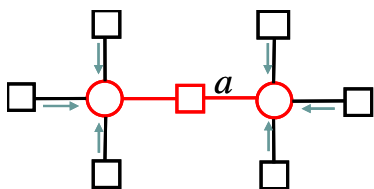
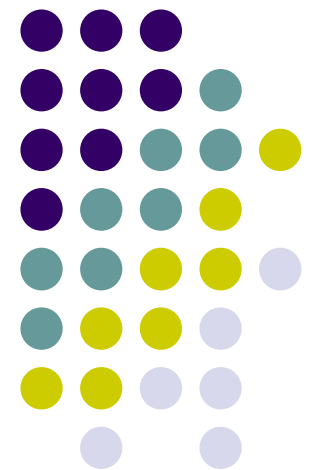




Probabilistic Graphical Models

Variational Inference: Loopy Belief Propagation



Eric Xing

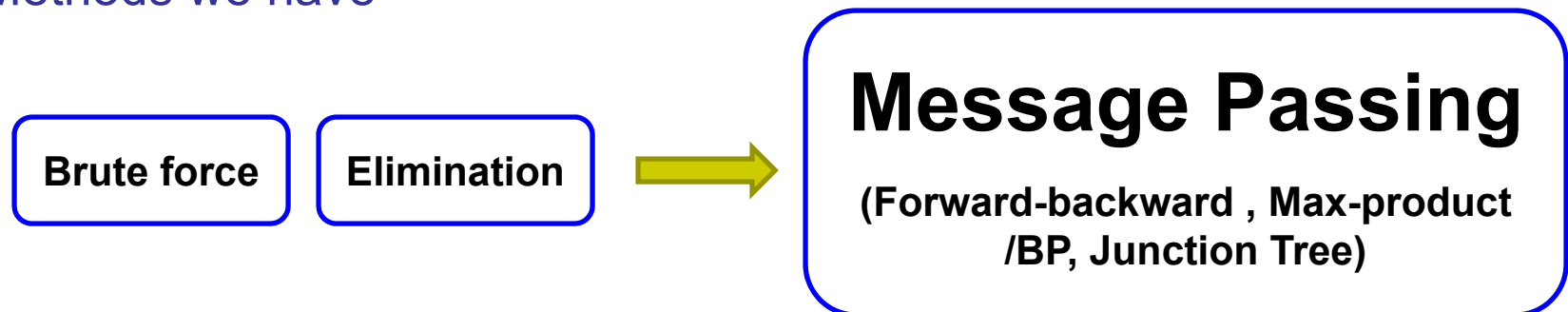
Lecture 12, February 22, 2016

Reading: See class website



Inference Problems

- Compute the likelihood of observed data
- Compute the marginal distribution $p(x_A)$ over a particular subset of nodes $A \subset V$
- Compute the conditional distribution $p(x_A|x_B)$ for disjoint subsets A and B
- Compute a mode of the density $\hat{x} = \arg \max_{x \in \mathcal{X}^m} p(x)$
- Methods we have



Individual computations independent

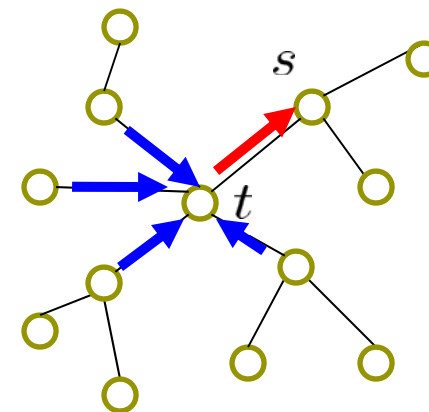
Sharing intermediate terms

Sum-Product Revisited



- Tree-structured GMs

$$p(x_1, \dots, x_m) = \frac{1}{Z} \prod_{s \in V} \psi_s(x_s) \prod_{(s,t) \in E} \psi_{st}(x_s, x_t)$$



- Message Passing on Trees:

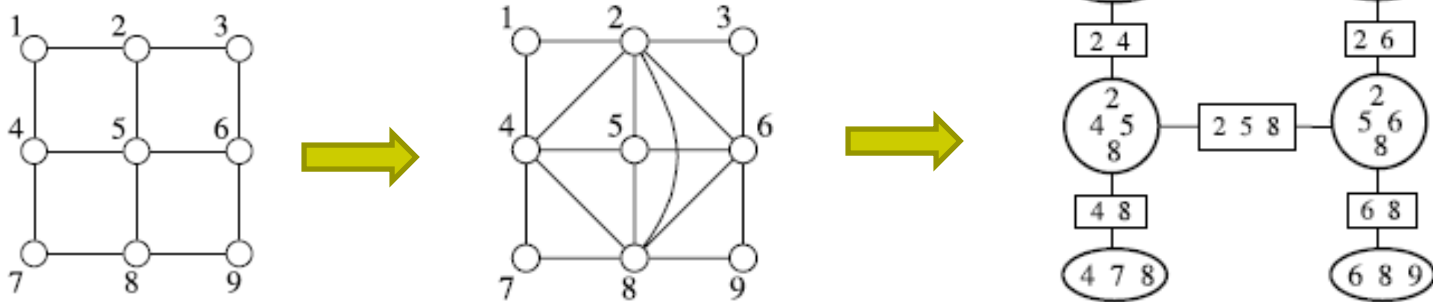
$$M_{t \rightarrow s}(x_s) \leftarrow \kappa \sum_{x'_t} \left\{ \psi_{st}(x_s, x'_t) \psi_t(x'_t) \prod_{u \in N(t) \setminus s} M_{u \rightarrow t}(x'_t) \right\}$$

- On trees, converge to a unique fixed point after a finite number of iterations



Junction Tree Revisited

- General Algorithm on Graphs with Cycles

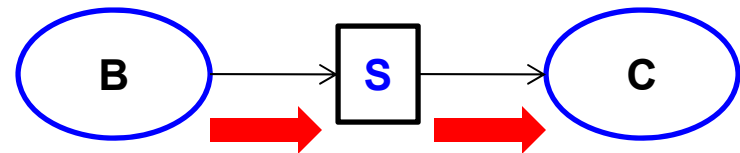


- Steps: **=> Triangularization** **=> Construct JTs**

=> Message Passing on Clique Trees

$$\tilde{\phi}_S(x_S) \leftarrow \sum_{x_{B \setminus S}} \phi_B(x_B)$$

$$\phi_C(x_C) \leftarrow \frac{\tilde{\phi}_S(x_S)}{\phi_S(x_S)} \phi_C(x_C)$$





Local Consistency

- Given a set of functions $\{\tau_C, C \in \mathcal{C}\}$ and $\{\tau_S, S \in \mathcal{S}\}$ associated with the cliques and separator sets
- They are locally consistent if:

$$\sum_{x'_S} \tau_S(x'_S) = 1, \forall S \in \mathcal{S}$$

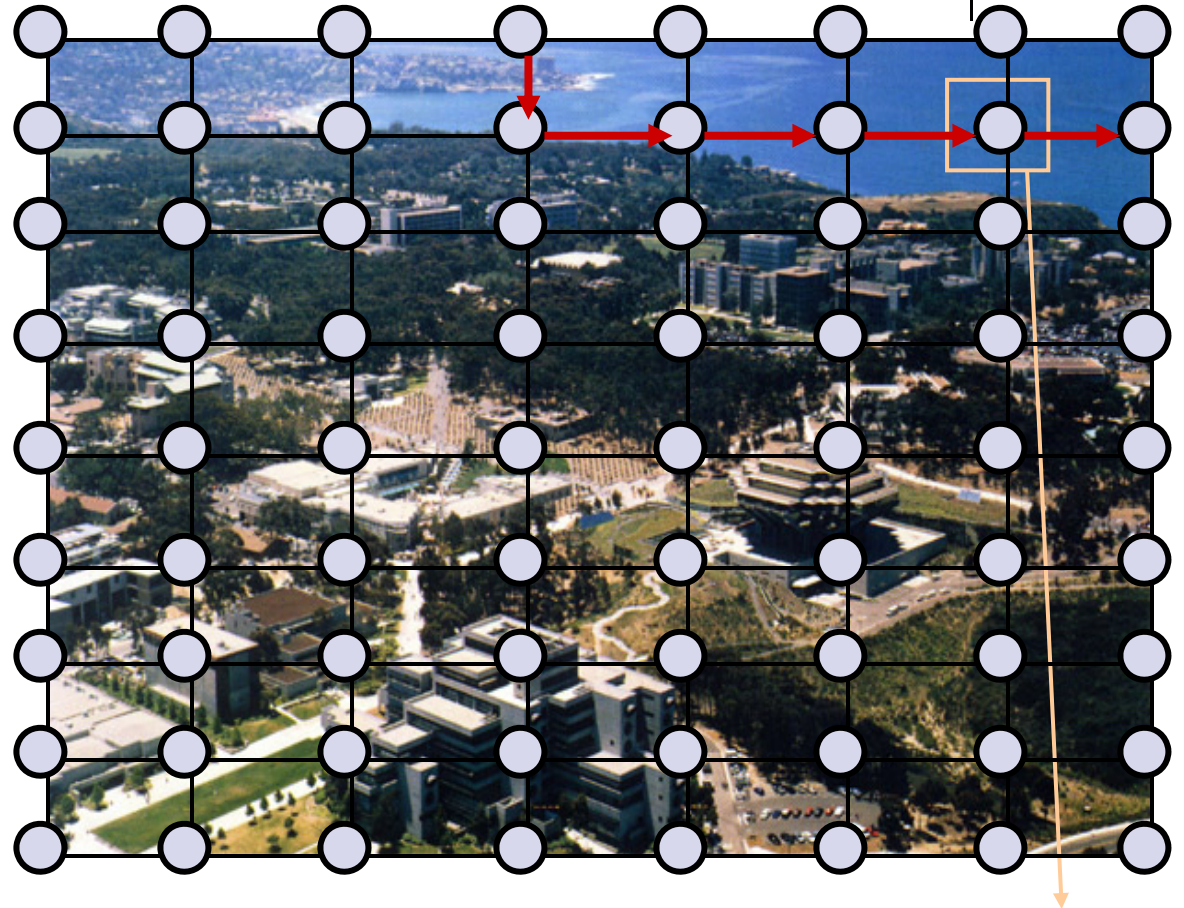
$$\sum_{x'_C | x'_S = x_S} \tau_C(x'_C) = \tau_S(x_S), \forall C \in \mathcal{C}, S \subset C$$

- For junction trees, local consistency is equivalent to global consistency!



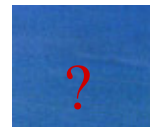
An Ising model on 2-D image

- Nodes encode hidden information (patch-identity).
- They receive local information from the image (brightness, color).
- Information is propagated through the graph over its edges.
- Edges encode 'compatibility' between nodes.



© Eric Xing @ CMU, 2005-2016

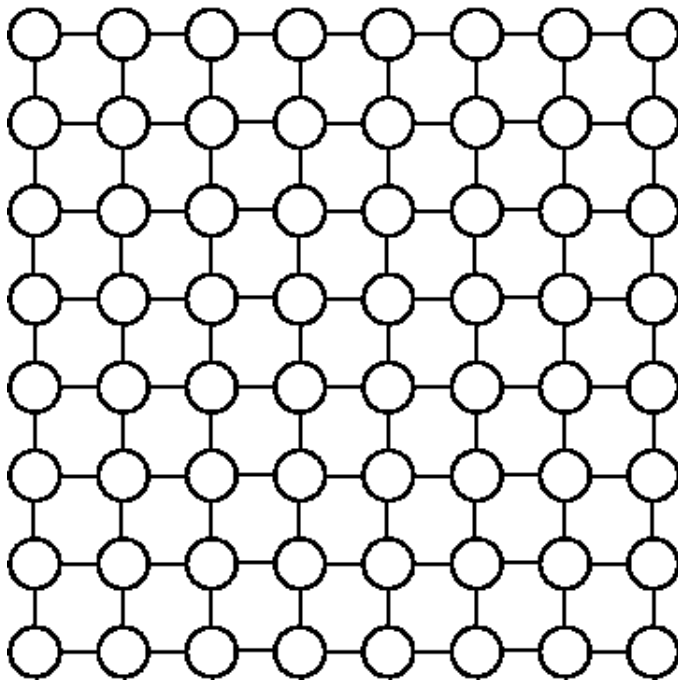
air or water ?





Why Approximate Inference?

- Why can't we just run junction tree on this graph?



$$p(X) = \frac{1}{Z} \exp \left\{ \sum_{i < j} \theta_{ij} X_i X_j + \sum_i \theta_{i0} X_i \right\}$$

- If NxN grid, tree width at least N
- N can be a huge number (~1000s of pixels)
 - If N ~ O(1000), we have a clique with 2^{100} entries



Approaches to inference

- Exact inference algorithms
 - The elimination algorithm
 - Message-passing algorithm (sum-product, belief propagation)
 - The junction tree algorithms

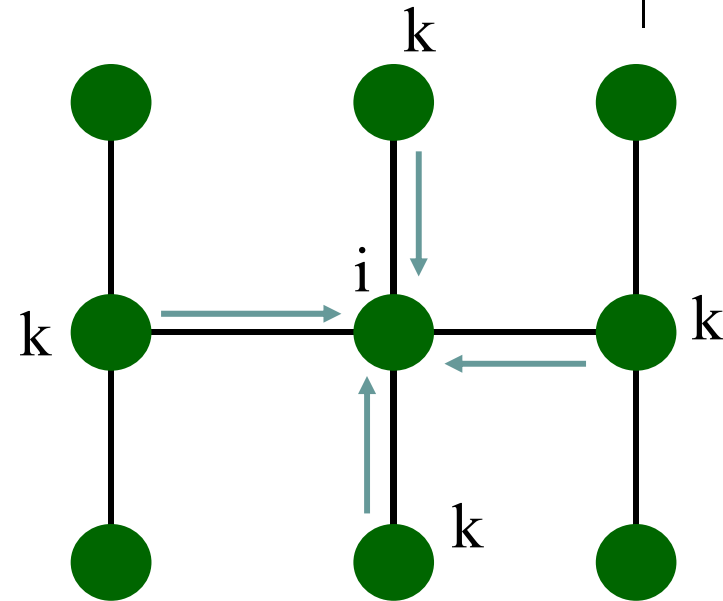
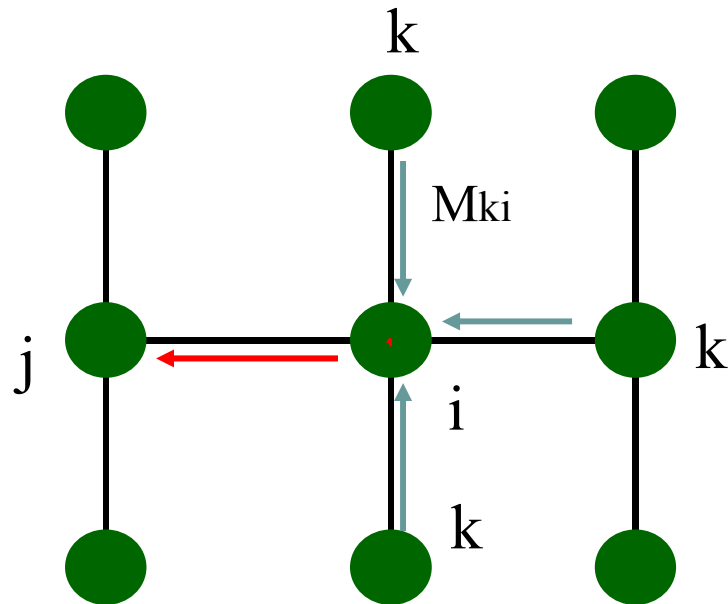
- Approximate inference techniques
 - Variational algorithms
 - Loopy belief propagation
 - Mean field approximation
 - Stochastic simulation / sampling methods
 - Markov chain Monte Carlo methods



Loopy Belief Propogation



Recap: Belief Propagation



- BP Message-update Rules

$$M_{i \rightarrow j}(x_j) \propto \sum_{x_i} \underbrace{\psi_{ij}(x_i, x_j)}_{\text{Compatibilities (interactions)}} \underbrace{\psi_i(x_i)}_{\text{external evidence}} \prod_k M_{k \rightarrow i}(x_i)$$

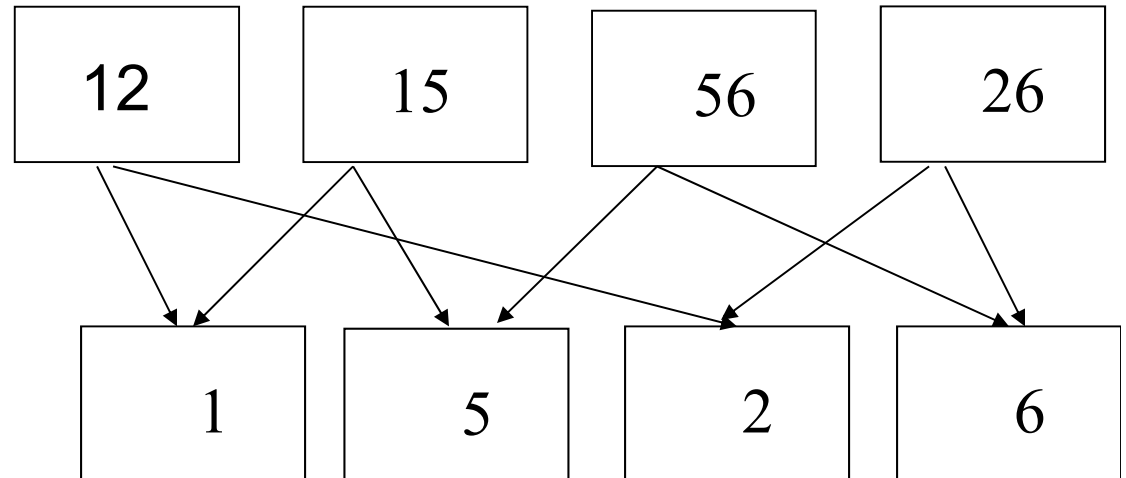
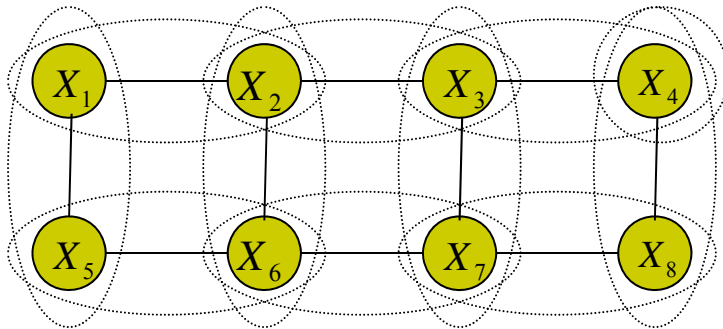
$$b_i(x_i) \propto \psi_i(x_i) \prod_k M_k(x_k)$$

- BP on trees always converges to exact marginals (cf. Junction tree algorithm)



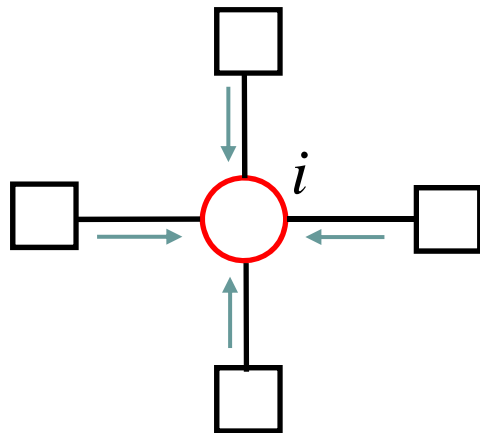
Region graphs (Factor Graph)

- It will be useful to look explicitly at the messages being passed
 - Messages from variable to factors
 - Messages from factors to variables
- Let us represent this graphically





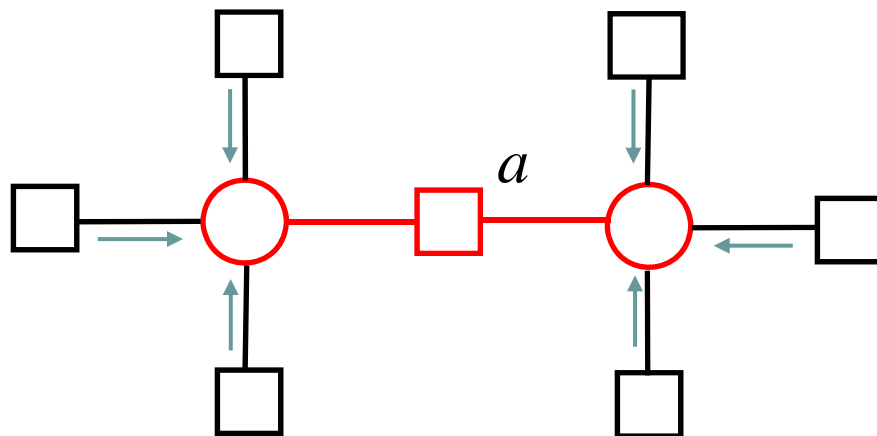
Beliefs and messages in FG



$$b_i(x_i) \propto f_i(x_i) \prod_{a \in N(i)} m_{a \rightarrow i}(x_i)$$

↑
“beliefs”

↑
“messages”



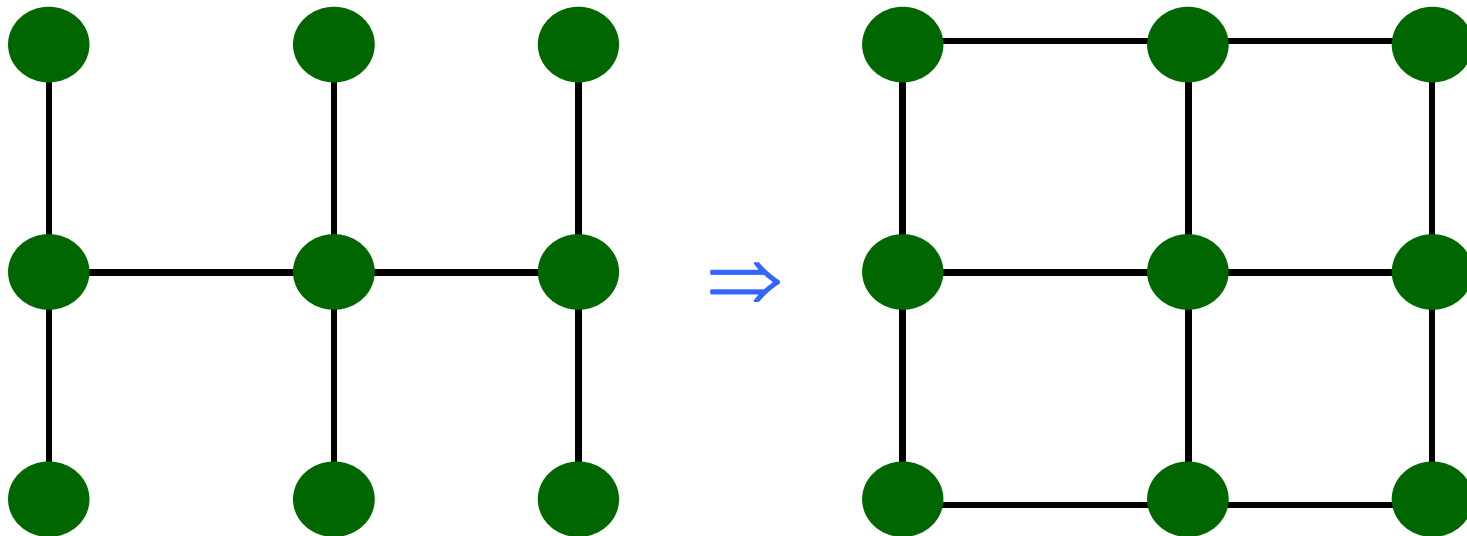
$$m_{i \rightarrow a}(x_i) = \prod_{c \in N(i) \setminus a} m_{c \rightarrow i}(x_i)$$

$$b_a(X_a) \propto f_a(X_a) \prod_{i \in N(a)} m_{i \rightarrow a}(x_i)$$

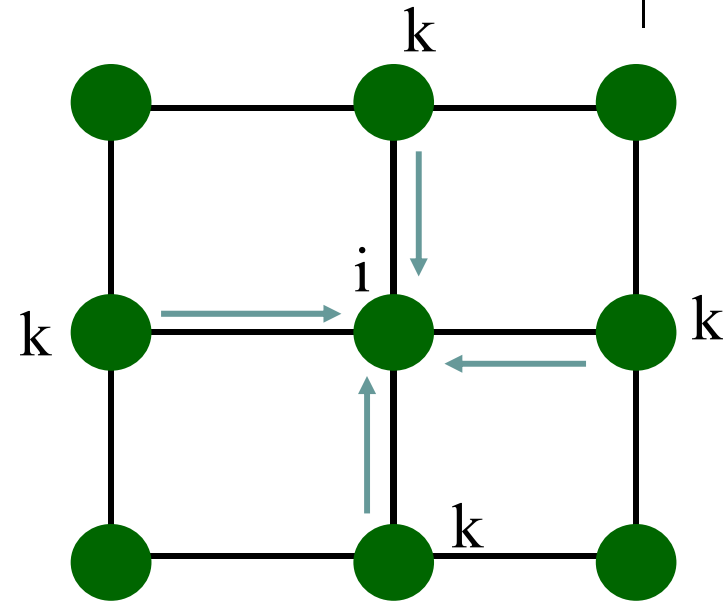
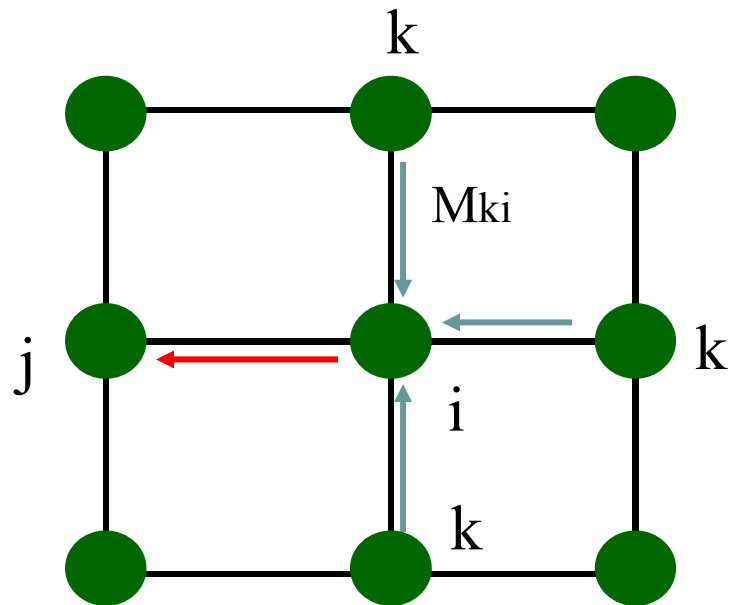
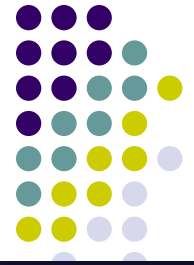
$$m_{a \rightarrow i}(x_i) = \sum_{X_a \setminus x_i} f_a(X_a) \prod_{j \in N(a) \setminus i} m_{j \rightarrow a}(x_j)$$



What if the graph is loopy?



Belief Propagation on loopy graphs



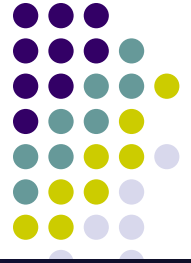
- BP Message-update Rules

$$M_{i \rightarrow j}(x_j) \propto \sum_{x_i} \psi_{ij}(x_i, x_j) \psi_i(x_i) \prod_k M_{k \rightarrow i}(x_i)$$

↑ Compatibilities (interactions)
 ↑ external evidence

$$b_i(x_i) \propto \psi_i(x_i) \prod_k M_k(x_k)$$

- May not converge or converge to a wrong solution



Loopy Belief Propagation

- A fixed point iteration procedure that tries to minimize F_{bethe}
- Start with random initialization of messages and beliefs
- While not converged do

$$b_i(x_i) \propto \prod_{a \in N(i)} m_{a \rightarrow i}(x_i) \quad b_a(X_a) \propto f_a(X_a) \prod_{i \in N(a)} m_{i \rightarrow a}(x_i)$$

$$m_{i \rightarrow a}^{\text{new}}(x_i) = \prod_{c \in N(i) \setminus a} m_{c \rightarrow i}(x_i) \quad m_{a \rightarrow i}^{\text{new}}(x_i) = \sum_{X_a \setminus x_i} f_a(X_a) \prod_{j \in N(a) \setminus i} m_{j \rightarrow a}(x_j)$$

- At convergence, stationarity properties are guaranteed
- However, not guaranteed to converge!



Loopy Belief Propagation

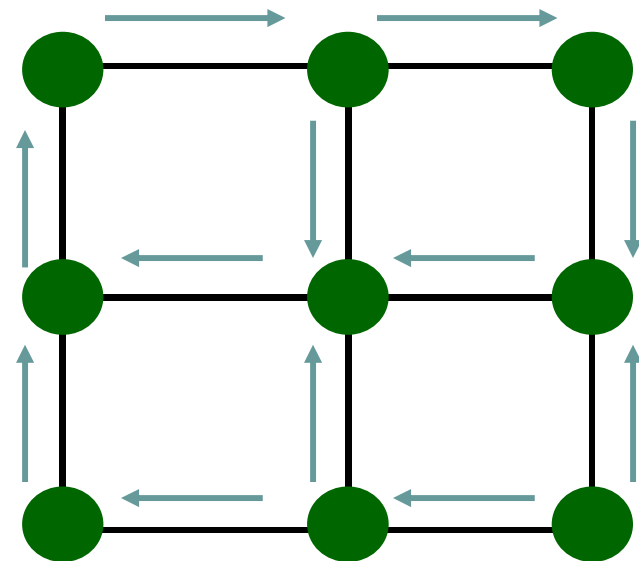
- If BP is used on graphs with loops, messages may circulate indefinitely
- But let's run it anyway and hope for the best ... 😊
- Empirically, a good approximation is still achievable
 - Stop after fixed # of iterations
 - Stop when no significant change in beliefs
 - If solution is not oscillatory but converges, it usually is a good approximation

[Loopy-belief Propagation for Approximate Inference: An Empirical Study](#)
Kevin Murphy, Yair Weiss, and Michael Jordan.
UAI '99 (Uncertainty in AI).]



So what is going on?

- Is it a dirty hack that you bet your luck?





Approximate Inference

- Let us call the actual distribution P

$$P(X) = 1/Z \prod_{f_a \in F} f_a(X_a)$$

- We wish to find a distribution Q such that Q is a “good” approximation to P
- Recall the definition of KL-divergence

$$KL(Q_1 \parallel Q_2) = \sum_X Q_1(X) \log\left(\frac{Q_1(X)}{Q_2(X)}\right)$$

- $KL(Q_1 \parallel Q_2) \geq 0$
- $KL(Q_1 \parallel Q_2) = 0$ iff $Q_1 = Q_2$
- We can therefore use KL as a scoring function to decide a good Q
- But, $KL(Q_1 \parallel Q_2) \neq KL(Q_2 \parallel Q_1)$

© Eric Xing @ CMU, 2005-2016



Which KL?

- Computing $KL(P||Q)$ requires inference!
- But $KL(Q||P)$ can be computed without performing inference on P

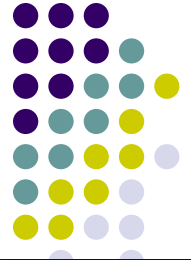
$$\begin{aligned}KL(Q \parallel P) &= \sum_X Q(X) \log\left(\frac{Q(X)}{P(X)}\right) \\&= \sum_X Q(X) \log Q(X) - \sum_X Q(X) \log P(X) \\&= -H_Q(X) - E_Q \log P(X)\end{aligned}$$

- Using $P(X) = 1/Z \prod_{f_a \in F} f_a(X_a)$

$$\begin{aligned}KL(Q \parallel P) &= -H_Q(X) - E_Q \log\left(1/Z \prod_{f_a \in F} f_a(X_a)\right) \\&= -H_Q(X) - \log 1/Z - \sum_{f_a \in F} E_Q \log f_a(X_a)\end{aligned}$$

© Eric Xing @ CMU, 2005-2016

Optimization function



$$KL(Q \parallel P) = \underbrace{-H_Q(X) - \sum_{f_a \in F} E_Q \log f_a(X_a)}_{F(P, Q)} + \log Z$$

- We will call $F(P, Q)$ the “Free energy” *
- $F(P, P) = ?$
- $F(P, Q) \geq F(P, P)$



The Energy Functional

- Let us look at the functional

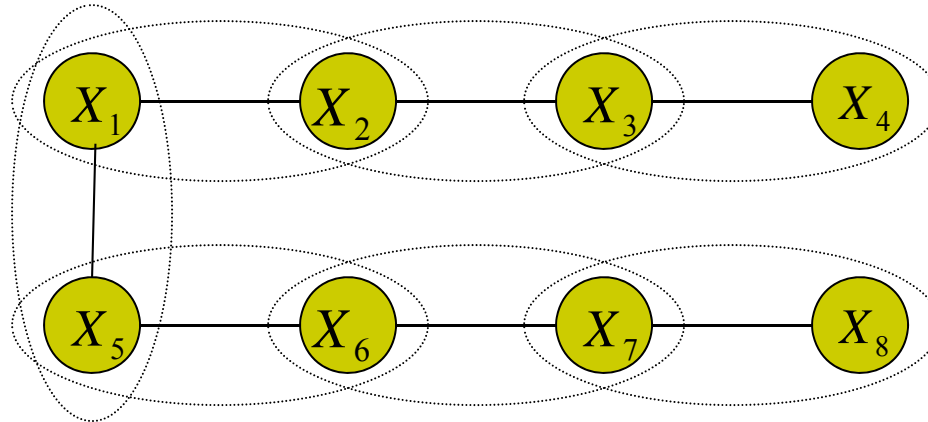
$$F(P, Q) = -H_Q(X) - \sum_{f_a \in F} E_Q \log f_a(X_a)$$

- $\sum_{f_a \in F} E_Q \log f_a(X_a)$ can be computed if we have marginals over each f_a
- $H_Q = -\sum_X Q(X) \log Q(X)$ is harder! Requires summation over all possible values
- Computing F , is therefore hard in general.
- Approach 1: Approximate $F(P, Q)$ with easy to compute $\hat{F}(P, Q)$



Tree Energy Functionals

- Consider a tree-structured distribution



- The probability can be written as: $b(\mathbf{x}) = \prod_a b_a(\mathbf{x}_a) \prod_i b_i(x_i)^{1-d_i}$

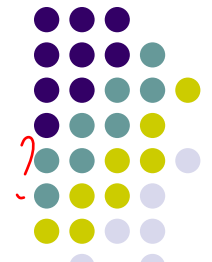
- $$H_{tree} = -\sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \ln b_a(\mathbf{x}_a) + \sum_i (d_i - 1) \sum_{x_i} b_i(x_i) \ln b_i(x_i)$$

- $$F_{Tree} = \sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \ln \frac{b_a(\mathbf{x}_a)}{f_a(\mathbf{x}_a)} + \sum_i (1 - d_i) \sum_{x_i} b_i(x_i) \ln b_i(x_i)$$

$$= F_{12} + F_{23} + \dots + F_{67} + F_{78} - F_1 - F_5 - F_2 - F_6 - F_3 - F_7$$

- involves summation over edges and vertices and is therefore easy to compute

Bethe Approximation to Gibbs Free Energy

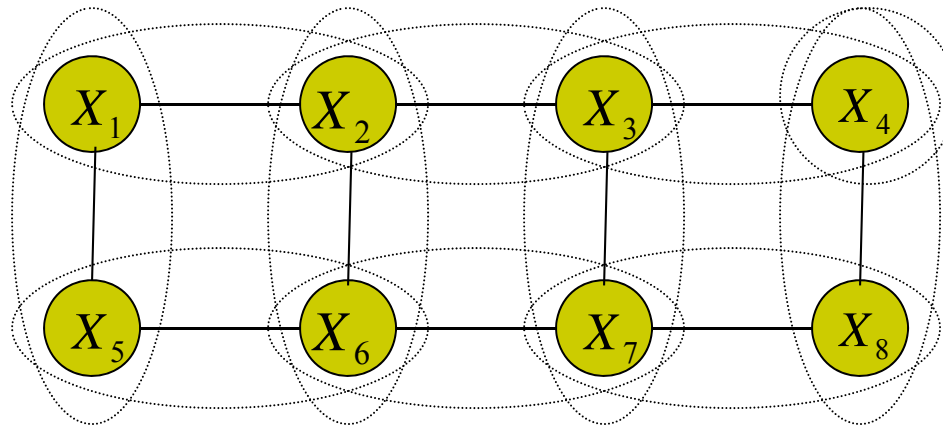


- For a general graph, choose $\hat{F}(P, Q) = F_{Bethe}$

$$H_{Bethe} = -\sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \ln b_a(\mathbf{x}_a) + \sum_i (d_i - 1) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \ln b_i(\mathbf{x}_i)$$

$$F_{Bethe} = \sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \ln \frac{b_a(\mathbf{x}_a)}{f_a(\mathbf{x}_a)} + \sum_i (1 - d_i) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \ln b_i(\mathbf{x}_i) = -\langle f_a(\mathbf{x}_a) \rangle - H_{Bethe}$$

- Called “Bethe approximation” after the physicist Hans Bethe



$$F_{Bethe} = F_{12} + F_{23} + \dots + F_{67} + F_{78} - F_1 - F_5 - 2F_2 - 2F_6 \dots - F_8$$

- Equal to the exact Gibbs free energy when the factor graph is a tree
- In general, H_{Bethe} is **not** the same as the H of a tree

© Eric Xing @ CMU, 2005-2016

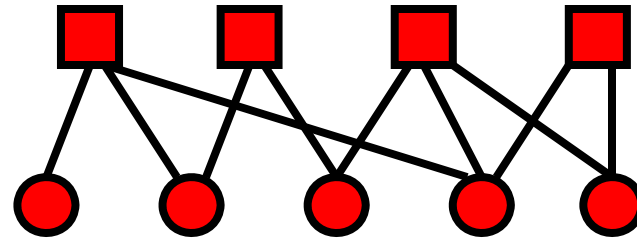


Bethe Approximation

- Pros:
 - Easy to compute, since entropy term involves sum over pairwise and single variables
- Cons:
 - $\hat{F}(P, Q) = F_{\text{bethe}}$ **may or may not** be well connected to $F(P, Q)$
 - It could, in general, be greater, equal or less than $F(P, Q)$
- Optimize each $b(\mathbf{x}_a)$'s.
 - For discrete belief, constrained opt. with *Lagrangian* multiplier
 - For continuous belief, not yet a general formula
 - Not always converge



Bethe Free Energy for FG



$$F_{Bethe} = \sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \ln \frac{b_a(\mathbf{x}_a)}{f_a(\mathbf{x}_a)} + \sum_i (1 - d_i) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \ln b_i(\mathbf{x}_i)$$

$$H_{Bethe} = - \sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \ln b_a(\mathbf{x}_a) + \sum_i (d_i - 1) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \ln b_i(\mathbf{x}_i)$$

$$F_{Bethe} = - \langle f_a(\mathbf{x}_a) \rangle - H_{betha}$$



Minimizing the Bethe Free Energy

- $$L = F_{Bethe} + \sum_i \gamma_i \left\{ 1 - \sum_{x_i} b_i(x_i) \right\} + \sum_a \sum_{i \in N(a)} \sum_{x_i} \lambda_{ai}(x_i) \left\{ b_i(x_i) - \sum_{X_a \setminus x_i} b_a(X_a) \right\}$$

- Set derivative to zero

Constrained Minimization of the Bethe Free Energy



$$L = F_{Bethe} + \sum_i \gamma_i \left\{ \sum_{x_i} b_i(x_i) - 1 \right\} + \sum_a \sum_{i \in N(a)} \sum_{x_i} \lambda_{ai}(x_i) \left\{ \sum_{X_a \setminus x_i} b_a(X_a) - b_i(x_i) \right\}$$

$$\frac{\partial L}{\partial b_i(x_i)} = 0 \quad \Longrightarrow \quad b_i(x_i) \propto \exp\left(\frac{1}{d_i - 1} \sum_{a \in N(i)} \lambda_{ai}(x_i)\right)$$

$$\frac{\partial L}{\partial b_a(X_a)} = 0 \quad \Longrightarrow \quad b_a(X_a) \propto \exp\left(-E_a(X_a) + \sum_{i \in N(a)} \lambda_{ai}(x_i)\right)$$



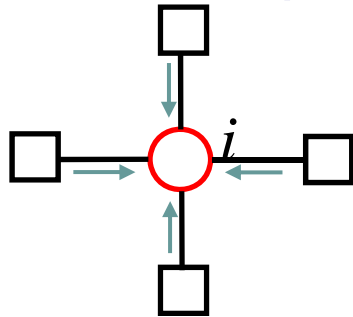
Bethe = BP on FG

- We had:

$$b_i(x_i) \propto \exp\left(\frac{1}{d_i - 1} \sum_{a \in N(i)} \lambda_{ai}(x_i)\right) \quad b_a(X_a) \propto \exp\left(-\log f_a(X_a) + \sum_{i \in N(a)} \lambda_{ai}(x_i)\right)$$

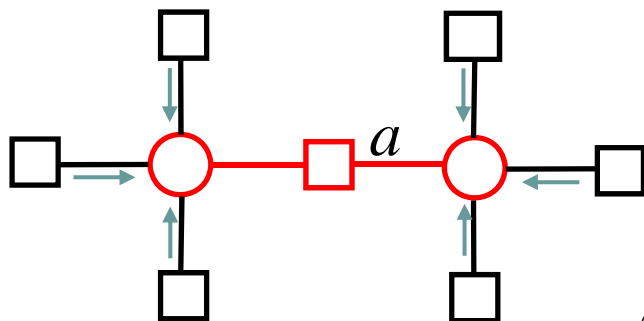
- Identify $\lambda_{ai}(x_i) = \log(m_{i \rightarrow a}(x_i)) = \log \prod_{b \in N(i) \neq a} m_{b \rightarrow i}(x_i)$

- to obtain BP equations:



$$b_i(x_i) \propto f_i(x_i) \prod_{a \in N(i)} m_{a \rightarrow i}(x_i)$$

↑ “beliefs” ↑ “messages”



$$b_a(X_a) \propto f_a(X_a) \prod_{i \in N(a)} \prod_{c \in N(i) \setminus a} m_{c \rightarrow i}(x_i)$$

The “belief” is the BP approximation of the marginal probability.

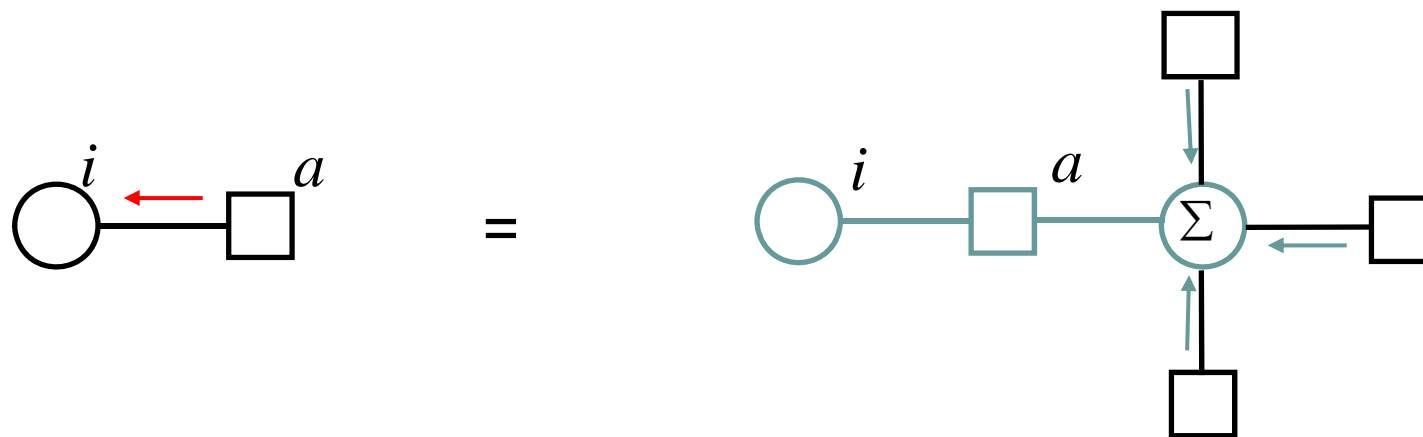


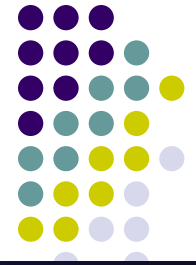
BP Message-update Rules

Using $b_{a \rightarrow i}(x_i) = \sum_{X_a \setminus x_i} b_a(X_a)$, we get

$$m_{a \rightarrow i}(x_i) = \sum_{X_a \setminus x_i} f_a(X_a) \prod_{j \in N(a) \setminus i} \prod_{b \in N(j) \setminus a} m_{b \rightarrow j}(x_j)$$

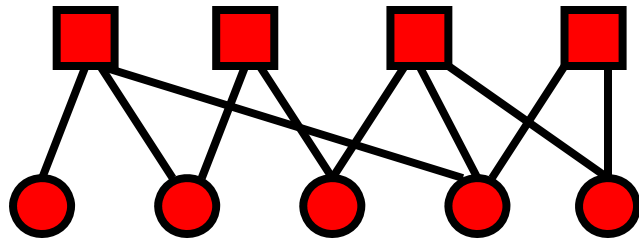
(A sum product algorithm)





Summary so far

$$P(X) = 1/Z \prod_{f_a \in F} f_a(X_a)$$



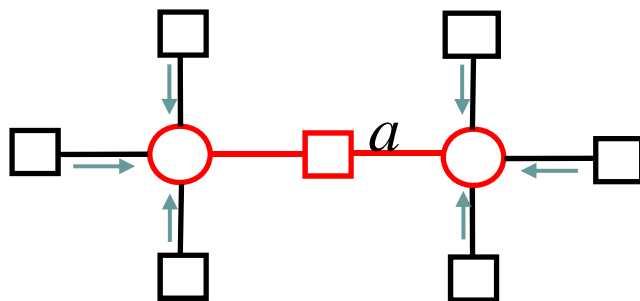
$$F(P, Q) = -H_Q(X) - \sum_{f_a \in F} E_Q \log f_a(X_a)$$



$$\hat{F}(P, Q) = \sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \log \frac{f_a(\mathbf{x}_a)}{b_a(\mathbf{x}_a)} + \sum_i (1 - d_i) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \log b_i(\mathbf{x}_i)$$



$$b_a(X_a) \propto \exp \left(-\log f_a(X_a) + \sum_{i \in N(a)} \lambda_{ai}(x_i) \right)$$



$$b_i(x_i) \propto \exp \left(\frac{1}{d_i - 1} \sum_{a \in N(i)} \lambda_{ai}(x_i) \right)$$

© Eric Xing @ CMU, 2005-2016



The Theory Behind LBP

- For a distribution $p(\mathbf{X}/\theta)$ associated with a complex graph, computing the marginal (or conditional) probability of arbitrary random variable(s) is intractable
- Variational methods
 - formulating probabilistic inference as an optimization problem:

$$q^* = \arg \min_{q \in \mathcal{S}} \left\{ F_{Bethe}(p, q) \right\}$$

$$F_{Bethe} = \sum_a \sum_{\mathbf{x}_a} b_a(\mathbf{x}_a) \ln \frac{b_a(\mathbf{x}_a)}{f_a(\mathbf{x}_a)} + \sum_i (1 - d_i) \sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) \ln b_i(\mathbf{x}_i) = -\langle f_a(\mathbf{x}_a) \rangle - H_{bethe}$$

q : a (tractable) probability distribution



The Theory Behind LBP

- But we do not optimize $q(\mathbf{X})$ explicitly, focus on the set of beliefs

- *e.g.*, $b = \{b_{i,j} = \tau(x_i, x_j), b_i = \tau(x_i)\}$

- Relax the optimization problem

- approximate objective:

$$H_q \approx F(b)$$

- relaxed feasible set:

$$\mathcal{M} \rightarrow \mathcal{M}_o \quad (\mathcal{M}_o \supseteq \mathcal{M})$$

$$b^* = \arg \min_{b \in \mathcal{M}_o} \left\{ \langle E \rangle_b + F(b) \right\}$$

- The loopy BP algorithm:

- a fixed point iteration procedure that tries to solve b^*



The Theory Behind LBP

- But we do not optimize $q(\mathbf{X})$ explicitly, focus on the set of beliefs

- *e.g.*, $b = \{b_{i,j} = \tau(x_i, x_j), b_i = \tau(x_i)\}$

- Relax the optimization problem

- approximate objective: $H_{\text{Betha}} = H(b_{i,j}, b_i)$

- relaxed feasible set: $\mathcal{M}_o = \{ \tau \geq 0 \mid \sum_{x_i} \tau(x_i) = 1, \sum_{x_i} \tau(x_i, x_j) = \tau(x_j) \}$

$$b^* = \arg \min_{b \in \mathcal{M}_o} \left\{ \langle E \rangle_b + F(b) \right\}$$

- The loopy BP algorithm:

- a fixed point iteration procedure that tries to solve b^*

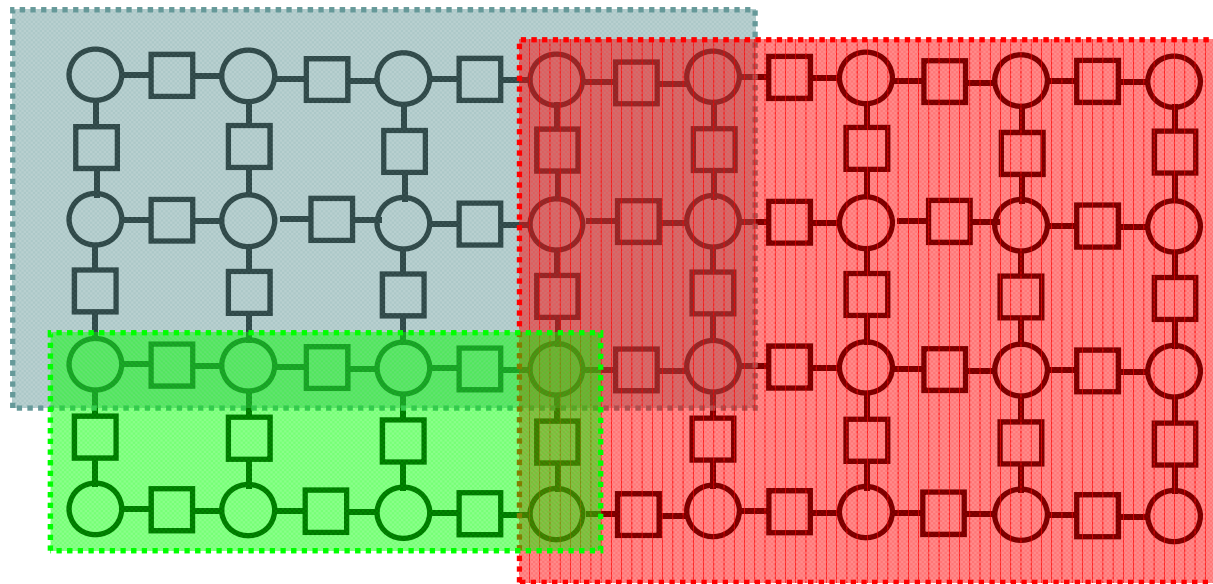
Region-based Approximations to the Gibbs Free Energy

(Kikuchi, 1951)



Exact: $G[q(X)]$ (*intractable*)

Regions: $G[\{b_r(X_r)\}]$



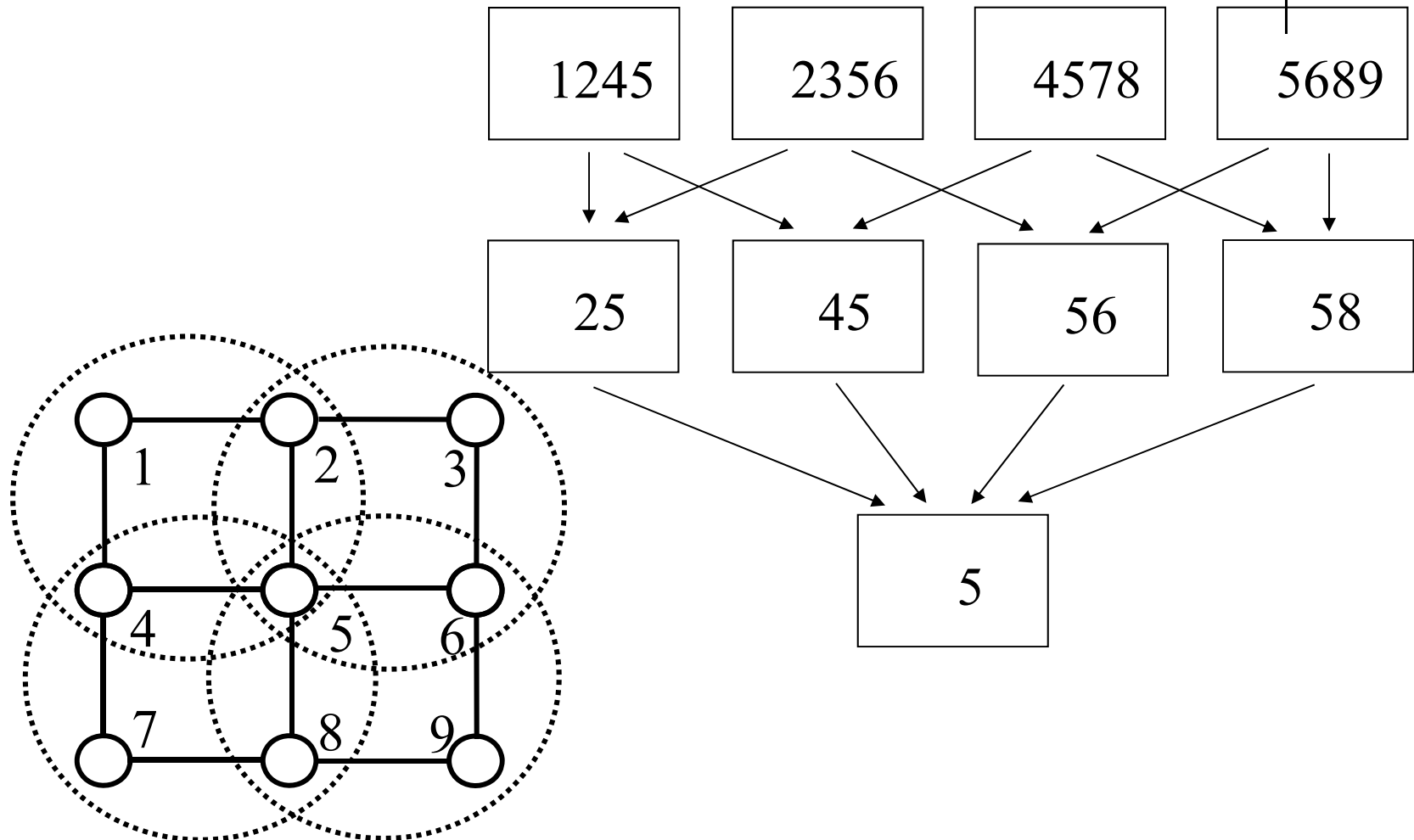
Generalized Belief Propagation



- Belief in a region is the product of:
 - Local information (factors in region)
 - Messages from parent regions
 - Messages into descendant regions from parents who are not descendants.
- Message-update rules obtained by enforcing marginalization constraints.

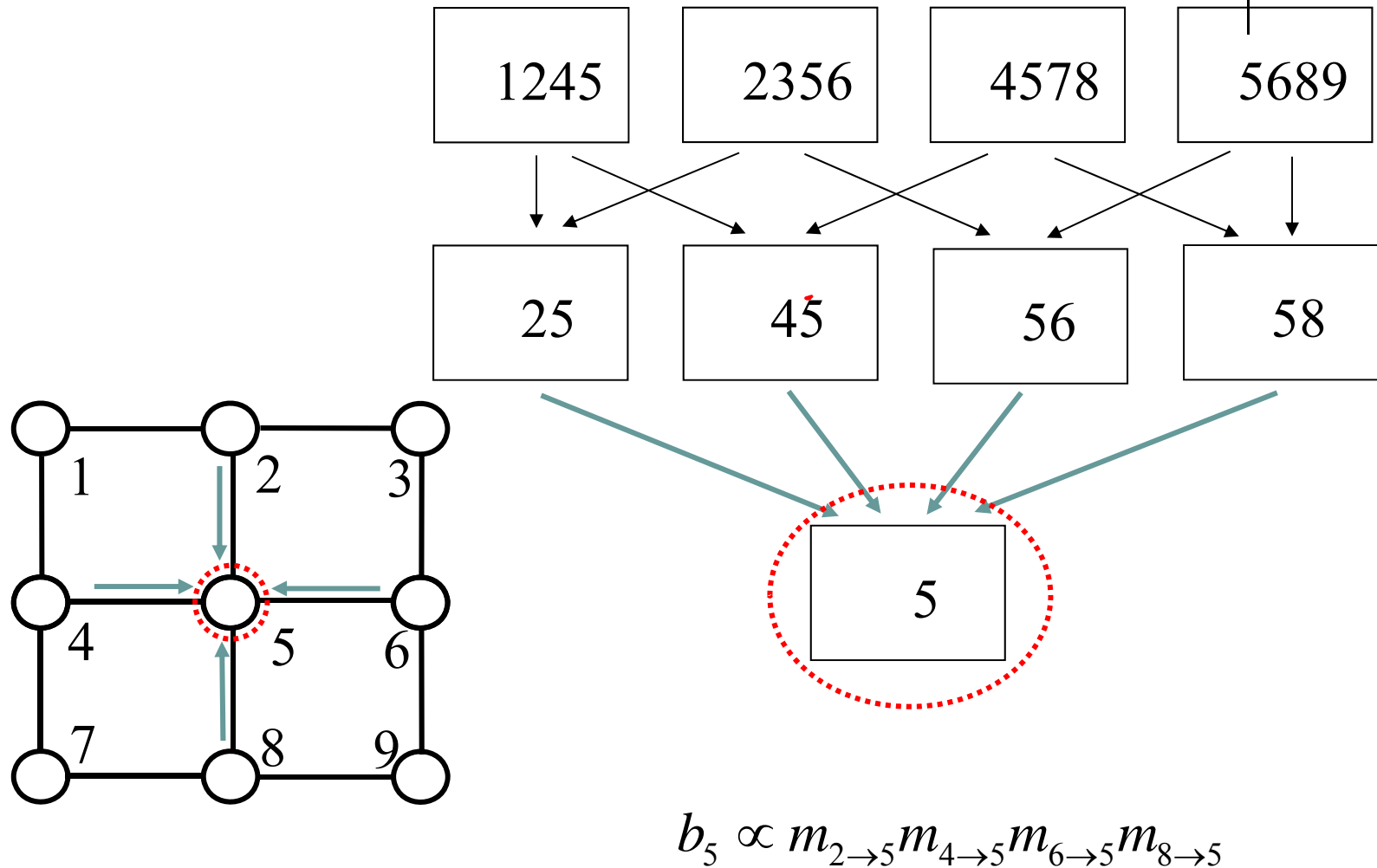


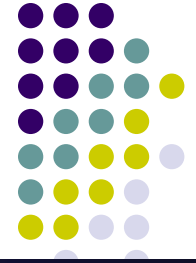
Generalized Belief Propagation



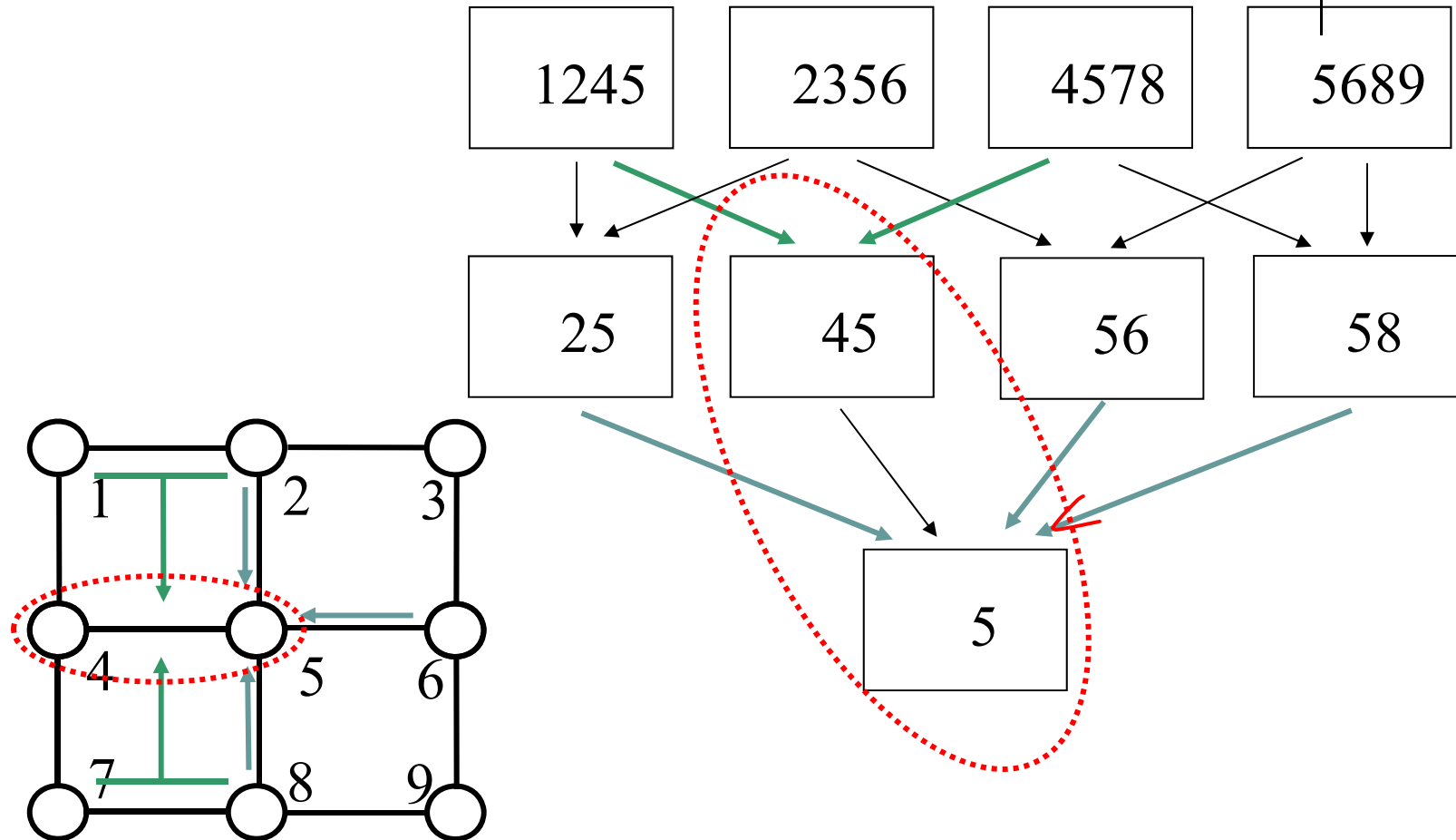


Generalized Belief Propagation





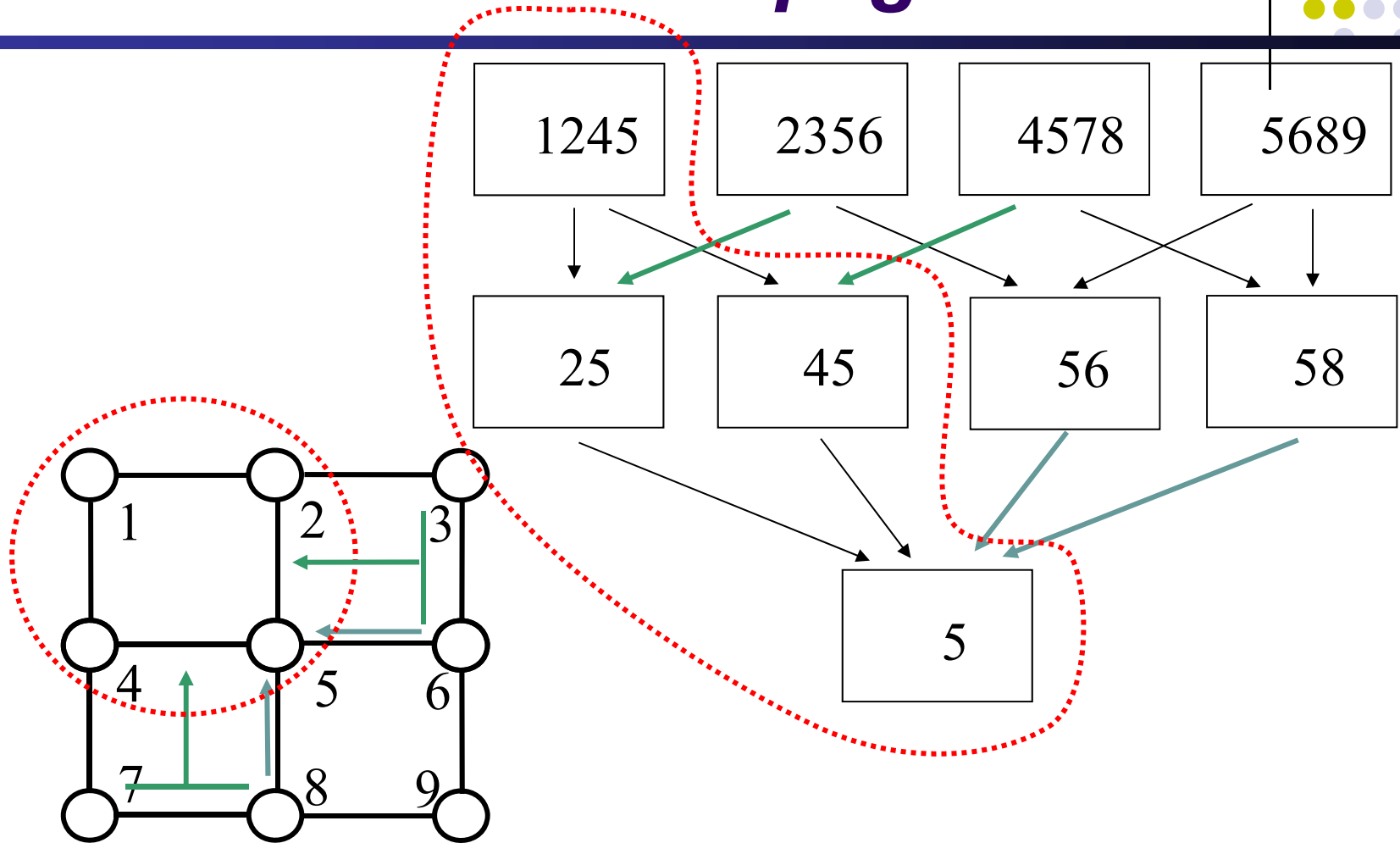
Generalized Belief Propagation



$$b_{45} \propto [f_{45}] [m_{12 \rightarrow 45} m_{78 \rightarrow 45} m_{2 \rightarrow 5} m_{6 \rightarrow 5} m_{8 \rightarrow 5}]$$

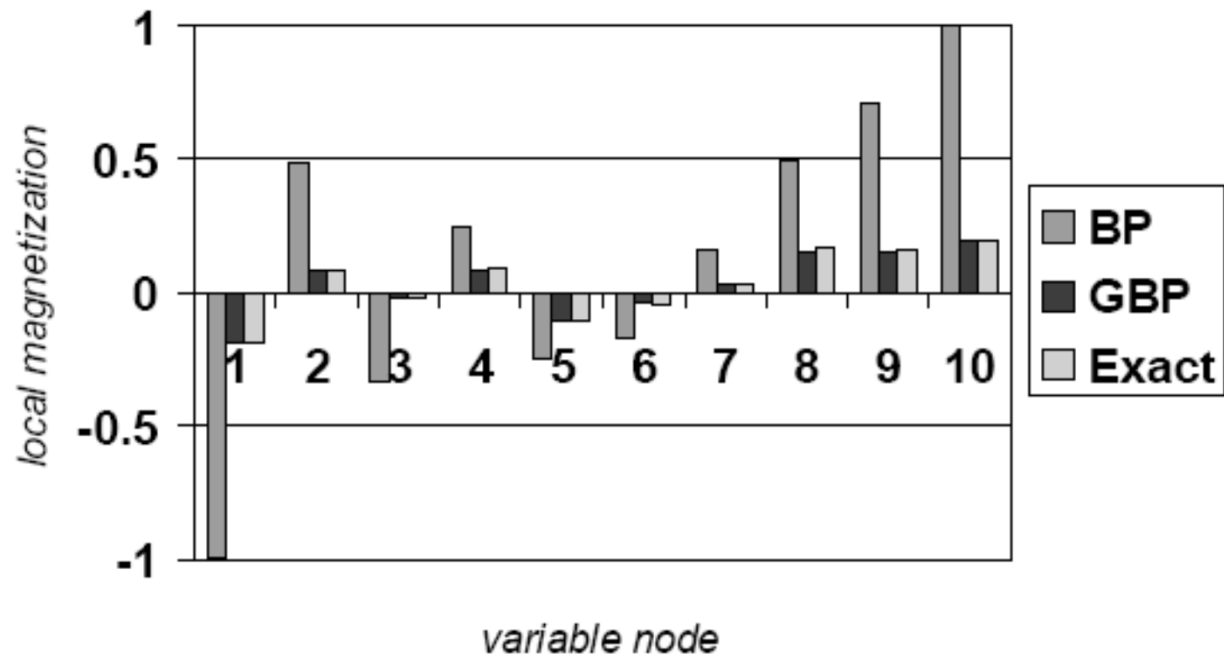


Generalized Belief Propagation

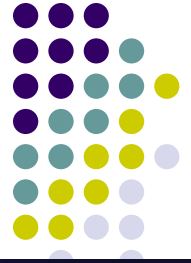


$$b_{1245} \propto [f_{12} f_{14} f_{25} f_{45}] [m_{36 \rightarrow 25} m_{78 \rightarrow 45} m_{6 \rightarrow 5} m_{8 \rightarrow 5}]$$

Some results



Summary



- We defined an objective function (F) for approximate inference
- However, we found that optimizing this function was hard
- We first approximated objective function F to simpler F_{bethe}
 - Minima of F_{bethe} turned out to be fixed points of BP
- Then we extended this to more complicated approximations
 - The resulting algorithms come under a family called Generalized Belief Propagation
- Next class, we will cover other methods of approximations