \\
&= \left(\exp(\tilde{\phi}_k(x_k)) \prod_{i \in \mathcal{N}(k)\setminus \ell} m_{i\to k}(x_k)\right) \sum_{x_\ell \in \mathcal{X}} \exp\left(\tilde{\psi}_{k\ell}(x_k, x_\ell) + \tilde{\phi}_\ell(x_\ell)\right) \prod_{j \in \mathcal{N}(\ell)\setminus k} m_{j\to \ell}(x_\ell),
\end{aligned}$$

at which point, comparing the above equation and the q_k equation (17), perforce, we have

$$m_{\ell\to k}(x_k) = \sum_{x_\ell} \exp\left(\tilde{\psi}_{k\ell}(x_k, x_\ell) + \tilde{\phi}_\ell(x_\ell)\right) \prod_{j \in \mathcal{N}(\ell)\setminus k} m_{j\to \ell}(x_\ell).$$

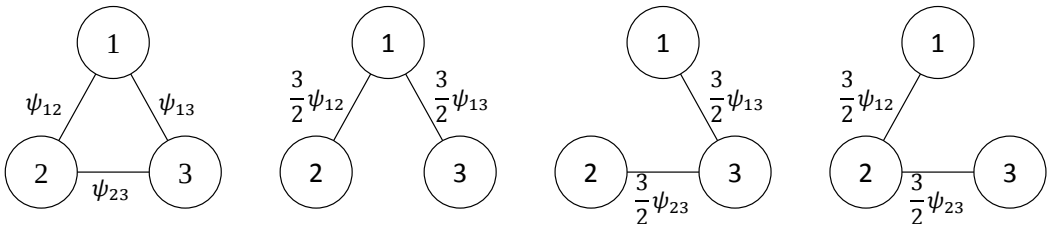
The above equation is just the sum-product message-passing update equation, given in (2)! Moreover, the above is in fact satisfied at a local extremum of φ_{Bethe} for all messages $m_{i\to j}$. Note that there is absolutely no dependence on iterations. This equation say that once we're at a local extremum of φ_{Bethe}, the above equation must hold for all $x_k \in \mathcal{X}$. This same argument holds for all the other $m_{i\to j}$'s, which means that we're at a fixed-point of the sum-product message-update equations.

Lastly, for any $k \in \mathcal{V}$ satisfying $d_k = 1$, consider again the sum above:

$$\begin{aligned}
q_k(x_k) &= \sum_{x_\ell \in \mathcal{X}} q_{k\ell}(x_k, x_\ell) \\
&\propto \sum_{x_\ell \in \mathcal{X}} \exp\left(\tilde{\psi}_{k\ell}(x_k, x_\ell) + \tilde{\phi}_k(x_k) + \tilde{\phi}_\ell(x_\ell)\right) \prod_{j \in \mathcal{N}(\ell)\setminus k} m_{j\to \ell}(x_\ell) \\
&= \exp(\tilde{\phi}_k(x_k)) \sum_{x_\ell \in \mathcal{X}} \exp\left(\tilde{\psi}_{k\ell}(x_k, x_\ell) + \tilde{\phi}_\ell(x_\ell)\right) \prod_{j \in \mathcal{N}(\ell)\setminus k} m_{j\to \ell}(x_\ell) \\
&= \exp(\tilde{\phi}_k(x_k)) m_{\ell\to k}(x_k),
\end{aligned}$$

which matches the sum-product node marginal equation (3). □

We've established that all possible fixed points of loopy BP are stationary points of φ_{Bethe}, the negative Bethe free energy. Unfortunately, Theorem 1 is only a statement about stationary points and says nothing about whether a fixed point of sum-product message updates is a local maximum or a local minimum (or neither) for φ_{Bethe}. Empirically, it has been found that fixed points corresponding to maxima of the negative Bethe free energy, i.e., minima of Bethe free energy, tend to be more stable. This empirical result intuitively makes sense since for loopy graphs, we can view local maxima of φ_{Bethe} as trying to approximate $\log Z$.



(a) Original graph (b) Tree $\mathcal{T}_1$ (c) Tree $\mathcal{T}_2$ (d) Tree $\mathcal{T}_3$

Figure 2: A loopy graph and all its spanning trees $\mathcal{T}_1$, $\mathcal{T}_2$, and $\mathcal{T}_3$. Log edge potentials are shown next to each edge.

16.5 Upper Bounds

Developing upper bounds for the log partition function is more involved. One practical approach is referred to as *tree-reweighted belief propagation.*[1] While some of the details are beyond the scope of our treatment, we'll sketch the main ideas via an example.

Suppose we want an upper bound on the log partition function for $p_{\mathbf{x}\mathbf{v}}$ defined over the graph in Fig. 2a, where we have a chosen, for convenience, a representation in which there are only edge potentials. For this collection of nodes, there are three possible spanning trees. In anticipation of their use, we assign log edge potentials to each of these trees as shown in Figs. 2b, 2c, and 2d.

In particular, we choose the log edge potentials so that their convex combinations (with uniform weights) coincide with the corresponding log edge potentials in the original graph. For example, note that

$$\begin{aligned}\tilde{\psi}_{12} &= \underbrace{\frac{1}{3}}_{\substack{\text{weight of}\\ \text{tree } \mathcal{T}_1}} \cdot \underbrace{\tilde{\psi}^1_{12}}_{\substack{\text{edge (1,2)'s}\\ \text{log potential}\\ \text{in tree } \mathcal{T}_1}} + \underbrace{\frac{1}{3}}_{\substack{\text{weight of}\\ \text{tree } \mathcal{T}_2}} \cdot \underbrace{\tilde{\psi}^2_{12}}_{\substack{\text{edge (1,2)'s}\\ \text{log potential}\\ \text{in tree } \mathcal{T}_2}} + \underbrace{\frac{1}{3}}_{\substack{\text{weight of}\\ \text{tree } \mathcal{T}_3}} \cdot \underbrace{\tilde{\psi}^3_{12}}_{\substack{\text{edge (1,2)'s}\\ \text{log potential}\\ \text{tree } \mathcal{T}_3}} \\ &= \frac{1}{3}\left(\frac{3}{2}\tilde{\psi}_{12}\right) + \frac{1}{3}\left(0 \cdot \tilde{\psi}_{12}\right) + \frac{1}{3}\left(\frac{3}{2}\tilde{\psi}_{12}\right).\end{aligned}$$

Hence, with

$$\tilde{\psi}_{ij}(x_i, x_j) = \frac{1}{3}\sum_{k=1}^{3} \tilde{\psi}^k_{ij}(x_i, x_j)$$

and using $\mathcal{E}_k$ to denote the edges in the spanning tree $\mathcal{T}_k$, we can upper bound the

[1]This approach is developed in M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "A New Class of Upper Bounds on the Log Partition Function," *IEEE Trans. Inform. Theory*, vol. 51, no. 7, pp. 2313–2335, July 2005.

log-partition function according to

$$\begin{aligned}
\ln Z &= \sup_{q\in\mathcal{P}} \left\{ \mathbb{E}_q \left[\sum_{(i,j)\in\mathcal{E}} \tilde{\psi}_{ij}(\mathsf{x}_i, \mathsf{x}_j) \right] + H(q) \right\} \\
&= \sup_{q\in\mathcal{P}} \left\{ \frac{1}{3} \sum_{k=1}^{3} \mathbb{E}_q \left[\sum_{(i,j)\in\mathcal{E}_k} \tilde{\psi}^k_{ij}(\mathsf{x}_i, \mathsf{x}_j) \right] + H(q) \right\} \\
&= \sup_{q\in\mathcal{P}} \left\{ \frac{1}{3} \sum_{k=1}^{3} \left\{ \mathbb{E}_q \left[\sum_{(i,j)\in\mathcal{E}_k} \tilde{\psi}^k_{ij}(\mathsf{x}_i, \mathsf{x}_j) \right] + H(q) \right\} \right\} \\
&\leq \frac{1}{3} \sum_{k=1}^{3} \sup_{q\in\mathcal{P}} \left\{ \mathbb{E}_q \left[\sum_{(i,j)\in\mathcal{E}_k} \tilde{\psi}^k_{ij}(\mathsf{x}_i, \mathsf{x}_j) \right] + H(q) \right\}, \qquad (18)
\end{aligned}$$

where the last step uses the key fact that for any real-valued functions f and g defined over the same domain

$$\sup_u \big\{ f(u) + g(u) \big\} \leq \Big(\sup_u f(u) \Big) + \Big(\sup_u g(u) \Big).$$

Finally, it suffices to note that the optimizations in (18) satisfy

$$\sup_{q\in\mathcal{P}} \left\{ \mathbb{E}_q \left[\sum_{(i,j)\in\mathcal{E}_k} \tilde{\psi}^k_{ij}(\mathsf{x}_i, \mathsf{x}_j) \right] + H(q) \right\} = \sup_{q\in\mathcal{P}(\mathcal{T}_k)} \left\{ \mathbb{E}_q \left[\sum_{(i,j)\in\mathcal{E}_k} \tilde{\psi}^k_{ij}(\mathsf{x}_i, \mathsf{x}_j) \right] + H(q) \right\}, \qquad (19)$$

since we know the optimizing distribution q^k for the tree $\mathcal{T}_k$ is the distribution defined by the potentials over $\mathcal{T}_k$ and thus lies in $\mathcal{P}(\mathcal{T}_k)$. But recall that all $q \in \mathcal{P}(\mathcal{T}_k)$ factor in the form (7) since they are trees, and thus (19) is a special case of the Bethe variational optimization where the underlying graph is a tree, and thus the optimizing node and edge potentials in the tree are obtained by the belief propagation algorithm!

Note that to obtain tighter upper bounds we can replace the uniform weighting over the spanning trees with nonuniform ones, and optimize over all possible such convex combinations.

As a closing comment, it should be emphasized that while this approach can be applied to graphs of arbitrary size, since there are N^{N-2} spanning trees for a graph $\mathcal{G}$ with N nodes, for this method to be practical it is necessary to avoid this complexity, which is addressed in Wainwright, Jaakkola and Willsky (2005).