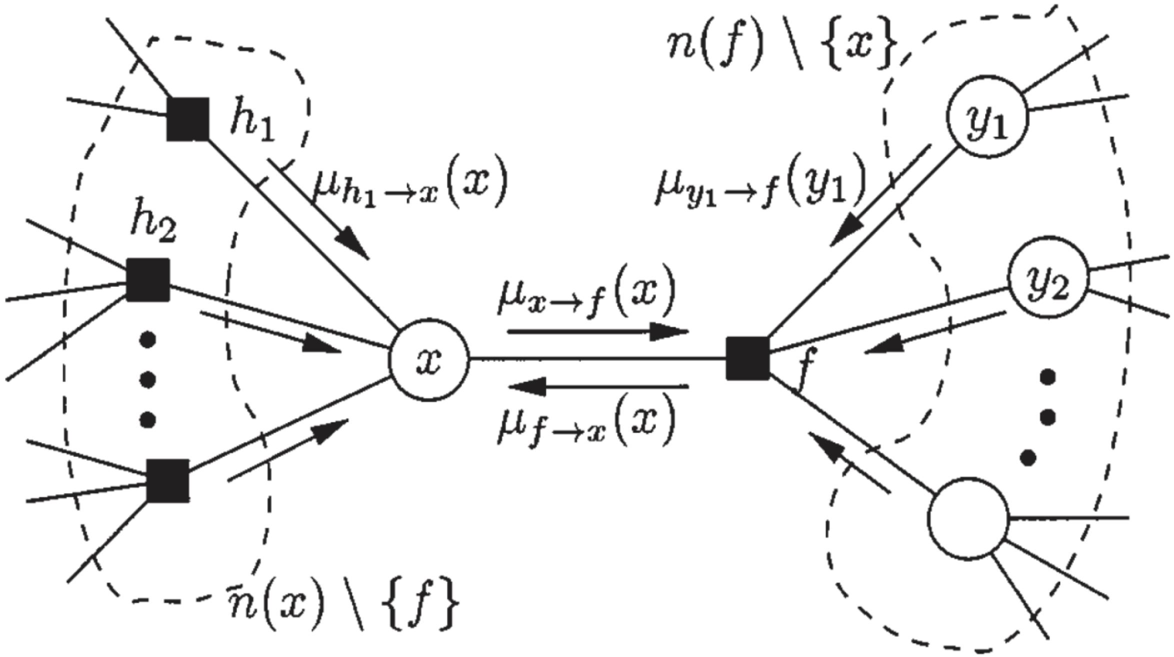
Approximate Bayesian Inference: Loopy belief propagation

Prof. Nicholas Zabaras
Center for Informatics and Computational Science
https://cics.nd.edu/
University of Notre Dame
Notre Dame, IN, USA

Email: nzabaras@gmail.com
URL: https://www.zabaras.com/

April 12, 2018

BP on a factor graph



- At convergence, we compute the final belief as a product of incoming messages:

$$bel(x) \propto \prod_{f \in nbr(x)} m_{f \to x}(x)$$

BP on a factor graph

- We now derive a version of BP that sends messages on a factor graph, as proposed in (Kschischang et al. 2001).

- We now have two kinds of messages
  - variables to factors

$$m_{x\to f}(x) = \prod_{h\in nbr(x)\backslash\{f\}} m_{h\to x}(x)$$

  - Factors to variables

$$m_{x\to f}(x) = \sum_{\mathbf{y}} f(x,\mathbf{y}) \prod_{y\in nbr(f)\backslash\{x\}} m_{y\to f}(y)$$

Here $nbr(x)$ are all factors that are connected to variable x and $nbr(f)$ are all variables connected to factor f.

Loopy belief propagation: algorithmic issues

- When applied to loopy graphs, belief propagation is not guaranteed to give correct results

- According to Judea Pearl,
  - When loops are present, the network is no longer singly connected and <u>local propagation will invariably run into trouble</u>.
  - If we permit the nodes to continue communicating with each other as if the network were singly connected, messages may circulate indefinitely around the loops.
  - Such oscillations **do not** normally occur in probabilistic networks which tend to bring all messages to equilibrium.
  - However, this asymptotic equilibrium is not coherent, in the sense that it does not represent the posterior probabilities of all nodes of the network

- Despite these reservations, Pearl advocated the use of belief propagation in loopy networks

Loopy belief propagation: algorithmic issues

- Interest in BP actually increases when McEliece et al. (1998) showed that a popular algorithm for error correcting codes could be viewed as an instance of BP applied to a certain kind of graph.
- This was an important observation since turbo codes have gotten very close to the theoretical lower bound on coding efficiency proved by Shannon
- In (Murphy et al. 1999), LBP was experimentally shown to also work well for inference in other kinds of graphical models beyond the error-correcting code context.
- Since then, the method has been widely used in many different applications

LBP on pairwise model

- To apply LBP on pairwise model, we proceed as:
  - Initialize all messages to the all 1's vector
  - In parallel, all nodes absorbs messages from its neighbors

$$\mathrm{bel}_s(x_s) \propto \psi_s(x_s) \prod_{t \in \mathrm{nbr}_s} m_{t\to s}(x_s)$$

  - Then, in parallel, each node sends messages to each of its neighbors

$$m_{s\to t}(x_t) = \sum_{x_s} \left(\psi_s(x_s)\psi_{st}(x_s, x_t) \prod_{u \in \mathrm{nbr}_s \backslash \mathrm{t}} m_{u\to s}(x_s) \right)$$

$m_{s\to t}(x_t)$ message is basically computed by multiplying together all incoming messages, except the one sent by the recipient and then passing through the potential.

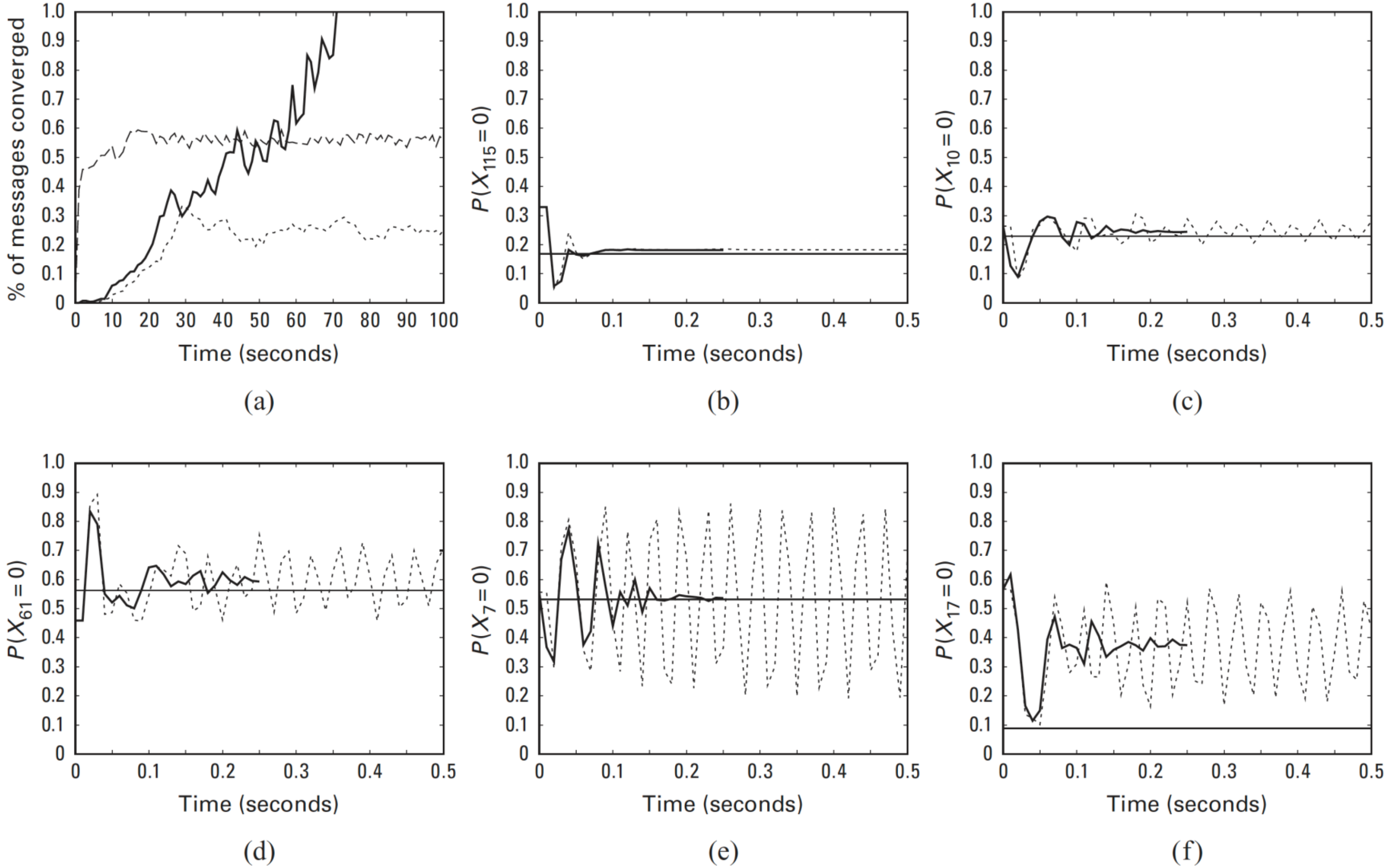
  - The above steps are repeated until convergence (i.e., no significant change in beliefs).

Convergence

- LBP does not always converge, and even when it does, it may converge to the wrong answers.

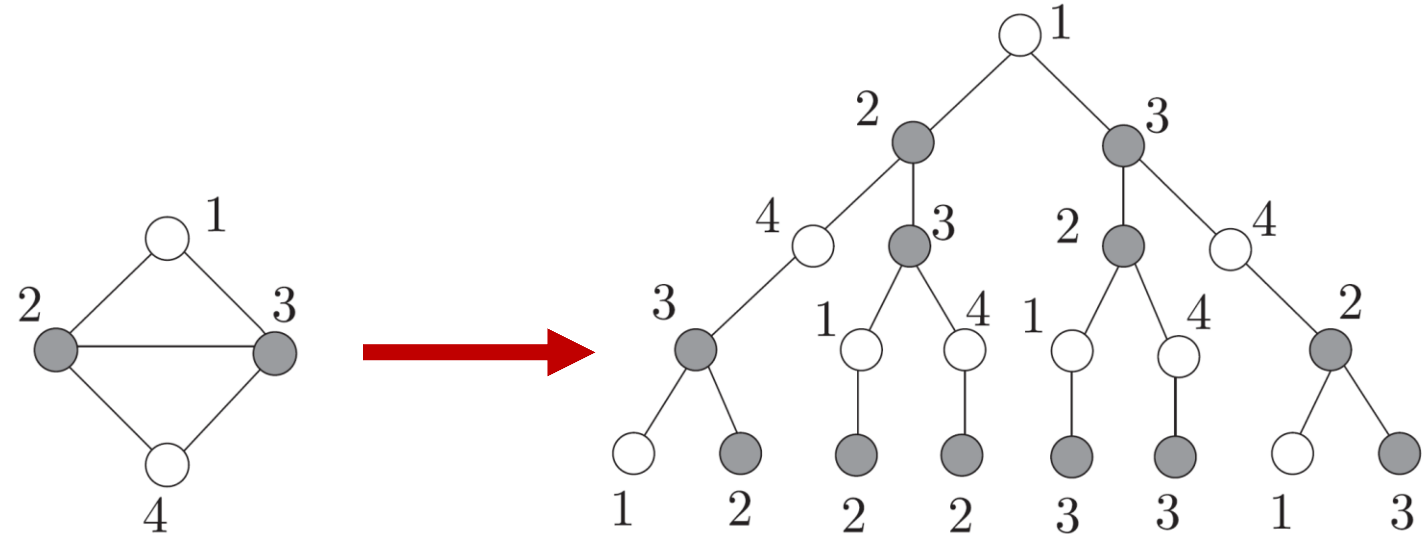
- This raises several questions
  - how can we predict when convergence will occur?
  - what can we do to increase the probability of convergence?
  - what can we do to increase the rate of convergence?

Convergence



When will LBP converge?

- To understand the LBP convergence, the key analysis tool is the computation tree.
- Computation tree visualizes the messages that are passed as the algorithm proceeds.



Computation tree rooted at node 1 after 4 rounds of message passing

When will LBP converge?

- The key insight is that T iterations of LBP is equivalent to exact computation in a computation tree of height $T + 1$.

- If the strengths of the connections on the edges is sufficiently weak, then the influence of the leaves on the root will diminish over time, and convergence will occur

Making LBP converge

- One simple way to reduce the chance of oscillation is to use **damping.**

- That is, instead of sending the message M_{ts}^k, we send a damped message of the form

$$M_{ts}^k(x_s) = \lambda M_{ts}(x_s) + (1-\lambda) M_{ts}^{k-1}(x_s)$$

where $0 \leq \lambda \leq 1$.

- Clearly, if $\lambda = 1$, it reduces to standard message passing.

- A standard practice is to use $\lambda \approx 0.5$.

Making LBP converge

- It is possible to devise methods, known as double loop algorithms
- These algorithms are guaranteed to converge to a local minimum of the same objective that LBP is minimizing
- Unfortunately, these methods are rather slow and complicated, and the accuracy of the resulting marginals is usually not much greater than with standard LBP.
- Consequently, these techniques are not very widely used

Increasing the convergence rate: message scheduling

- Even if LBP converges, it may take a long time.
- The standard approach when implementing LBP is to perform **synchronous updates**
  - all nodes absorb messages in parallel, and then send out messages in parallel.
- The new messages at iteration $k+1$ are computed in parallel using

$$m^{k+1} = \left(f_1(m^k), \ldots, f_E(m^k) \right)$$

where E is the number of edges.

- This is analogous to the Jacobi method for solving linear systems of equations

Increasing the convergence rate: message scheduling

- Now, Gauss-Seidel method converges faster when solving linear systems of equations.
- It performs asynchronous updates in a fixed round-robin fashion.
- We can apply the same idea to LBP, using updates of the form

$$m_i^{k+1} = f_i\left(\{m_j^{k+1}: j < i\}, \{m_j^k, j > i\}\right)$$

where the message for edge i is computed using new messages (iteration k + 1) from edges earlier in the ordering, and using old messages (iteration k) from edges later in the ordering.

- This raises the question of what order to update the messages in

Increasing the convergence rate: message scheduling

- One simple idea is to use a fixed or random order.
- A smarter approach is to:
  - pick a set of spanning trees
  - perform an up-down sweep on one tree at a time, keeping all the other messages fixed.
  - This is known as tree re-parameterization (Wainwright et al. 2001)
- We can do even better by using an adaptive ordering
  - The intuition is that we should focus our computational efforts on those variables that are most uncertain.
  - Elidan et al. 2006) proposed a technique known as residual belief propagation (RBP)
    - In RBP, messages are scheduled to be sent according to the norm of the difference from their previous value.
    - The residual of new message at iteration k

$$r(s,t,k) = \left\|\log m_{st} - \log m_{st}^k\right\|_\infty = \max_i \left|\log \frac{m_{st}(i)}{m_{st}^k(i)}\right|$$

Increasing the convergence rate: message scheduling

- We can store messages in a priority queue, and always send the one with highest residual.
- When a message is sent from s to t, all of the other messages that depend on m_{st} (i.e., messages of the form m_{tu} where $u \in nbr(t) \backslash s$) need to be recomputed along with the residuals, and added to the queue.
- In (Elidan et al. 2006), it is showed (experimentally) that this method converges more often, and much faster, than synchronous updating, asynchronous updating with a fixed order, and the TRP.
- A refinement of residual BP was presented in (Sutton and McCallum 2007)
  - an upper bound on the residual of a message is used.
  - This was observed to be about five times faster.

Accuracy of LBP

- For a a single loop graph, the max-product version of LBP finds the correct MAP estimate, if converged (Weiss 2000).

- For more general graphs, one can bound the error in the approximate marginals computed by LBP.

- For a Gaussian model, if the method converges, the means are exact, although the variances are not
  - Typically, the beliefs are over-confident.

Fast message computation for large state spaces

- The cost of computing each message in BP (whether in a tree or a loopy graph) is $O\left(K^f\right)$, where K is the number of states and f is the size of the largest factor.
- For pairwise Markov model, f=2.
- In many vision problems (e.g., image denoising),
  - K is quite large because it represents the discretization of some underlying continuous space.
  - So, $O(K^2)$ per message is quite expensive.
- Fortunately, for certain kinds of pairwise potential functions of the form $\psi_{st}(x_s, x_t) = \psi(x_s - x_t)$, one can compute sum-product message is $O(K \log K)$ time by using FFT.

Fast message computation for large state spaces

- The key insight is that message computation is just convolution:

$$M_{st}^{k}(x_t) = \sum_{x_s} \psi(x_s - x_t) h(x_s)$$

where $h(x_s) = \psi_s(x_s) \prod_{v \in nbr(s) \backslash \mathrm{t}} M_{vs}^{k-1}(x_s)$.

  - If the potential function $\psi(z)$ is a Gaussian-like potential, we can compute the convolution in $O(K)$ time by sequentially convolving with a small number of box filters (Felzenszwalb and Huttenlocher 2006).

- For the max-product case, a technique called the distance transform can be used to compute messages in $O(K)$ time.
  - However, this only works if $\psi(z) = \exp\big(-E(z)\big)$ where, $E(z)$ is : (a) quadratic, (b) truncated linear or Potts model (Felzenszwalb and Huttenlocher 2006)

Other speedup tricks for LBP: Multi-scale methods

- This method, also known as multi-grid technique, is specific to 2d lattice structures (commonly arise in computer vision).
- In the computer vision context, (Felzenszwalb and Huttenlocher 2006) suggested using the following heuristic to significantly speedup BP:
  - construct a coarse-to-fine grid
  - compute messages at the coarse level
  - Use this to initialize messages at the level below
- When we reach the bottom level, just a few iterations of standard BP are required since long-range communication has already been achieved via the initialization process

Exact inference as variational optimization problem

- The goal of variational inference is to find the distribution q that maximizes the energy functional:
  $L(q) = -\mathbb{KL}(q|p) + \log Z = \mathbb{E}_q[\log\,\tilde{p}(\mathbf{x})] + \mathbb{H}(q) \leq \log Z$

where $\tilde{p}(\mathbf{x}) = \mathrm{Z}p(\mathbf{x})$ is the un-normalized posterior.

- If we write $\log\,\tilde{p}(\mathbf{x}) = \boldsymbol{\theta}^T \phi(\mathbf{x})$ and we let $q = p$, then the exact energy functional becomes:

$$\max_{\boldsymbol{\mu} \in \mathbb{M}(G)} \boldsymbol{\theta}^{\boldsymbol{T}} \boldsymbol{\mu} + H(\boldsymbol{\mu})$$

where $\boldsymbol{\mu} = \mathbb{E}_p[\phi(\mathbf{x})]$ is a joint distribution over all state configurations $\mathbf{x}$.

- Since the KL divergence is zero when $p = q$, we know

$$\max_{\boldsymbol{\mu} \in \mathbb{M}(G)} \boldsymbol{\theta}^{\boldsymbol{T}} \boldsymbol{\mu} + H(\boldsymbol{\mu}) = \log Z\,(\boldsymbol{\theta})$$

- This is a way to cast exact inference as a variational optimization problem.

Mean field as variational optimization problem

- Let, F be an edge subgraph of the original graph G such that $\mathbb{I}(F) \subseteq \mathbb{I}$ be the subset of sufficient statistics associated with the cliques of F.
- Also assume Ω be the set of canonical parameters for the full model, and define the canonical parameter space for the submodel as follows

$$\Omega(F) \triangleq \{\boldsymbol{\theta} \in \Omega : \boldsymbol{\theta}_{st} = 0 \ \forall \alpha \in \mathbb{I} \backslash \mathbb{I}(F)\}$$

- In other words, we require that the natural parameters associated with the sufficient statistics α outside of our chosen class to be zero.
- For example, in the case of a fully factorized approximation, F_0, we remove all edges from the graph giving

$$\Omega(F_0) \triangleq \{\boldsymbol{\theta} \in \Omega : \boldsymbol{\theta}_{st} = 0 \ \forall (s,t) \in E\}$$

Mean field as variational optimization problem

- Next, we define the mean parameter space of the restricted model as follows
$$\mathbb{M}_F(G) \triangleq \{\boldsymbol{\mu} \in \mathbb{R}^d : \boldsymbol{\mu} = \mathbb{E}_\theta[\phi(\mathbf{x})] \text{ for some } \boldsymbol{\theta} \in \Omega(F)\}$$
- This is called an inner approximation to the marginal polytope, since $\mathbb{M}_F(G) \subseteq \mathbb{M}(G)$.
- $\mathbb{M}_F(G)$ is a non-convex polytope, which results in multiple local optima.
- We define the entropy of our approximation $\mathbb{H}\big(\boldsymbol{\mu}(F)\big)$ as the entropy of the distribution $\boldsymbol{\mu}$ defined on submodel F.
- Then we define the mean field energy functional optimization problem as
$$\max_{\boldsymbol{\mu} \in \mathbb{M}_F(G)} \boldsymbol{\theta}^T \boldsymbol{\mu} + \mathbb{H}(\boldsymbol{\mu}) \leq \log Z(\boldsymbol{\theta})$$

Mean field as variational optimization problem

- In the case of the fully factorized mean field approximation for pairwise UGMs, we can write this objective as:

$$\max_{\boldsymbol{\mu}\in\mathcal{P}^d} \sum_{s\in\mathcal{V}} \sum_{x_s} \theta_s(x_s)\mu_s(x_s) + \sum_{(s,t)\in\mathcal{E}} \sum_{x_s,x_t} \theta_{st}(x_s,x_t)\mu_s(x_s)\mu_t(x_t) + \sum_{s\in\mathcal{V}} \mathbb{H}(\boldsymbol{\mu}_s)$$

Where $\boldsymbol{\mu} \in \mathcal{P}$ and $\mathcal{P}$ is the probability simplex over $\mathcal{X}$.

- Mean field involves a concave objective being maximized over a non-convex set
- It is typically optimized using coordinate ascent, since it is easy to optimize a scalar concave function over $\mathcal{P}$ for each μ_s.

LBP as a variational optimization problem

- For considering all possible probability distributions which are Markov w.r.t. our model, we need to consider all vectors $\boldsymbol{\mu} \in \mathbb{M}(G)$.
- However, $\mathbb{M}(G)$ is exponentially large and it is usually infeasible to optimize over.
- A standard strategy in combinatorial optimization is to relax the constraints
- In this case, instead of requiring probability vector $\boldsymbol{\mu}$ to live in $\mathbb{M}(G)$, we consider a vector $\boldsymbol{\tau}$ that only satisfies the following local consistency constraints

$$\sum_{x_s} \tau_{\boldsymbol{s}}(x_{\boldsymbol{s}}) = 1$$

$$\sum_{x_t} \tau_{st}(x_s, x_t) = \tau_{\boldsymbol{s}}(x_{\boldsymbol{s}})$$

LBP as a variational optimization problem

- The first constraint in previous slide is called the normalization constraint whereas the second constraint is called the marginalization constraint.

- We then define the set

$$\mathbb{L}(G) \triangleq \{\tau \geq 0 \text{ with normalization constraint holding } \forall s$$

LBP as a variational optimization problem

- We call the terms $\tau_{st}, \tau_{\boldsymbol{s}}(x_{\boldsymbol{s}}) \in \mathbb{L}(G)$ pseudo marginals (since it may not correspond to marginals of any valid probability distribution).

- We claim that $\mathbb{M}(G) \subseteq \mathbb{L}(G)$ with equality if G is a tree.
  - To see this, first consider an element $\boldsymbol{\mu} \in \mathbb{M}(G)$.
  - Any such vector must satisfy the normalization and marginalization constraints, hence $\mathbb{M}(G) \subseteq \mathbb{L}(G)$.
  - Now suppose, T is a tree, and let $\boldsymbol{\mu} \in \mathbb{L}(T)$. By definition, this satisfies the normalization and marginalization constraints.
  - However, any tree can be represented in the form

$$p_{\mu}(\mathbf{x}) = \prod_{s \in \mathcal{V}} \mu_s(x_s) \prod_{s,t \in \mathcal{E}} \frac{\mu_{st}(x_s, x_t)}{\mu_s(x_s)\mu_t(x_t)}$$

  - Hence satisfying normalization and local consistency is enough to define a valid distribution for any tree. Hence, $\boldsymbol{\mu} \in \mathbb{M}(T)$.

LBP as a variational optimization problem

- The exact entropy of a tree structured distribution $\boldsymbol{\mu} \in \mathbb{M}(T)$ is

$$\mathbb{H}(\boldsymbol{\mu}) = \sum_{s \in \mathcal{V}} H_s(\mu_s) - \sum_{(s,t) \in \mathcal{E}} I_{st}(\mu_{st})$$

$$H_s(\mu_s) = -\sum_{x_s \in \mathcal{X}_s} \mu_s(x_s) \log \mu_s(x_s)$$

$$I_{st}(\mu_{st}) = \sum_{(x_s, x_t) \in \mathcal{X}_s, \mathcal{X}_t} \log \frac{\mu_{st}(x_s, x_t)}{\mu_s(x_s)\mu_t(x_t)}$$

- Note that we can rewrite the mutual information term in the form $I_{st}(\mu_{st}) = H_s(\mu_s) + H_t(\mu_t) - H_{st}(\mu_{st})$ and hence, have the following alternative expression

$$\mathbb{H}(\mu) = -\sum_{s \in \mathcal{V}} (\boxed{d_s} - 1) H_s(\mu_s) + \sum_{(s,t) \in \mathcal{E}} H_{st}(\mu_{st})$$

Degree of neighbor

LBP as a variational optimization problem

- The Bethe approximation to the entropy is simply the use

$$\mathbb{H}(\boldsymbol{\mu}) = \sum_{s\in\mathcal{V}} H_s(\mu_s) - \sum_{(s,t)\in\mathcal{E}} I_{st}(\mu_{st})$$

even when we don't have a tree.:

$$\mathbb{H}_{\text{Bethe}}(\boldsymbol{\mu}) = \sum_{s\in\mathcal{V}} H_s(\mu_s) - \sum_{(s,t)\in\mathcal{E}} I_{st}(\mu_{st})$$

- We define the Bethe free energy as

$$F_{\text{Bethe}}(\boldsymbol{\tau}) \triangleq -[\boldsymbol{\theta}^T\boldsymbol{\tau} + \mathbb{H}_{\text{Bethe}}(\boldsymbol{\tau})]$$

- We define the Bethe energy functional as the negative of the Bethe free energy.

LBP as a variational optimization problem: The LBP objective

- Combining the outer approximation $\mathbb{L}(G)$ with the Bethe approximation to the entropy, we get the following Bethe variational problem

$$\min_{\boldsymbol{\tau}\in\mathbb{L}(G)} F_{\mathrm{Bethe}}(\boldsymbol{\tau}) = \max_{\boldsymbol{\tau}\in\mathbb{L}(G)} \boldsymbol{\theta}^T\boldsymbol{\tau} + \mathbb{H}_{\mathrm{Bethe}}(\boldsymbol{\tau})$$

- The space we are optimizing over is a convex set, but the objective itself is not concave (since $\mathbb{H}_{\mathrm{Bethe}}(\boldsymbol{\tau})$ is not concave).
- Thus there can be multiple local optima of the BVP.
- The value obtained by the BVP is an approximation to $\log Z(\boldsymbol{\theta})$
- In the case of trees, the approximation is exact.
- For models with attractive potentials, the approximation turns out to be an upper bound (Sudderth et al. 2008).

LBP as a variational optimization problem

- Any fixed point of the LBP algorithm defines a stationary point of the above constrained objective

- We define the normalization constraint as $C_{ss}(\boldsymbol{\tau}) \triangleq 1 - \sum_{x_s} \tau_s(x_s)$ and the marginalization constraint as $C_{ts}(\boldsymbol{\tau}) \triangleq \tau_s(x_s) - \sum_{x_t} \tau_{st}(x_s, x_t)$ for each edge $t \to s$.

- We can now write the Lagrangian as

$$\mathcal{L}(\boldsymbol{\tau}, \boldsymbol{\lambda}; \boldsymbol{\theta})$$

$$\triangleq \boldsymbol{\theta}^T \boldsymbol{\tau} + \mathbb{H}_{\text{Bethe}}(\boldsymbol{\tau}) + \sum_s \lambda_{ss} C_{ss}(\boldsymbol{\tau})$$

$$+ \sum_{s,t} \left[\sum_{x_s} \lambda_{ts}(x_s) C_{ts}(x_s; \boldsymbol{\tau}) + \left[\sum_{x_s} \lambda_{st}(x_t) C_{st}(x_t; \boldsymbol{\tau}) \right] \right]$$

LBP as a variational optimization problem

- Setting $\nabla_\tau \mathcal{L} = 0$, we obtain

$$\log \tau_s(x_s) = \lambda_{ss} + \theta_s(x_s) + \sum_{t \in nbr(s)} \lambda_{ts}(x_s)$$

$$\log \frac{\tau_{st}(x_s, x_t)}{\tilde{\tau}_s(x_s)\tilde{\tau}_t(x_t)}$$

$$= \lambda_{ss} + \lambda_{tt} + \theta_{st}(x_s, x_t) + \theta_s(x_s) + \theta_t(x_t) + \sum_{u \in nbr(s) \backslash t} \lambda_{us}(x_s)$$

$$+ \sum_{u \in nbr(t) \backslash s} \lambda_{ut}(x_t)$$

- To make the connection to message passing, define $M_{ts}(x_s) = \exp\left(\lambda_{ts}(x_s)\right)$. We can rewrite above equations

$$\tau_s(x_s) \propto \exp\left(\theta_s(x_s)\right) \prod_{t \in nbr(s)} M_{ts}(x_s)$$

LBP as a variational optimization problem

$$\tau_{st}(x_s, x_t) \\ \propto \exp\left(\theta_{st}(x_s, x_t) + \theta_s(x_s) + \theta_t(x_t)\right) \\ \times \prod_{u \in nbr(s)\backslash t} M_{us}(x_s) \prod_{u \in nbr(t)\backslash s} M_{ut}(x_t)$$

where the λ terms are absorbed into the constant of proportionality.

- We see that this is equivalent to the usual expression for the node and edge marginals in LBP.
- To derive an equation for the messages in terms of other messages, we enforce the marginalization condition $\tau_s(x_s) = \sum_{x_t} \tau_{st}(x_s, x_t)$. This yields

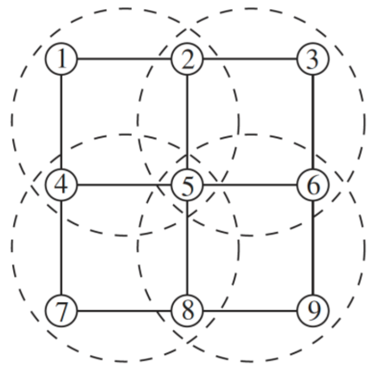
$$M_{ts}(x_s) \propto \sum_{x_t} \left[\exp\{\theta_{st}(x_s, x_t) + \theta_t(x_t)\} \prod_{u \in nbr(t)\backslash s} M_{ut}(x_t) \right]$$

Loopy BP vs mean field

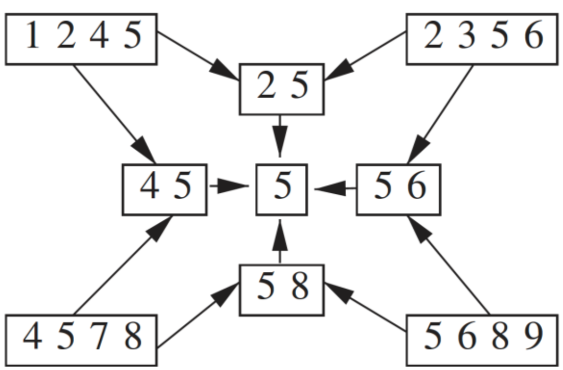
- The advantages of LBP are:
  - LBP is exact for trees whereas MF is not, suggesting LBP is more accurate.
  - LBP optimizes over node and edge marginals, whereas MF only optimizes over node marginals, again suggesting LBP will be more accurate
  - In the case that the true edge marginals factorize, so $\boldsymbol{\mu}_{st} = \boldsymbol{\mu}_s \boldsymbol{\mu}_t$, the free energy approximations will be the same in both cases.
  - What is less obvious, but which nevertheless seems to be true, is that the MF objective has many more local optima
  - So optimizing the MF objective seems to be harder
- The advantages of MF are:
  - It gives a lower bound on the partition function.
    - This is useful when using it as a subroutine inside a learning algorithm
  - MF is easier to extend to other distributions besides discrete and Gaussian.

Extension of BP: Generalized BP

- We can improve the accuracy of loopy BP by clustering together nodes that form a tight loop.
  - This is known as the cluster variational method
  - The result is a hyper-graph, which is a graph where there are hyper-edges between sets of vertices instead of between single vertices
    - A junction tree is a kind of hyper-graph
  - We can represent hyper-graph using a poset (partially ordered set) diagram, where each node represents a hyper-edge, and there is an arrow $e_1 \to e_2$ if $e_1 \subset e_2$.



Kikuchi clusters superimposed on 3×3 lattice graph



Corresponding hypergraph

Extension of BP: Generalized BP

- Let t be the size of the largest hyper-edge in the hyper-graph

- If we allow t to be as large as the tree-width of the graph, then we can represent the hyper-graph as a tree, and the method will be exact, just as LBP is exact on regular trees (with tree-width 1).

- In this way, we can define a continuum of approximations, from LBP all the way to exact inference

Extension of BP: Generalized BP

- We define $\mathbb{L}_t(G)$ to be the set of all pseudo-marginals such that normalization and marginalization constraints hold on a hyper-graph whose largest hyper-edge is of size
- Furthermore, we approximate the entropy as

$$\mathbb{H}_{\text{Kikuchi}}(\boldsymbol{\tau}) \triangleq \sum_{g \in \mathcal{E}} c(g) H_g\left(\tau_g\right)$$

where $H_g\left(\tau_g\right)$ is the entropy of the joint (pseudo) distribution on the vertices in set g.

- These are related to Mobious numbers in set theory
- Finally, we can define the Kikuchi free energy3

$$F_{\text{Kikuchi}}(\boldsymbol{\tau}) \triangleq -[\boldsymbol{\theta}^T \boldsymbol{\tau} + \mathbb{H}_{\text{Kikuchi}}(\boldsymbol{\tau})]$$

- The variational problem becomes

$$\min_{\boldsymbol{\tau} \in \mathbb{L}_t(G)} F_{\text{Kikuchi}}(\boldsymbol{\tau}) = \max_{\boldsymbol{\tau} \in \mathbb{L}_t(G)} \boldsymbol{\theta}^T \boldsymbol{\tau} + \mathbb{H}_{\text{Kikuchi}}(\boldsymbol{\tau})$$

Extension of BP: Generalized BP

- Just as with the Bethe free energy, this is not a concave objective

- There are several possible algorithms for finding a local optimum of this objective.

- The method gives more accurate results than LBP, but at increased computational cost (because of the need to handle clusters of node.

- This cost, plus the complexity of the approach, have precluded it from widespread use

Extension of BP: Convex belief propagation

- The mean field energy functional is concave, but it is maximized over a non-convex inner approximation to the marginal polytope.
- The Bethe and Kikuchi energy functional are not concave, but they are maximized over a convex outer approximation to the marginal polytope.
- Consequently, for both MF and LBP:
  - the optimization problem has multiple optima
  - the methods are sensitive to the initial conditions
- In Convex BP, we try to come up with an approximation which involves a concave objective being maximized over a convex set

Extension of BP: Convex belief propagation

- Convex belief propagation (CBP) involves working with a set of tractable sub-models $\mathcal{F}$, such as trees or planar graphs.
- For each model, $F \subset G$, the entropy is higher, $\mathbb{H}\big(\mu(F)\big) \geq \mathbb{H}\big(\mu(G)\big)$, since F has fewer constraints.
- Consequently, any convex combination of such subgraphs will have higher entropy, too

$$\mathbb{H}\big(\mu(G)\big) \leq \sum_{F \in \mathcal{F}} \rho(F)\mathbb{H}\big(\mu(F)\big) \triangleq \mathbb{H}(\boldsymbol{\mu}, \rho)$$

where $\rho(F) \geq 0$ and $\sum_F \rho(F) = 1$.

- Furthermore, $\mathbb{H}(\boldsymbol{\mu}, \rho)$ is a concave function of $\boldsymbol{\mu}$.

Extension of BP: Convex belief propagation

- We now define the convex free energy as
$$F_{\text{Convex}}(\boldsymbol{\mu}, \rho) \triangleq -[\boldsymbol{\mu}^T \boldsymbol{\theta} + \mathbb{H}(\boldsymbol{\mu}, \rho)]$$

- We define the concave energy functional as the negative of the convex free energy

- Having defined an upper bound on the entropy, we now consider a convex outerbound on the marginal polytope of mean parameters

Extension of BP: Convex belief propagation

- We want to ensure we can evaluate the entropy of any vector $\boldsymbol{\tau}$ in this set, so we restrict it so that the projection of $\boldsymbol{\tau}$ onto the subgraph G lives in the projection of $\mathbb{M}$ onto F:

$$\mathbb{L}(G;\mathcal{F}) \triangleq \{\boldsymbol{\tau} \in \mathbb{R}^d : \boldsymbol{\tau}(F) \in \mathbb{M}(F) \forall F \in \mathcal{F}\}$$

- This is a convex set since each $\mathbb{M}(F)$ is a projection of a convex set.

- We define our problem as:

$$\min_{\tau \in \mathbb{L}(G;\mathcal{F})} F_{\text{Convex}}(\boldsymbol{\tau}, \rho) = \max_{\tau \in \mathbb{L}(G;\mathcal{F})} \boldsymbol{\tau}^T \boldsymbol{\theta} + \mathbb{H}(\boldsymbol{\tau}, \rho)$$

- This is a concave objective being maximized over a convex set, and hence has a unique maximum.

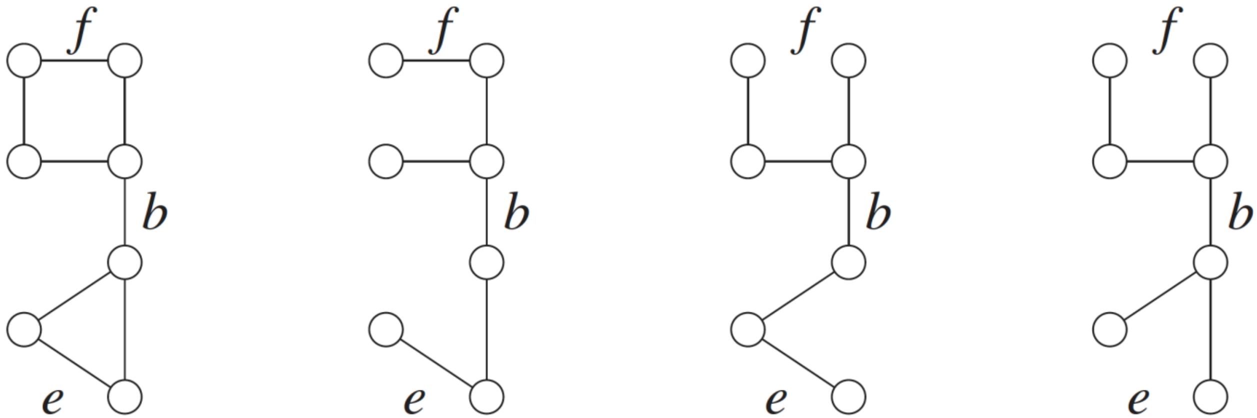
Extension of BP: Tree reweighted propagation

- Consider the specific case where $\mathcal{F}$ is all spanning trees of a graph.
- To compute the upper bound, obtained by averaging over all trees, note that the terms $\sum_F \rho(F) H(\mu(F)_s)$ for single nodes will just be H_s, since node s appears in every tree and $\sum_F \rho(F) = 1$.
- But the mutual information term I_{st} receives weight $\rho_{st} = \mathbb{E}_\rho[\mathbb{I}(s,t) \in \mathcal{E}(T)]$, known as the edge appearance probability.
- Hence we have the following upper bound on the entropy

$$\mathbb{H}(\boldsymbol{\mu}) \leq \sum_{s \in \mathcal{V}} H_s(\mu_s) - \sum_{(s,t) \in \mathcal{E}} \rho_{st} I_{st}(\mu_{st})$$

- The edge appearance probabilities live in a space called the spanning tree polytope

Extension of BP: Tree reweighted propagation



- Suppose each tree has equal weight under ρ.
  - The edge f occurs in 1 of the 3 and so, $\rho_f = \frac{1}{3}$
  - The edge e occurs in 2 of the 3 and so, $\rho_e = \frac{2}{3}$
  - The edge b occurs in all 3 and so, $\rho_b = 1$ and so on.

- Ideally we can find a distribution ρ or equivalently edge probability that the above bound as tight as possible

Extension of BP: Tree reweighted propagation

- We require $\boldsymbol{\mu}(T) \in \mathbb{M}(T)$ for each tree T, which means enforcing normalization and local consistency.
- Since we have to do this for every tree, we are enforcing normalization and local consistency on every edge. Hence, $\mathbb{L}(G; \mathcal{F}) = \mathbb{L}(G)$.
- So our final optimization problem is

$$\max_{\boldsymbol{\tau} \in \mathbb{L}(G)} \left\{ \tau^T \boldsymbol{\theta} + \sum_{s \in \mathcal{V}} H_s(\tau_s) - \sum_{(s,\boldsymbol{t}) \in \mathcal{E}} \rho_{st} I_{st}(\tau_{st}) \right\}$$

which is the same as the LBP objective except for the crucial ρ_{st} weights.

- As long as $\rho_{st} > 0, \forall s, t$, this problem is strictly concave with a unique maximum
- To find the global optima, we can use tree reweighted belief propagation (TRBP).

Extension of BP: Tree reweighted propagation

- In TRBP, the message from t to s is now a function of all messages sent from other neighbors and messages sent from s to t.

$$M_{ts} \propto \sum_{x_t} \exp\left(\frac{1}{\rho_{st}}\theta_{st}(x_s, x_t) + \theta_t(x_t\ )\right)\frac{\prod_{v\in nbr(t)\backslash \mathrm{s}}[M_{vt}(x_t)]^{\rho_{vt}}}{[M_{st}(x_t)]^{1-\rho_{ts}}}$$

- At convergence, the node and edge pseudo marginals are given by

$$\tau_s(x_s) \propto \exp(\theta_s(x_s\ )) \prod_{v \propto s} [M_{vs}(x_s)]^{\rho_{vs}}$$

$$\tau_{st}(x_s, x_t) \propto \varphi_{st}(x_s, x_t)\frac{\prod_{v\in nbr(s)\backslash \mathrm{t}}[M_{vs}(x_s)]^{\rho_{vs}}}{[M_{ts}(x_s)]^{1-\rho_{st}}}\frac{\prod_{v\in nbr(t)\backslash \mathrm{s}}[M_{vt}(x_t)]^{\rho_{vt}}}{[M_{st}(x_t)]^{1-\rho_{ts}}}$$

where

Extension of BP: Tree reweighted propagation

$$\varphi_{st}(x_s, x_t) \triangleq \exp\left(\frac{1}{\rho_{st}}\theta_{st}(x_s, x_t) + \theta_s(x_s) + \theta_t(x_t)\right)$$

- If $\rho_{st} = 1\ \forall (s,t) \in \mathcal{E}$, the algorithm reduces to the standard LBP algorithm.
- In general, this message passing scheme is not guaranteed to converge to the unique global optimum.
- Double-loop methods can guarantee to converge. However in practice, using damped version is often sufficient.
- Convex version of the Kikuchi free energy can be devised, which one can optimize with a modified version of generalized belief propagation.

Extension of BP: Tree reweighted propagation

- TRBP entropy approximation is an upper bound on the true entropy and hence, an upper bound on $\log Z$.

- Using $I_{st} = H_s + H_t - H_{st}$, we can rewrite the upper bound as

$$\log \hat{Z}(\boldsymbol{\theta}) \triangleq \boldsymbol{\tau}^T \boldsymbol{\theta} + \sum_{st} \rho_{st} H_{st}(\tau_{st}) + \sum_{s} c_s H_s(\tau_s) \leq \log Z(\boldsymbol{\theta})$$

where

$$c_s \triangleq 1 - \sum_{t} \rho_{st}$$