# Probabilistic & Unsupervised Learning Approximate Inference

# **Belief Propagation**

#### Maneesh Sahani

maneesh@gatsby.ucl.ac.uk

Gatsby Computational Neuroscience Unit, and MSc ML/CSML, Dept Computer Science University College London

Term 1, Autumn 2022

## **Recall: Belief Propagation on undirected trees**

Joint distribution of undirected tree:

$$p(\mathcal{X}) = \frac{1}{Z} \prod_{\text{nodes } i} f_i(X_i) \prod_{\text{edges } (ij)} f_{ij}(X_i, X_j)$$



Messages computed recursively:

$$M_{j
ightarrow i}(X_i):=\sum_{X_j} \mathit{f}_{ij}(X_i,X_j)\mathit{f}_j(X_j) \prod_{l\in \mathsf{ne}(j)ackslash i} M_{l
ightarrow j}(X_j)$$

Marginal distributions:

$$p(X_i) \propto f_i(X_i) \prod_{k \in ne(i)} M_{k \to i}(X_i)$$
  
 $p(X_i, X_j) \propto f_{ij}(X_i, X_j) f_i(X_i) f_j(X_j) \prod_{k \in ne(i) \setminus j} M_{k \to i}(X_i) \prod_{l \in ne(j) \setminus i} M_{l \to j}(X_j)$ 

## **Loopy Belief Propagation**

Joint distribution of undirected graph:

$$p(\mathcal{X}) = \frac{1}{Z} \prod_{\text{nodes } i} f_i(X_i) \prod_{\text{edges } (ij)} f_{ij}(X_i, X_j)$$



Messages computed recursively (with few guarantees of convergence):

$$M_{j \rightarrow i}(X_i) := \sum_{X_j} f_{ij}(X_i, X_j) f_j(X_j) \prod_{l \in \mathsf{ne}(j) \setminus i} M_{l \rightarrow j}(X_j)$$

Marginal distributions are approximate in general:

$$p(X_i) \approx b_i(X_i) \propto f_i(X_i) \prod_{k \in ne(i)} M_{k \to i}(X_i)$$
  
 $p(X_i, X_j) \approx b_{ij}(X_i, X_j) \propto f_{ij}(X_i, X_j) f_i(X_i) f_j(X_j) \prod_{k \in ne(i) \setminus j} M_{k \to i}(X_i) \prod_{l \in ne(j) \setminus i} M_{l \to j}(X_j)$ 

Accuracy: BP posterior marginals are approximate on all non-trees because evidence is over counted, but converged approximations are frequently found to be good (particularly in their means).

Accuracy: BP posterior marginals are approximate on all non-trees because evidence is over counted, but converged approximations are frequently found to be good (particularly in their means).

Convergence: no general guarantee, but BP does converge in some cases:

- Trees.
- Graphs with a single loop.
- Distributions with sufficiently weak interactions.
- Graphs with long (and weak) loops
- Gaussian networks: means correct, variances may also converge.

Accuracy: BP posterior marginals are approximate on all non-trees because evidence is over counted, but converged approximations are frequently found to be good (particularly in their means).

Convergence: no general guarantee, but BP does converge in some cases:

- Trees.
- Graphs with a single loop.
- Distributions with sufficiently weak interactions.
- Graphs with long (and weak) loops
- Gaussian networks: means correct, variances may also converge.
- Damping: Common approach to encourage convergence (cf EP)

$$M_{i \to j}^{\mathsf{new}}(X_j) := (1 - \alpha) M_{i \to j}^{\mathsf{old}}(X_j) + \alpha \sum_{X_i} f_{ij}(X_i, X_j) f_i(X_i) \prod_{k \in \mathsf{ne}(i) \setminus j} M_{k \to i}(X_i)$$

Accuracy: BP posterior marginals are approximate on all non-trees because evidence is over counted, but converged approximations are frequently found to be good (particularly in their means).

Convergence: no general guarantee, but BP does converge in some cases:

- Trees.
- Graphs with a single loop.
- Distributions with sufficiently weak interactions.
- Graphs with long (and weak) loops
- Gaussian networks: means correct, variances may also converge.

Damping: Common approach to encourage convergence (cf EP)

$$M_{i 
ightarrow j}^{\mathsf{new}}(X_j) := (1 - lpha) M_{i 
ightarrow j}^{\mathsf{old}}(X_j) + lpha \sum_{X_i} f_{ij}(X_i, X_j) f_i(X_i) \prod_{k \in \mathsf{ne}(i) \setminus j} M_{k 
ightarrow i}(X_i)$$

- Grouping variables: Variables can be grouped into cliques to improve accuracy.
  - Region graph approximations.
  - Cluster variational method.
  - Junction graph.

## **Different Interpretations of Loopy Belief Propagation**

Loopy BP can be interpreted as a fixed point algorithm from a few different perspectives:

- Expectation propagation.
- Tree-based reparametrization.
- Bethe free energy.

## **Different Interpretations of Loopy Belief Propagation**

Loopy BP can be interpreted as a fixed point algorithm from a few different perspectives:

- Expectation propagation.
- Tree-based reparametrization.
- Bethe free energy.

## Loopy BP as message-based Expectation Propagation



Approximate pairwise factors  $f_{ij}$  by product of messages:

 $f_{ij}(X_i, X_j) \approx \tilde{f}_{ij}(X_i, X_j) = M_{i \to j}(X_j) M_{j \to i}(X_i)$ 

Thus, the full joint is approximated by a factorised distribution:

$$p(\mathcal{X}) \approx \frac{1}{Z} \prod_{\text{nodes } i} f_i(X_i) \prod_{\text{edges } (ij)} \tilde{f}_{ij}(X_i, X_j) = \frac{1}{Z} \prod_{\text{nodes } i} \left( f_i(X_i) \prod_{j \in \text{ne}(i)} M_{j \to i}(X_i) \right) = \prod_{\text{nodes } i} b_i(X_i)$$

but with multiple factors for most  $X_i$ .



Then the EP updates to the messages are:



Then the EP updates to the messages are:

► Deletion:

$$q_{\neg ij}(\mathcal{X}) = f_i(X_i)f_j(X_j) \prod_{k \in \mathsf{ne}(i) \setminus j} M_{k \to i}(X_i) \prod_{l \in \mathsf{ne}(j) \setminus i} M_{l \to j}(X_j) \prod_{s \neq i, j} f_s(X_s) \prod_{t \in \mathsf{ne}(s)} M_{t \to s}(X_s)$$



Then the EP updates to the messages are:

► Deletion:

$$q_{\neg ij}(X_i, X_j) = f_i(X_i)f_j(X_j) \prod_{k \in \mathsf{ne}(i) \setminus j} M_{k \to i}(X_i) \prod_{l \in \mathsf{ne}(j) \setminus i} M_{l \to j}(X_j) \prod_{s \neq i, j} f_s(X_s) \prod_{t \in \mathsf{ne}(s)} M_{t \to s}(X_s)$$



Then the EP updates to the messages are:

► Deletion:

$$q_{\neg ij}(X_i, X_j) = f_i(X_i)f_j(X_j) \prod_{k \in ne(i) \setminus j} M_{k \to i}(X_i) \prod_{l \in ne(j) \setminus i} M_{l \to j}(X_j) \prod_{s \neq i, j} f_s(X_s) \prod_{t \in ne(s)} M_{t \to s}(X_s)$$

► Projection:

 $\{M_{i \to j}^{\text{new}}, M_{j \to i}^{\text{new}}\} = \operatorname{argmin} \mathsf{KL}[f_{ij}(X_i, X_j)q_{\neg ij}(X_i, X_j) \| M_{j \to i}(X_i) M_{i \to j}(X_j)q_{\neg ij}(X_i, X_j)]$ 



Then the EP updates to the messages are:

► Deletion:

$$q_{\neg ij}(X_i, X_j) = f_i(X_i)f_j(X_j) \prod_{k \in ne(i) \setminus j} M_{k \to i}(X_i) \prod_{l \in ne(j) \setminus i} M_{l \to j}(X_j) \prod_{s \neq i, j} f_s(X_s) \prod_{t \in ne(s)} M_{t \to s}(X_s)$$

Projection:

 $\{M_{i \to j}^{\text{new}}, M_{j \to i}^{\text{new}}\} = \operatorname{argmin} \mathsf{KL}[f_{ij}(X_i, X_j)q_{\neg ij}(X_i, X_j) \| M_{j \to i}(X_i)M_{i \to j}(X_j)q_{\neg ij}(X_i, X_j)]$ 

Now,  $q_{\neg ij}()$  factors  $\Rightarrow$  rhs factors  $\Rightarrow$  min is achieved by marginals of  $f_{ij}()q_{\neg ij}()$ 



Then the EP updates to the messages are:

Deletion:

$$q_{\neg ij}(X_i, X_j) = f_i(X_i)f_j(X_j) \prod_{k \in \mathsf{ne}(i) \setminus j} M_{k \to i}(X_i) \prod_{l \in \mathsf{ne}(j) \setminus i} M_{l \to j}(X_j) \prod_{s \neq i, j} f_s(X_s) \prod_{t \in \mathsf{ne}(s)} M_{t \to s}(X_s)$$

Projection:

 $\{M_{i \to j}^{\text{new}}, M_{j \to i}^{\text{new}}\} = \operatorname{argmin} \mathsf{KL}[f_{ij}(X_i, X_j)q_{-ij}(X_i, X_j) \| M_{j \to i}(X_i) M_{i \to j}(X_j)q_{-ij}(X_i, X_j)]$ 

Now,  $q_{\neg ij}()$  factors  $\Rightarrow$  rhs factors  $\Rightarrow$  min is achieved by marginals of  $f_{ij}()q_{\neg ij}()$ 

Thus message-based EP in a loopy graph need not be seen as two separate approximations (one to the sites and one to the cavity) as we had in the EP lecture.

- Thus message-based EP in a loopy graph need not be seen as two separate approximations (one to the sites and one to the cavity) as we had in the EP lecture.
- Instead, we can see it as a more severe constraint on the approximate sites: not just to an ExpFam factor, but to a product of ExpFam messages.

- Thus message-based EP in a loopy graph need not be seen as two separate approximations (one to the sites and one to the cavity) as we had in the EP lecture.
- Instead, we can see it as a more severe constraint on the approximate sites: not just to an ExpFam factor, but to a product of ExpFam messages.
- On a tree-structured graph the message-factored version of EP finds the same marginals as standard EP.

- Thus message-based EP in a loopy graph need not be seen as two separate approximations (one to the sites and one to the cavity) as we had in the EP lecture.
- Instead, we can see it as a more severe constraint on the approximate sites: not just to an ExpFam factor, but to a product of ExpFam messages.
- On a tree-structured graph the message-factored version of EP finds the same marginals as standard EP.
  - Messages are calculated in exactly the same way as before (cf NLSSM).

- Thus message-based EP in a loopy graph need not be seen as two separate approximations (one to the sites and one to the cavity) as we had in the EP lecture.
- Instead, we can see it as a more severe constraint on the approximate sites: not just to an ExpFam factor, but to a product of ExpFam messages.
- On a tree-structured graph the message-factored version of EP finds the same marginals as standard EP.
  - Messages are calculated in exactly the same way as before (cf NLSSM).
  - Pairwise marginals can be found after convergence by computing  $\tilde{P}(\mathbf{z}_{i-1}, \mathbf{z}_i)$  as required (cf Forward-backward for HMMs).

- Thus message-based EP in a loopy graph need not be seen as two separate approximations (one to the sites and one to the cavity) as we had in the EP lecture.
- Instead, we can see it as a more severe constraint on the approximate sites: not just to an ExpFam factor, but to a product of ExpFam messages.
- On a tree-structured graph the message-factored version of EP finds the same marginals as standard EP.
  - Messages are calculated in exactly the same way as before (cf NLSSM).
  - Pairwise marginals can be found after convergence by computing  $\tilde{P}(\mathbf{z}_{i-1}, \mathbf{z}_i)$  as required (cf Forward-backward for HMMs).
  - Would not be true of fully-factored variational approximation.

- Thus message-based EP in a loopy graph need not be seen as two separate approximations (one to the sites and one to the cavity) as we had in the EP lecture.
- Instead, we can see it as a more severe constraint on the approximate sites: not just to an ExpFam factor, but to a product of ExpFam messages.
- On a tree-structured graph the message-factored version of EP finds the same marginals as standard EP.
  - Messages are calculated in exactly the same way as before (cf NLSSM).
  - Pairwise marginals can be found after convergence by computing  $\tilde{P}(\mathbf{z}_{i-1}, \mathbf{z}_i)$  as required (cf Forward-backward for HMMs).
  - Would not be true of fully-factored variational approximation.
- Factorisation view remains valid even when original sites lie in the appropriate ExpFam already so loopy BP in (eg) discrete graphs can be seen as a form of EP.

- Thus message-based EP in a loopy graph need not be seen as two separate approximations (one to the sites and one to the cavity) as we had in the EP lecture.
- Instead, we can see it as a more severe constraint on the approximate sites: not just to an ExpFam factor, but to a product of ExpFam messages.
- On a tree-structured graph the message-factored version of EP finds the same marginals as standard EP.
  - Messages are calculated in exactly the same way as before (cf NLSSM).
  - Pairwise marginals can be found after convergence by computing  $\tilde{P}(\mathbf{z}_{i-1}, \mathbf{z}_i)$  as required (cf Forward-backward for HMMs).
  - Would not be true of fully-factored variational approximation.
- Factorisation view remains valid even when original sites lie in the appropriate ExpFam already so loopy BP in (eg) discrete graphs can be seen as a form of EP.
- However, this view does not help us understand the convergence properties of BP.

## **Different Interpretations of Loopy Belief Propagation**

Loopy BP can be interpreted as a fixed point algorithm from a few different perspectives:

- Expectation propagation.
- ► Tree-based reparametrization.
- Bethe free energy.

## Loopy BP as tree-based reparametrisation

Tree-structured distributions can be parametrised in many ways:

$$p(\mathcal{X}) = \frac{1}{Z} \prod_{\text{nodes } i} f_i(X_i) \prod_{\text{edges}(ij)} f_{ij}(X_i, X_j) \qquad \text{undirected tree}$$
(1)  
$$= p(X_r) \prod_{i \neq r} p(X_i | X_{\text{pa}(i)}) \qquad \text{directed (rooted) tree}$$
(2)  
$$= \prod_{\text{nodes } i} p(X_i) \prod_{\text{edges}(ij)} \frac{p(X_i, X_j)}{p(X_i)p(X_j)} \qquad \text{pairwise marginals}$$
(3)

where (3) requires that  $\sum_{X_i} p(X_i, X_j) = p(X_i)$ .

The undirected tree representation is not unique—multiplying a factor  $f_{ij}(X_i, X_j)$  by  $g(X_i)$  and dividing  $f_i(X_i)$  by the same  $g(X_i)$  does not change the distribution.

BP can be seen as an iterative replacement of  $f_i(X_i)$  by the local marginal of  $p_{ij}(X_i, X_j)$ , along with the corresponding reparametrisation of  $f_{ij}(X_i, X_j)$ . Cf. Hugin propagation.

Converged BP on a tree finds  $p(X_i)$  and  $p(X_i, X_j)$ , allowing us to transform (1) to (3).



Define  $f_{ij}^0 = f_{ij}$  (absorbing singleton factors), and  $f_i^0 = p_i^0 = 1$ . Iterate over edges (*ij*):





Define  $f_{ij}^0 = f_{ij}$  (absorbing singleton factors), and  $f_i^0 = p_i^0 = 1$ . Iterate over edges (*ij*):

 $p^n(X_i, X_j) = \frac{1}{Z_{ij}^n} f_i^{n-1}(X_i) f_{ij}^{n-1}(X_i, X_j) f_j^{n-1}(X_j) \quad \text{[store } Z_{ij}^n \text{s to obtain joint normaliser]}$ 





$$p^{n}(X_{i}, X_{j}) = \frac{1}{Z_{ij}^{n}} f_{i}^{n-1}(X_{i}) f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})$$

$$f_{i}^{n}(X_{i}) = p^{n}(X_{i}) = \sum_{X_{j}} p^{n}(X_{i}, X_{j}) = f_{i}^{n-1}(X_{i}) \underbrace{\sum_{X_{j}} f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})}_{M_{j \to i}}$$





$$p^{n}(X_{i}, X_{j}) = \frac{1}{Z_{ij}^{n}} f_{i}^{n-1}(X_{i}) f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})$$

$$f_{i}^{n}(X_{i}) = p^{n}(X_{i}) = \sum_{X_{j}} p^{n}(X_{i}, X_{j}) = f_{i}^{n-1}(X_{i}) \underbrace{\sum_{X_{j}} f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})}_{M_{j \to i}}$$

$$f_{ij}^{n}(X_{i}, X_{j}) = \frac{f_{ij}^{n-1}(X_{i}, X_{j})}{M_{j \to i}(X_{i})}$$





$$p^{n}(X_{i}, X_{j}) = \frac{1}{Z_{ij}^{n}} t_{i}^{n-1}(X_{i}) t_{ij}^{n-1}(X_{i}, X_{j}) t_{j}^{n-1}(X_{j})$$

$$f_{i}^{n}(X_{i}) = p^{n}(X_{i}) = \sum_{X_{j}} p^{n}(X_{i}, X_{j}) = t_{i}^{n-1}(X_{i}) \underbrace{\sum_{X_{j}} t_{ij}^{n-1}(X_{i}, X_{j}) t_{j}^{n-1}(X_{j})}_{M_{j \to i}}$$





$$p^{n}(X_{i}, X_{j}) = \frac{1}{Z_{ij}^{n}} t_{i}^{n-1}(X_{i}) f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})$$

$$f_{i}^{n}(X_{i}) = p^{n}(X_{i}) = \sum_{X_{j}} p^{n}(X_{i}, X_{j}) = f_{i}^{n-1}(X_{i}) \underbrace{\sum_{X_{j}} f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})}_{M_{j \to i}(X_{i})}$$





$$p^{n}(X_{i}, X_{j}) = \frac{1}{Z_{ij}^{n}} f_{i}^{n-1}(X_{i}) f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})$$

$$f_{i}^{n}(X_{i}) = p^{n}(X_{i}) = \sum_{X_{j}} p^{n}(X_{i}, X_{j}) = f_{i}^{n-1}(X_{i}) \underbrace{\sum_{X_{j}} f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})}_{M_{j \to i}}$$

$$f_{ij}^{n}(X_{i}, X_{j}) = \frac{f_{ij}^{n-1}(X_{i}, X_{j})}{M_{j \to i}(X_{i})}$$





$$p^{n}(X_{i}, X_{j}) = \frac{1}{Z_{ij}^{n}} t_{i}^{n-1}(X_{i}) t_{ij}^{n-1}(X_{i}, X_{j}) t_{j}^{n-1}(X_{j})$$

$$f_{i}^{n}(X_{i}) = p^{n}(X_{i}) = \sum_{X_{j}} p^{n}(X_{i}, X_{j}) = t_{i}^{n-1}(X_{i}) \underbrace{\sum_{X_{j}} t_{ij}^{n-1}(X_{i}, X_{j}) t_{j}^{n-1}(X_{j})}_{M_{j \to i}}$$



$$p^{n}(X_{i}, X_{j}) = \frac{1}{Z_{ij}^{n}} t_{i}^{n-1}(X_{i}) f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})$$

$$f_{i}^{n}(X_{i}) = p^{n}(X_{i}) = \sum_{X_{j}} p^{n}(X_{i}, X_{j}) = f_{i}^{n-1}(X_{i}) \underbrace{\sum_{X_{j}} f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})}_{M_{j \to i}(X_{i})}$$



$$p^{n}(X_{i}, X_{j}) = \frac{1}{Z_{ij}^{n}} f_{i}^{n-1}(X_{i}) f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})$$

$$f_{i}^{n}(X_{i}) = p^{n}(X_{i}) = \sum_{X_{j}} p^{n}(X_{i}, X_{j}) = f_{i}^{n-1}(X_{i}) \underbrace{\sum_{X_{j}} f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})}_{M_{j \to i}}$$

$$f_{ij}^{n}(X_{i}, X_{j}) = \frac{f_{ij}^{n-1}(X_{i}, X_{j})}{M_{j \to i}(X_{i})}$$





$$p^{n}(X_{i}, X_{j}) = \frac{1}{Z_{ij}^{n}} f_{i}^{n-1}(X_{i}) f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})$$

$$f_{i}^{n}(X_{i}) = p^{n}(X_{i}) = \sum_{X_{j}} p^{n}(X_{i}, X_{j}) = f_{i}^{n-1}(X_{i}) \underbrace{\sum_{X_{j}} f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})}_{M_{j \to i}}$$

$$f_{ij}^{n}(X_{i}, X_{j}) = \frac{f_{ij}^{n-1}(X_{i}, X_{j})}{M_{j \to i}(X_{i})}$$





$$p^{n}(X_{i}, X_{j}) = \frac{1}{Z_{ij}^{n}} f_{i}^{n-1}(X_{i}) f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})$$

$$f_{i}^{n}(X_{i}) = p^{n}(X_{i}) = \sum_{X_{j}} p^{n}(X_{i}, X_{j}) = f_{i}^{n-1}(X_{i}) \underbrace{\sum_{X_{j}} f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})}_{M_{j \to i}}$$

$$f_{ij}^{n}(X_{i}, X_{j}) = \frac{f_{ij}^{n-1}(X_{i}, X_{j})}{M_{j \to i}(X_{i})}$$



$$p(\mathcal{X}) \propto \prod_{\substack{(ij) \\ \downarrow}} f_{ij}(X_i, X_j)$$
  
 $\downarrow$   
 $p(\mathcal{X}) = \prod_i p(X_i) \prod_{(ij)} \frac{p(X_i, X_j)}{p(X_i)p(X_k)}$ 

$$p^{n}(X_{i}, X_{j}) = \frac{1}{Z_{ij}^{n}} t_{i}^{n-1}(X_{i}) t_{ij}^{n-1}(X_{i}, X_{j}) t_{j}^{n-1}(X_{j})$$

$$f_{i}^{n}(X_{i}) = p^{n}(X_{i}) = \sum_{X_{j}} p^{n}(X_{i}, X_{j}) = t_{i}^{n-1}(X_{i}) \underbrace{\sum_{X_{j}} t_{ij}^{n-1}(X_{i}, X_{j}) t_{j}^{n-1}(X_{j})}_{M_{j \to i}}$$



$$p(\mathcal{X}) \propto \prod_{\substack{(ij) \\ \downarrow}} f_{ij}(X_i, X_j)$$
  
 $\downarrow$   
 $p(\mathcal{X}) = \prod_i p(X_i) \prod_{(ij)} \frac{p(X_i, X_j)}{p(X_i)p(X_k)}$ 

$$p^{n}(X_{i}, X_{j}) = \frac{1}{Z_{ij}^{n}} f_{i}^{n-1}(X_{i}) f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})$$

$$f_{i}^{n}(X_{i}) = p^{n}(X_{i}) = \sum_{X_{j}} p^{n}(X_{i}, X_{j}) = f_{i}^{n-1}(X_{i}) \underbrace{\sum_{X_{j}} f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})}_{M_{j \to i}(X_{i})}$$



$$p^{n}(X_{i}, X_{j}) = \frac{1}{Z_{ij}^{n}} f_{i}^{n-1}(X_{i}) f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})$$

$$f_{i}^{n}(X_{i}) = p^{n}(X_{i}) = \sum_{X_{j}} p^{n}(X_{i}, X_{j}) = f_{i}^{n-1}(X_{i}) \underbrace{\sum_{X_{j}} f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})}_{M_{j \to i}}$$

$$f_{ij}^{n}(X_{i}, X_{j}) = \frac{f_{ij}^{n-1}(X_{i}, X_{j})}{M_{j \to i}(X_{i})}$$



$$p^{n}(X_{i}, X_{j}) = \frac{1}{Z_{ij}^{n}} f_{i}^{n-1}(X_{i}) f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})$$

$$f_{i}^{n}(X_{i}) = p^{n}(X_{i}) = \sum_{X_{j}} p^{n}(X_{i}, X_{j}) = f_{i}^{n-1}(X_{i}) \underbrace{\sum_{X_{j}} f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})}_{M_{j \to i}}$$

$$f_{ij}^{n}(X_{i}, X_{j}) = \frac{f_{ij}^{n-1}(X_{i}, X_{j})}{M_{j \to i}(X_{i})}$$



Define  $f_{ij}^0 = f_{ij}$  (absorbing singleton factors), and  $f_i^0 = p_i^0 = 1$ . Iterate over edges (*ij*):

$$p^{n}(X_{i}, X_{j}) = \frac{1}{Z_{ij}^{n}} t_{i}^{n-1}(X_{i}) f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})$$

$$f_{i}^{n}(X_{i}) = p^{n}(X_{i}) = \sum_{X_{j}} p^{n}(X_{i}, X_{j}) = f_{i}^{n-1}(X_{i}) \underbrace{\sum_{X_{j}} f_{ij}^{n-1}(X_{i}, X_{j}) f_{j}^{n-1}(X_{j})}_{M_{j \to i}}$$

$$f_{ij}^{n}(X_{i}, X_{j}) = \frac{f_{ij}^{n-1}(X_{i}, X_{j})}{M_{j \to i}(X_{i})}$$

After all messages have propagated:

$$f_{i}^{\infty}(X_{i}) = \prod_{j \in ne(i)} M_{j \to i}(X_{i}) = p(X_{i})$$

$$f_{ij}^{\infty}(X_{i}, X_{j}) = \frac{\prod_{k \in ne(i) \setminus j} M_{k \to i}(X_{i})f_{ij}(X_{i}, X_{j}) \prod_{l \in ne(j) \setminus i} M_{l \to j}(X_{j})}{\prod_{k \in ne(i) \setminus j} M_{k \to i}(X_{i})M_{j \to i}(X_{i})M_{i \to j}(X_{j})} = \frac{p(X_{i}, X_{j})}{p(X_{i})p(X_{j})}$$

If BP converges on a non-tree, it will have successfully reparametrised the distribution to have locally consistent beliefs:

$$p(\mathcal{X}) \propto \prod_i b(X_i) \prod_{(ij)} rac{b(X_i, X_j)}{b(X_i)b(X_j)}$$
 with  $\sum_{X_j} b(X_i, X_j) = b(X_i)$  etc

However, the marginals will not usually be correct or globally consistent. That is

$$\sum_{\mathcal{X}_{\neg i}} \Big(\prod_{i} b(X_{i}) \prod_{(ij)} \frac{b(X_{i}, X_{j})}{b(X_{i})b(X_{j})}\Big) \neq b(X_{i})$$

and the product will not generally be normalised.

- What can be said about these pseudomarginals?
- Consider the following (theoretical) message scheduling scheme:
  - Identify all the spanning trees of the graph.
  - Pass messages along edges of each spanning tree in turn.
  - Iterate over spanning trees to convergence

### Loopy BP as tree-based reparametrisation

At convergence, loopy BP has reparametrised the joint distribution as:

$$p(\mathcal{X}) = rac{1}{Z} \prod_{ ext{nodes } i} f_i^\infty(X_i) \prod_{ ext{edges } (ij)} f_{ij}^\infty(X_i, X_j)$$

where for any tree T embedded in the graph,

$$f_{ij}^{\infty}(X_i) = p^{T}(X_i)$$
$$f_{ij}^{\infty}(X_i, X_j) = \frac{p^{T}(X_i, X_j)}{p^{T}(X_i)p^{T}(X_j)}$$

Thus, the local marginals of all subtrees are locally consistent with each other, and the pseudomarginals represent valid beliefs for any of the subtrees.

$$p(\mathcal{X}) = \frac{1}{Z} \prod_{\text{nodes } i} b_i(X_i) \prod_{\text{edges } (j)} \frac{b_{ij}(X_i, X_j)}{b_i(X_i)b_j(X_j)}$$

## **Different Interpretations of Loopy Belief Propagation**

Loopy BP can be interpreted as a fixed point algorithm from a few different perspectives:

- Expectation propagation.
- Tree-based reparametrization.
- Bethe free energy.

In the reparametrisation view, BP solves for marginal beliefs  $b_{ij}(X_i, X_j)$  and  $b_i(X_i) = \sum_{X_i} b_{ij}(X_i, X_j)$  such that

$$p(\mathcal{X}) \propto \prod_i f_i(X_i) \prod_{(ij)} f_{ij}(X_i, X_j) \propto \prod_i b_i(X_i) \prod_{(ij)} rac{b_{ij}(X_i, X_j)}{b_i(X_i) b_j(X_j)}$$

In the reparametrisation view, BP solves for marginal beliefs  $b_{ij}(X_i, X_j)$  and  $b_i(X_i) = \sum_{X_i} b_{ij}(X_i, X_j)$  such that

$$p(\mathcal{X}) \propto \prod_i f_i(X_i) \prod_{(ij)} f_{ij}(X_i, X_j) \propto \prod_i b_i(X_i) \prod_{(ij)} \frac{b_{ij}(X_i, X_j)}{b_i(X_i)b_j(X_j)}$$

Another view of loopy BP is as a set of fixed point equations for finding stationary points of an objective function called the Bethe free energy, which is defined in terms of the locally consistent beliefs (or pseudomarginals)  $b_i \ge 0$  and  $b_{ij} \ge 0$ :

$$\sum_{x_i} b_i(x_i) = 1 \qquad \forall i$$
$$\sum_{x_j} b_{ij}(x_i, x_j) = b_i(x_i) \qquad \forall i, j \in ne(i), x_i$$

Recall that the variational free energy is:  $\mathcal{F}(q) = \langle \log P(\mathcal{X}) \rangle_q + \mathbf{H}[q]$ 

Recall that the variational free energy is:  $\mathcal{F}(q) = \langle \log P(\mathcal{X}) \rangle_q + \mathbf{H}[q]$ 

We define the (negative) Bethe free energy:  $\mathcal{F}_{\text{bethe}}(b) = \mathcal{E}_{\text{bethe}}(b) + \mathcal{H}_{\text{bethe}}(b)$  where both terms are approximations to the corresponding variational likelihood terms.

Recall that the variational free energy is:  $\mathcal{F}(q) = \langle \log \mathcal{P}(\mathcal{X}) 
angle_q + \mathbf{H}[q]$ 

We define the (negative) Bethe free energy:  $\mathcal{F}_{\text{bethe}}(b) = \mathcal{E}_{\text{bethe}}(b) + \mathcal{H}_{\text{bethe}}(b)$  where both terms are approximations to the corresponding variational likelihood terms.

The Bethe average energy is the expected log-joint evaluated as though the pseudomarginals were correct:

$$\mathcal{E}_{\mathsf{bethe}}(b) = \sum_i \sum_{x_i} b_i(x_i) \log f_i(x_i) + \sum_{(ij)} \sum_{x_i, x_j} b_{ij}(x_i, x_j) \log f_{ij}(x_i, x_j)$$

Recall that the variational free energy is:  $\mathcal{F}(q) = \langle \log \mathcal{P}(\mathcal{X}) 
angle_q + \mathbf{H}[q]$ 

We define the (negative) Bethe free energy:  $\mathcal{F}_{\text{bethe}}(b) = \mathcal{E}_{\text{bethe}}(b) + \mathcal{H}_{\text{bethe}}(b)$  where both terms are approximations to the corresponding variational likelihood terms.

The Bethe average energy is the expected log-joint evaluated as though the pseudomarginals were correct:

$$\mathcal{E}_{\mathsf{bethe}}(b) = \sum_i \sum_{x_i} b_i(x_i) \log f_i(x_i) + \sum_{(ij)} \sum_{x_i, x_j} b_{ij}(x_i, x_j) \log f_{ij}(x_i, x_j)$$

The Bethe entropy is the sum of the pseudomarginal entropies corrected for pairwise (pseudo)interactions, but neglecting higher-order dependence:

$$\begin{split} \mathcal{H}_{\text{bethe}}(b) &= \sum_{i} \mathbf{H}[b_{i}] - \sum_{(ij)} \mathbf{KL}[b_{ij} || b_{i} b_{j}] \\ &= -\sum_{i} \sum_{x_{i}} b_{i}(x_{i}) \log b_{i}(x_{i}) - \sum_{(ij)} \sum_{x_{i}, x_{j}} b_{ij}(x_{i}, x_{j}) \log \frac{b_{ij}(x_{i}, x_{j})}{b_{i}(x_{i}) b_{j}(x_{j})} \end{split}$$

Recall that the variational free energy is:  $\mathcal{F}(q) = \langle \log \mathcal{P}(\mathcal{X}) 
angle_q + \mathbf{H}[q]$ 

We define the (negative) Bethe free energy:  $\mathcal{F}_{\text{bethe}}(b) = \mathcal{E}_{\text{bethe}}(b) + \mathcal{H}_{\text{bethe}}(b)$  where both terms are approximations to the corresponding variational likelihood terms.

The Bethe average energy is the expected log-joint evaluated as though the pseudomarginals were correct:

$$\mathcal{E}_{\mathsf{bethe}}(b) = \sum_i \sum_{x_i} b_i(x_i) \log f_i(x_i) + \sum_{(ij)} \sum_{x_i, x_j} b_{ij}(x_i, x_j) \log f_{ij}(x_i, x_j)$$

The Bethe entropy is the sum of the pseudomarginal entropies corrected for pairwise (pseudo)interactions, but neglecting higher-order dependence:

$$\begin{split} \mathcal{H}_{\text{bethe}}(b) &= \sum_{i} \mathbf{H}[b_{i}] - \sum_{(ij)} \mathbf{KL}[b_{ij} || b_{i} b_{j}] \\ &= -\sum_{i} \sum_{x_{i}} b_{i}(x_{i}) \log b_{i}(x_{i}) - \sum_{(ij)} \sum_{x_{i}, x_{j}} b_{ij}(x_{i}, x_{j}) \log \frac{b_{ij}(x_{i}, x_{j})}{b_{i}(x_{i}) b_{j}(x_{j})} \end{split}$$

• On a tree, both the beliefs and the Bethe entropy expression are correct, so  $\mathcal{F}_{bethe} = \mathcal{F}$ .

Recall that the variational free energy is:  $\mathcal{F}(q) = \langle \log \mathcal{P}(\mathcal{X}) 
angle_q + \mathbf{H}[q]$ 

We define the (negative) Bethe free energy:  $\mathcal{F}_{\text{bethe}}(b) = \mathcal{E}_{\text{bethe}}(b) + \mathcal{H}_{\text{bethe}}(b)$  where both terms are approximations to the corresponding variational likelihood terms.

The Bethe average energy is the expected log-joint evaluated as though the pseudomarginals were correct:

$$\mathcal{E}_{\mathsf{bethe}}(b) = \sum_i \sum_{x_i} b_i(x_i) \log f_i(x_i) + \sum_{(ij)} \sum_{x_i, x_j} b_{ij}(x_i, x_j) \log f_{ij}(x_i, x_j)$$

The Bethe entropy is the sum of the pseudomarginal entropies corrected for pairwise (pseudo)interactions, but neglecting higher-order dependence:

$$\begin{split} \mathcal{H}_{\text{bethe}}(b) &= \sum_{i} \mathsf{H}[b_{i}] - \sum_{(ij)} \mathsf{KL}[b_{ij} || b_{i}b_{j}] \\ &= -\sum_{i} \sum_{x_{i}} b_{i}(x_{i}) \log b_{i}(x_{i}) - \sum_{(ij)} \sum_{x_{i}, x_{j}} b_{ij}(x_{i}, x_{j}) \log \frac{b_{ij}(x_{i}, x_{j})}{b_{i}(x_{i})b_{j}(x_{j})} \end{split}$$

- On a tree, both the beliefs and the Bethe entropy expression are correct, so  $\mathcal{F}_{bethe} = \mathcal{F}$ .
- Message updates in loopy BP can now be derived by finding the stationary points of a Lagrangian with local consistency and normalisation constraints. The BP messages are related to the Lagrange multipliers.

### Bethe fixed point equations

The Bethe free-energy Lagrangian is:

$$\mathcal{L} = \sum_{i} \sum_{x_{i}} b_{i}(x_{i}) \log f_{i}(x_{i}) + \sum_{(ij)} \sum_{x_{i}, x_{j}} b_{ij}(x_{i}, x_{j}) \log f_{ij}(x_{i}, x_{j})$$

$$= \sum_{i} \sum_{x_{i}} b_{i}(x_{i}) \log b_{i}(x_{i}) - \sum_{(ij)} \sum_{x_{i}, x_{j}} b_{ij}(x_{i}, x_{j}) \log \frac{b_{ij}(x_{i}, x_{j})}{b_{i}(x_{i})b_{j}(x_{j})}$$

$$= \sum_{i} \sum_{x_{i}} b_{i}(x_{i}) \log b_{i}(x_{i}) - \sum_{(ij)} \sum_{x_{i}, x_{j}} b_{ij}(x_{i}, x_{j}) \log \frac{b_{ij}(x_{i}, x_{j})}{b_{i}(x_{i})b_{j}(x_{j})}$$

$$= \sum_{i} \sum_{x_{i}} b_{i}(x_{i}) \log b_{i}(x_{i}) - \sum_{(ij)} \sum_{x_{i}, x_{j}} b_{ij}(x_{i}, x_{j}) \log \frac{b_{ij}(x_{i}, x_{j})}{b_{i}(x_{i})b_{j}(x_{j})}$$

$$= \sum_{i} \sum_{x_{i}} b_{i}(x_{i}) \log b_{i}(x_{i}) - \sum_{(ij)} b_{ij}(x_{i}, x_{j}) - b_{i}(x_{i}) \log \frac{b_{ij}(x_{i}, x_{j})}{b_{i}(x_{i})b_{j}(x_{i}, x_{j}) - b_{j}(x_{j})}$$

$$= \sum_{i} \sum_{x_{i}} b_{i}(x_{i}, x_{i}) \log b_{i}(x_{i}, x_{j}) - b_{i}(x_{i}) + \sum_{x_{j}} \xi_{ji}(x_{j}) \left(\sum_{x_{i}} b_{ij}(x_{i}, x_{j}) - b_{j}(x_{j})\right) \right]$$

$$= \sum_{i} \sum_{i} \sum_{x_{i}} b_{i}(x_{i}) \log b_{i}(x_{i}, x_{j}) - b_{i}(x_{i}) + \sum_{x_{j}} \xi_{ji}(x_{j}) \left(\sum_{x_{i}} b_{ij}(x_{i}, x_{j}) - b_{j}(x_{j})\right) \right]$$

$$= \sum_{i} \sum_{x_{i}} b_{i}(x_{i}) \log b_{i}(x_{i}) + \sum_{x_{j}} \xi_{ji}(x_{j}) \left(\sum_{x_{i}} b_{ij}(x_{i}, x_{j}) - b_{i}(x_{i})\right) + \sum_{x_{j}} \xi_{ji}(x_{j}) \left(\sum_{x_{i}} b_{ij}(x_{i}, x_{j}) - b_{j}(x_{j})\right) \right]$$

$$= \sum_{i} \sum_{x_{i}} b_{i}(x_{i}) \log b_{i}(x_{i}) + \sum_{x_{j}} b_{ij}(x_{i}, x_{j}) + b_{i}(x_{j}) \log b_{i}(x_{j}) + \sum_{x_{j}} b_{ij}(x_{j}) \log b_{i}(x_{j}) + \sum_{x_{j}} b_{i}(x_{j}) b_{i}(x_{j}) + \sum_{x_{j}}$$

Xi

Setting derivatives wrt beliefs to 0 gives

$$\frac{\partial \mathcal{L}}{\partial b_i(x_i)} = \log f_i(x_i) - \log b_i(x_i) + \sum_{j \in ne(i)} \underbrace{\sum_{ij} \frac{b_{ij}(x_i, x_j)}{b_i(x_i)}}_{=1 \text{ by constraint}} + \xi_i - \sum_{j \in ne(i)} \xi_{ij}(x_i) + const = 0$$

$$\Rightarrow b_i(x_i) \propto f_i(x_i) \prod_{j \in \mathsf{ne}(i)} e^{-\xi_{ij}(x_i)}$$

 $\frac{\partial \mathcal{L}}{\partial b_{ij}(x_i, x_j)} = \log f_{ij}(x_i, x_j) - \log b_{ij}(x_i, x_j) + \log b_i(x_i)b_j(x_j) + \xi_{ij}(x_i) + \xi_{ji}(x_j) + const = 0$  $\Rightarrow b_{ij}(x_i, x_j) \propto f_{ij}(x_i, x_j)b_i(x_i)b_j(x_j)e^{\xi_{ij}(x_i)}e^{\xi_{ij}(x_j)}e^{\xi_{ij}(x_j)}$ 

#### Bethe fixed point messages

The Bethe Lagrangian fixed point equations are:

$$egin{aligned} b_i(x_i) \propto f_i(x_i) \prod_{j \in \mathsf{ne}(i)} e^{-\xi_{ij}(x_i)} \ b_{ij}(x_i,x_j) \propto f_{ij}(x_i,x_j) b_i(x_i) b_j(x_j) e^{\xi_{ij}(x_i)} e^{\xi_{ij}(x_j)} \end{aligned}$$

Comparison with BP suggests that messages should have the form  $M_{j \to i}(x_i) = e^{-\xi_{ij}(x_i)}$ . Indeed, solving for  $\xi_{ij}(x_i)$  by enforcing the constraint  $\sum_{x_i} b_{ij}(x_i, x_j) = b_i(x_i)$  we have:

$$\begin{split} \sum_{x_j} b_{ij}(x_i, x_j) &\propto \sum_{x_j} f_{ij}(x_i, x_j) b_i(x_i) b_j(x_j) e^{\xi_{ij}(x_i)} e^{\xi_{ij}(x_j)} \\ &\Rightarrow b_i(x_i) \propto b_i(x_i) e^{\xi_{ij}(x_i)} \sum_{x_j} f_{ij}(x_i, x_j) b_j(x_j) e^{\xi_{jj}(x_j)} \\ &\Rightarrow e^{-\xi_{ij}(x_i)} \propto \sum_{x_j} f_{ij}(x_i, x_j) b_j(x_j) e^{\xi_{ji}(x_j)} \\ &= \sum_{x_j} f_{ij}(x_i, x_j) f_j(x_j) \prod_{l \in \mathsf{ne}(j) \setminus i} e^{-\xi_{jj}(x_j)} \end{split}$$

thus recovering the BP message passing rules.

- Fixed points of loopy BP are exactly the stationary points of the Bethe free energy.
- Stable fixed points of loopy BP are local maxima of Bethe free enegy (note the negative definition of free energy for consistency with the variational free energy).
- For binary attractive networks, Bethe free energy at fixed points of loopy BP provides an upper bound on the log partition function log Z—this is useful for learning undirected graphical models as it leads to a lower bound on the log likelihood.

## Loopy BP vs mean-field approximation

- Beliefs b<sub>i</sub> and b<sub>ij</sub> in loopy BP are only locally consistent pseudomarginals, not necessarily consistent marginals of the implied joint distribution.
- Bethe free energy accounts for interactions between different sites, while variational free energy assumes independence.
- The loop series or Plefka expansion of the log partition function Z: the variational free energy forms the first order terms, while Bethe free energy contains higher order terms (involving generalized loops).
- Loopy BP tends to be significantly more accurate whenever it converges.

## **Extensions and variations**

- Generalized BP: group variables together to treat their interactions exactly.
- Convergent alternatives: Fixed points of loopy BP are stationary points of the Bethe free enegy. We can also derive algorithms that **increase** the Bethe free energy at every step, and are thus are guaranteed to converge.



- Convex alternatives: We can derive convex cousins of the negative of the Bethe free energy. These give rise to algorithms that will converge to a unique global maximum.
- We have considered sum-product loopy BP to compute marginals. The treatment of loopy Viterbi or max-product algorithms is different.

## References

- Probabilistic Reasoning in Intelligent Systems. J. Pearl. Morgan Kaufman, 1988.
- Turbo decoding as an instance of Pearl's belief propagation algorithm. R. J. McEliece, D. J. C. MacKay and J. F. Cheng. IEEE Journal on Selected Areas in Communication, 1998, 16(2):140-152.
- Iterative decoding of compound codes by probability propagation in graphical models. F. Kschischang and B. Frey. IEEE Journal on Selected Areas in Communication, 1998, 16(2):219-230.
- A family of algorithms for approximate Bayesian inference. T. Minka. PhD Thesis, 2001.
- Tree-based reparameterization framework for analysis of sum-product and related algorithms. M. J. Wainwright, T. S. Jaakkola and A. S. Willsky. IEEE Transactions on Information Theory, 2004, 49(5).
- Constructing free energy approximations and generalized belief propagation algorithms. J. S. Yedidia, W. T. Freeman and Y. Weiss. IEEE Transactions on Information Theory, 2005, 51:2282-2313.