Probabilistic & Unsupervised Learning

Belief Propagation

Yee Whye Teh
ywteh@gatsby.ucl.ac.uk

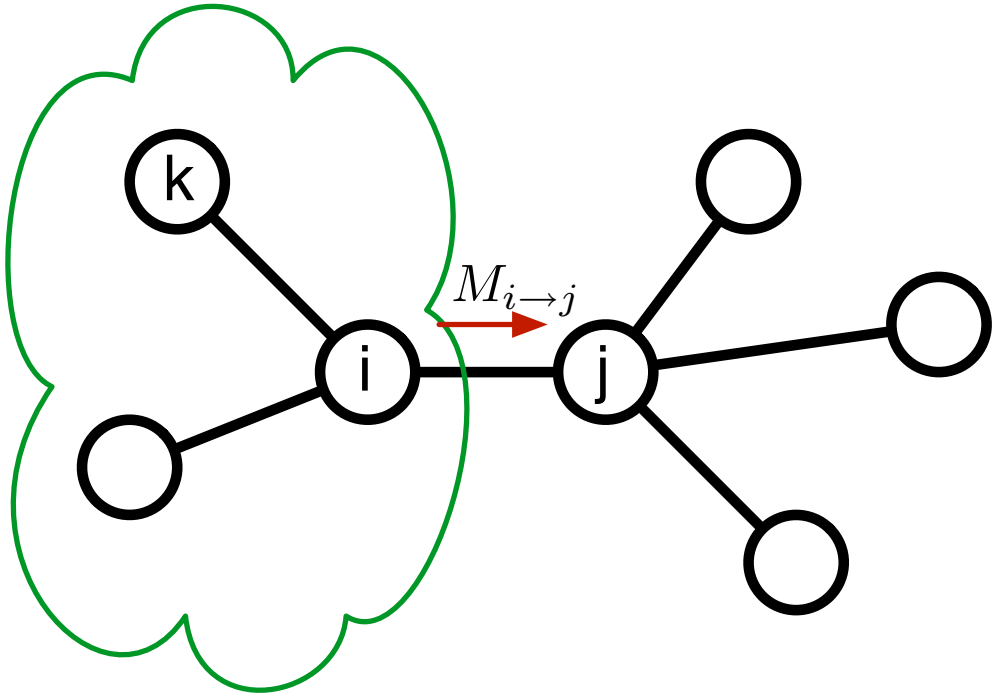
Gatsby Computational Neuroscience Unit
University College London

Term 1, Autumn 2008

Recap: Belief Propagation on Undirected Trees

Joint distribution of undirected tree:

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



Recursively compute messages:

$$M_{i \to j}(X_j) := \sum_{X_i} f_{ij}(X_i, X_j) f_i(X_i) \prod_{k \in \text{ne}(i) \setminus j} M_{k \to i}(X_i)$$

Marginal distributions:

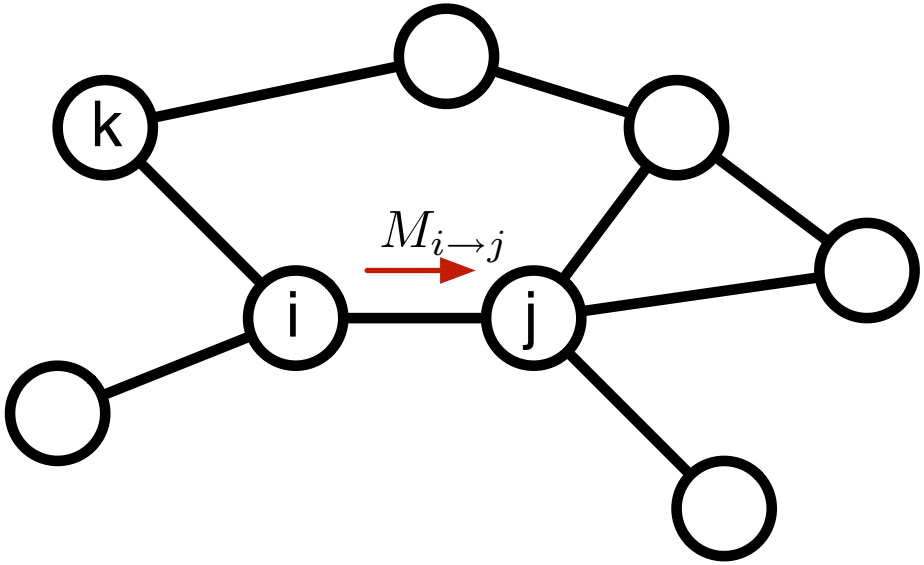
$$p(X_i) \propto f_i(X_i) \prod_{k \in \text{ne}(i)} M_{k \to i}(X_i)$$

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

Loopy Belief Propagation

Joint distribution of undirected graph:

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



Recursively compute messages (and hope that updates converge):

$$M_{i \to j}(X_j) := \sum_{X_i} f_{ij}(X_i, X_j) f_i(X_i) \prod_{k \in \text{ne}(i) \setminus j} M_{k \to i}(X_i)$$

Approximate marginal distributions:

$$p(X_i) \approx b_i(X_i) \propto f_i(X_i) \prod_{k \in \text{ne}(i)} M_{k \to i}(X_i)$$

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

Practical Considerations

- **Convergence**: Loopy BP is not guaranteed to converge for most graphs.
  - Trees: BP will converge.
  - Single loop: BP will converge for graphs containing at most one loop.
  - Weak interactions: BP will converge for graphs with weak enough interactions.
  - Long loops: BP more likely to converge for graphs with long (weakly interacting) loops.
  - Gaussian networks: Means correct, variances many converge under some conditions.
- **Damping**: Popular approach to encourage convergence.

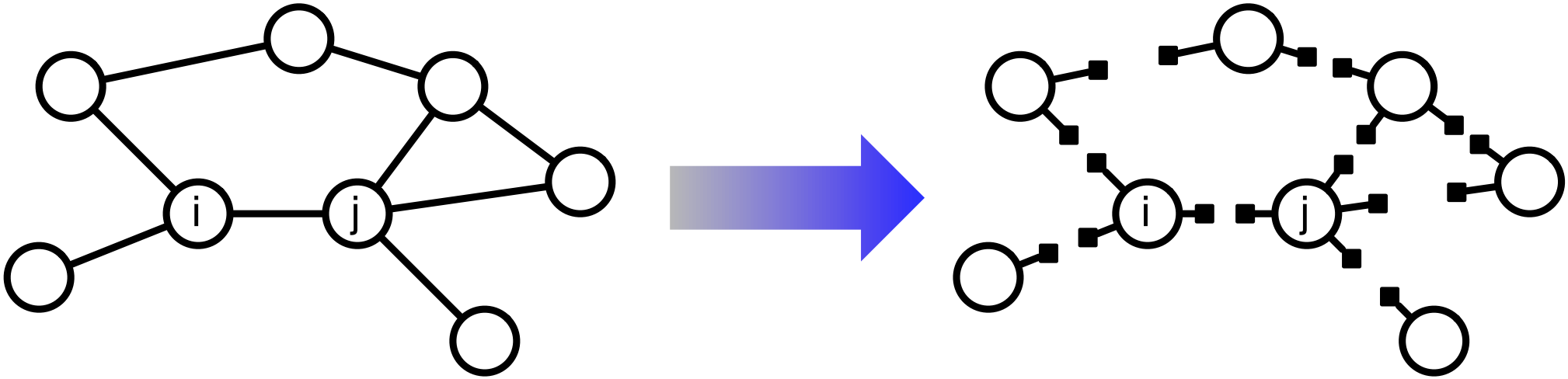
$$M_{i\to j}^{\text{new}}(X_j) := (1-\alpha)M_{i\to j}^{\text{old}}(X_j) + \alpha \sum_{X_i} f_{ij}(X_i, X_j) f_i(X_i) \prod_{k \in \text{ne}(i)\setminus j} M_{k\to i}(X_i)$$

- **Convergent alternatives**: to loopy BP.
  - Algorithms with same fixed points as BP, but guaranteed to converge.
  - Algorithms based on same principles as BP.
- **Other graphical models**: equivalent formulations for DAG and factor graphs.
- **Grouping variables**: Multiple variables can be grouped into regions to improve accuracy.
  - Region graph approximations.
  - Cluster variational method.
  - Junction graph.

Different Perspectives on Loopy Belief Propagation

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

Loopy BP as Expectation Propagation



Approximate each factor f_{ij} describing interaction between i and j as:

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

The full joint distribution is thus approximated by a factorized distribution:

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

Loopy BP as Expectation Propagation

Each EP update to $\tilde{f}_{ij}$ is as follows:

- "Corrected" distribution is:

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

- Moments are just marginal distributions on i and j.
- Thus optimal $\tilde{f}_{ij}(X_i, X_j)$ minimizing

$$\mathbf{KL}[f_{ij}(X_i, X_j)q_{\neg ij}(\mathbf{X}) \| \tilde{f}_{ij}(X_i, X_j)q_{\neg ij}(\mathbf{X})]$$

is given by:

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

Similarly for $M_{j\to i}(X_i)$.

Loopy BP as Tree-based Reparametrization

Many ways of parametrizing tree-structured distributions.

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

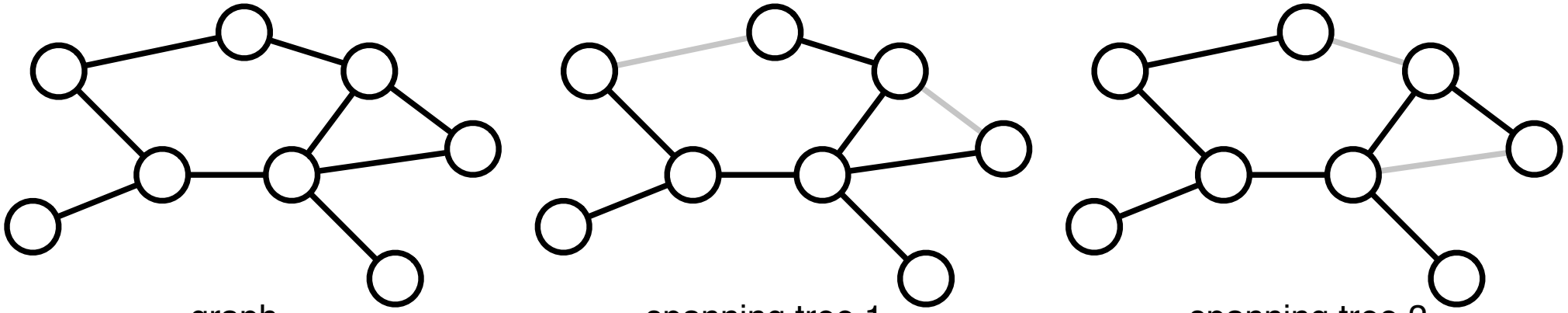
$$= p(X_r) \prod_{i \neq r} p(X_i | X_{\text{pa}(i)}) \qquad \text{directed (rooted) tree} \qquad (2)$$

$$= \prod_{\text{nodes } i} p(X_i) \prod_{\text{edges } (ij)} \frac{p(X_i, X_j)}{p(X_i)p(X_j)} \qquad \text{locally consistent marginals} \qquad (3)$$

Undirected tree representation is redundant—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 on tree can be understood as reparametrizing (1) by locally consistent factors. This results in (3), from which the local marginals can be read off.

Loopy BP as Tree-based Reparametrization



graph spanning tree 1 spanning tree 2

$$p(\mathbf{X}) = \frac{1}{Z} \prod_{\text{nodes } i} f_i^0(X_i) \prod_{\text{edges } (ij)} f_{ij}^0(X_i, X_j)$$

$$= \frac{1}{Z} \prod_{\text{nodes } i \in T_1} f_i^0(X_i) \prod_{\text{edges } (ij) \in T_1} f_{ij}^0(X_i, X_j) \prod_{\text{edges } (ij) \notin T_1} f_{ij}^0(X_i, X_j)$$

$$= \frac{1}{Z} \prod_{\text{nodes } i \in T_1} f_i^1(X_i) \prod_{\text{edges } (ij) \in T_1} f_{ij}^1(X_i, X_j) \prod_{\text{edges } (ij) \notin T_1} f_{ij}^1(X_i, X_j)$$

where $f_i^1(X_i) = p^{T_1}(X_i)$, $f_{ij}^1(X_i, X_j) = \frac{p^{T_1}(X_i, X_j)}{p^{T_1}(X_i) p^{T_1}(X_j)}$, $f_{ij}^1 = f_{ij}^0$.

$$= \frac{1}{Z} \prod_{\text{nodes } i \in T_2} f_i^1(X_i) \prod_{\text{edges } (ij) \in T_2} f_{ij}^1(X_i, X_j) \prod_{\text{edges } (ij) \notin T_2} f_{ij}^1(X_i, X_j)$$

$\ldots$

Loopy BP as Tree-based Reparametrization

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

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

where for any tree T embedded in the graph,

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

In particular, all local marginals of all trees are locally consistent with each other:

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

Loopy BP as Optimizing Bethe Free Energy

$$p(\mathbf{X}) = \frac{1}{Z} \prod_i f_i(X_i) \prod_{(ij)} f_{ij}(X_i, X_j)$$

Loopy BP can be derived as fixed point equations for finding stationary points of an objective function called the Bethe free energy.

The Bethe free energy is not optimized wrt a full distribution over $\mathbf{X}$, rather over locally consistent pseudomarginals or beliefs $b_i \geq 0$ and $b_{ij} \geq 0$:

$$\sum_{X_i} b_i(X_i) = 1 \qquad \forall i$$

$$\sum_{X_j} b_{ij}(X_i, X_j) = b_i(X_i) \qquad \forall i, j \in \mathsf{ne}(i)$$

Loopy BP as Optimizing Bethe Free Energy

$$\mathcal{F}_{\text{bethe}}(b) = \mathcal{E}_{\text{bethe}}(b) + \mathcal{H}_{\text{bethe}}(b)$$

The Bethe average energy is "exact":

$$\mathcal{E}_{\text{bethe}}(b) = \sum_i \sum_{X_i} b_i(X_i) \log f_i(X_i) + \sum_{(ij)} \sum_{X_i, X_j} b_{ij}(X_i, X_j) \log f_{ij}(X_i, X_j)$$

While the Bethe entropy is approximate:

$$\mathcal{H}_{\text{bethe}}(b) = -\sum_i \sum_{X_i} b_i(X_i) \log b_i(X_i) - \sum_{(ij)} \sum_{X_i, X_j} b_{ij}(X_i, X_j) \log \frac{b_{ij}(X_i, X_j)}{b_i(X_i) b_j(X_j)}$$

Factors in denominator are to account for overcount of entropy on edges, so that the Bethe entropy is exact on trees.

Message updates in loopy BP can now derived by finding the stationary points of the Lagrangian (with Lagrange multipliers included to enforce local consistency). Messages are related to the Lagrange multipliers.

Loopy BP as Optimizing Bethe Free Energy

- Fixed points of loopy BP are exactly the stationary points of the Bethe free energy.
- Stable fixed points of loopy BP are local maximum of Bethe free enegy (note we used inverted notion of free energy to be consistent with the variational free energy).
- For binary attractive networks, Bethe free energy at fixed points of loopy BP forms lower bound on log partition function $\log Z$.

Loopy BP vs Variational Approximation

- Beliefs b_i and b_{ij} in loopy BP are only locally consistent pseudomarginals and do not necessarily form a full joint distribution.

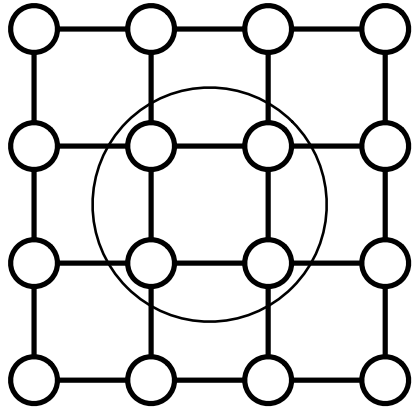
- Bethe free energy accounts for interactions between different sites, while variational free energy assumes independence.

- 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 signficantly 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 derive algorithms **minimizing** the Bethe free energy thus are guaranteed to converge.

- Convex alternatives: We can derive convex cousins of Bethe free energy. These give rise to algorithms that will converge to the unique global minimum.

- 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.

References