INFERENCE IN BAYESIAN NETWORKS
- BELIEF PROPAGATION

Instructor: Sael Lee

Combination of slides:

"Pearl's algorithm" by Tomas Singliar & Daniel Lowd's slide for UW CSE 573 & "B elief Propagation" by Jakob Metzler & " Generalized BP" by Jonathan Yedidia

OUTLINE

- Motivation
- Pearl's BP Algorithm
- Generalized Belief Propagation

PROBABILISTIC INFERENCE

Computing the a posteriori belief of a variable in a general Bayesian Network is NP-hard

- Approximate inference
  - MCMC sampling
  - Belief Propagation

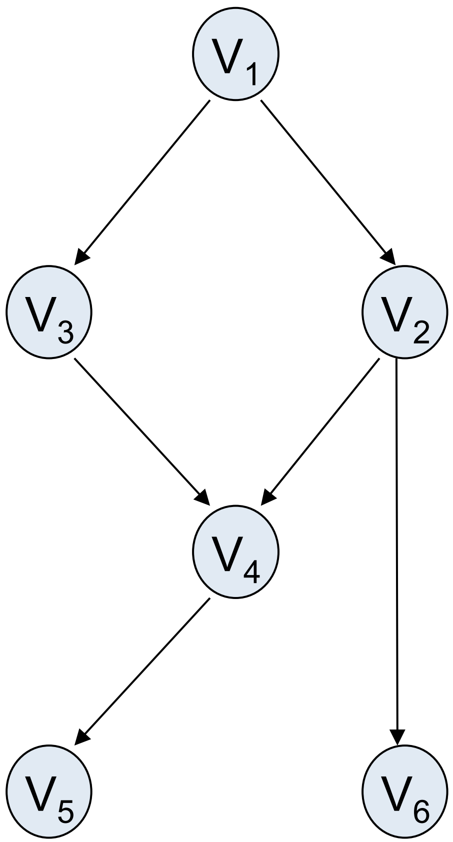
BELIEF PROPAGATION

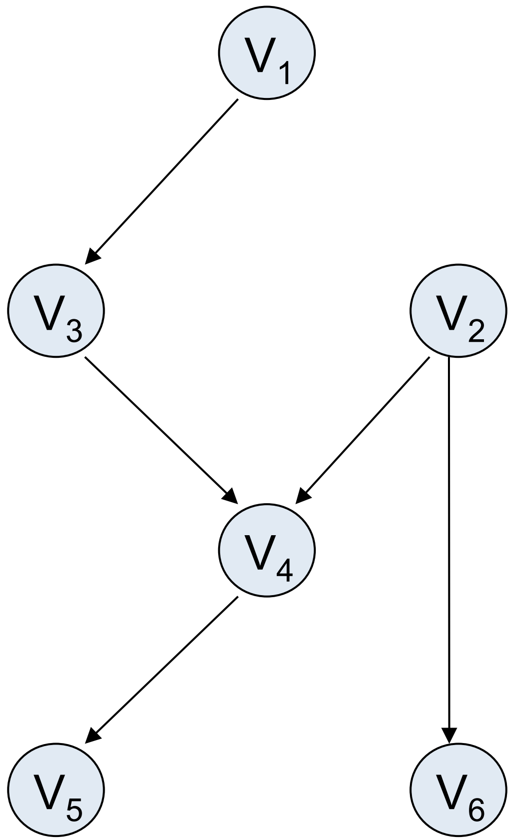
- BP is a message passing algorithm that solves approxi mate inference problems in graphical model, including Bayesian networks and Markov random fields.
- Calculates marginal distribution for each of the unobs erved variable, conditional on any observed variables.
- It was first proposed by Judea Pearl in 1982 for trees (exact) and later extended to polytrees and general gra phs (approximate).

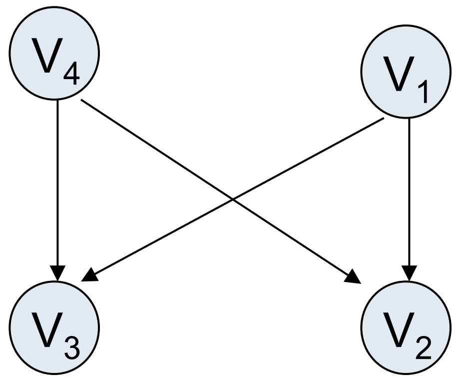
BAYESIAN BELIEF NETWORKS

- *(G, P)* directed acyclic graph with the joint p.d. *P*
- each node is a variable of a multivariate distribution
- links represent causal dependencies
  - CPT in each node
- Polytree
  - What is a polytree?
    - A Bayesian network graph is a *polytree* if (an only if) there is at most one path between any two nodes, V_i and V_k
    - implies each node separates the graph into two disjoint compone nts
  - Why do we care about polytrees?
    - Exact BN inference is NP-hard...
    - ...but on polytrees, takes linear time.

EXAMPLES: POLYTREE OR NOT?







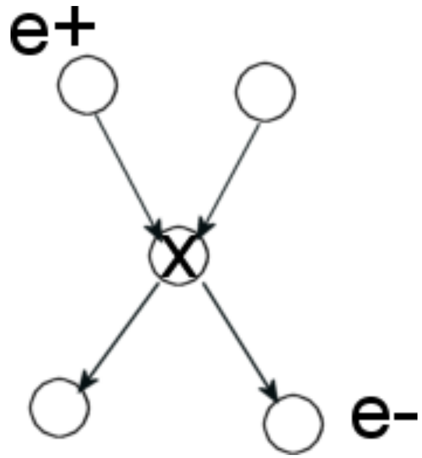
OUR INFERENCE TASK

- We know the values of some *evidence variables* E:

$$V_{e_1}, \ldots, V_{e_{|E|}}$$

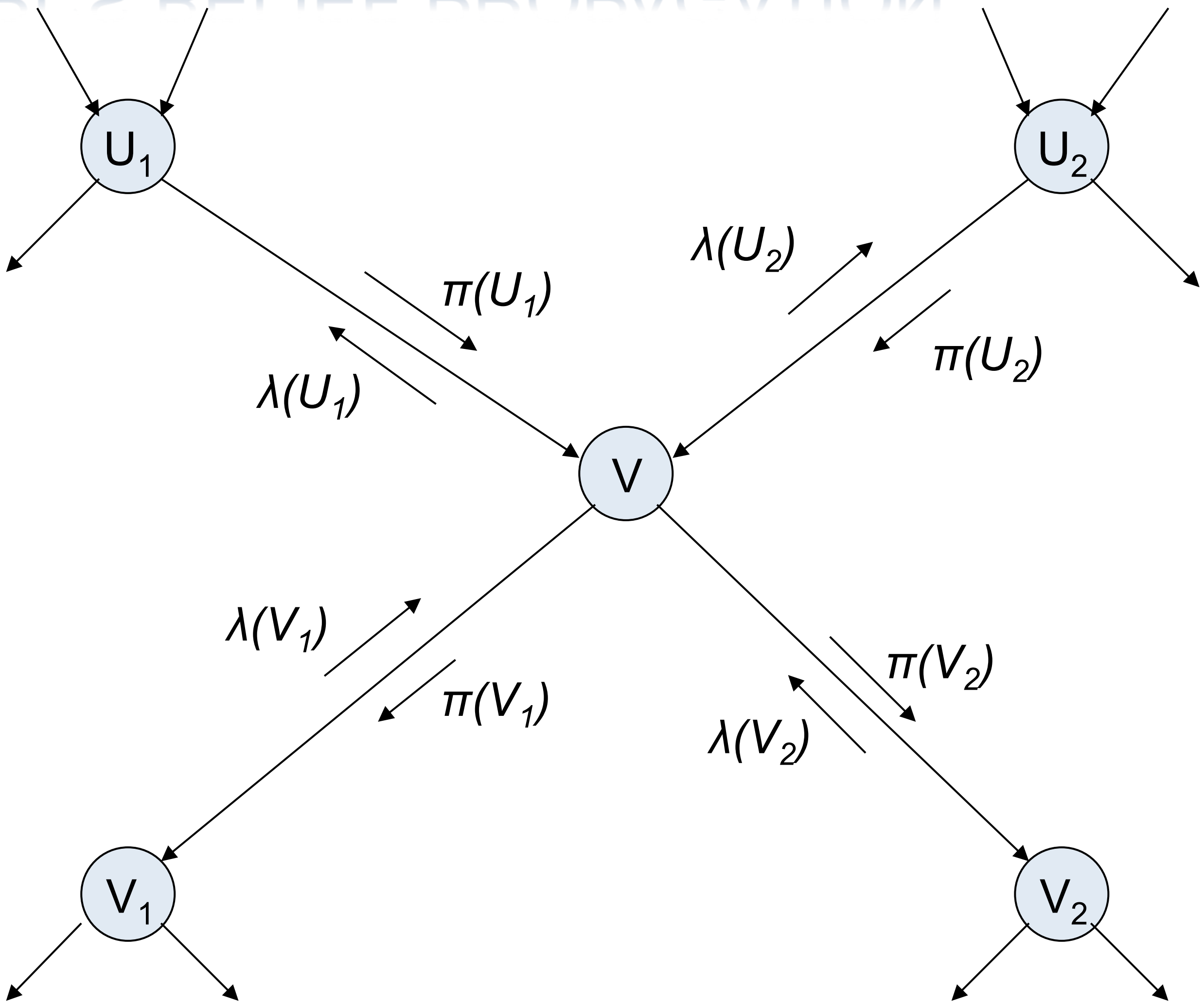
- We wish to compute the posterior probability $P(X_i \mid E)$ for **all** *non-evidence variables* X_i.

PEARL'S BELIEF PROPAGATION



- We have the evidence *E*
- Local computation for one node *V* desired
- Information flows through the paths of *G*
  - *flows as messages of two types – λ and π*
- ***V* splits** network into two *disjoint* parts
  - Strong independence assumptions induced – crucial!
- Denote E_V^+ the part of evidence accessible through the parents of ***V*** **(*causal*)**
  - passed downward in ***π messages***
- Analogously, let E_V^- be the ***diagnostic* evidence**
  - passed upwards in ***λ messages***

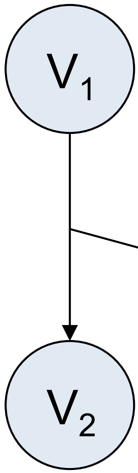
PEARL'S BELIEF PROPAGATION



THE Π MESSAGES

- What are the messages?
- For simplicity, let the nodes be binary

V_1=T	0.8
V_1=F	0.2



P	V_1=T	V_1=F
V_2=T	0.4	0.9
V_2=F	0.6	0.1

The message passes on information.

What information? Observe:

$$P(V_2) = P(V_2 | V_1=T)P(V_1=T) + P(V_2 | V_1=F)P(V_1=F)$$

The information needed is the CPT of $V_1 = \pi_V(V_1)$

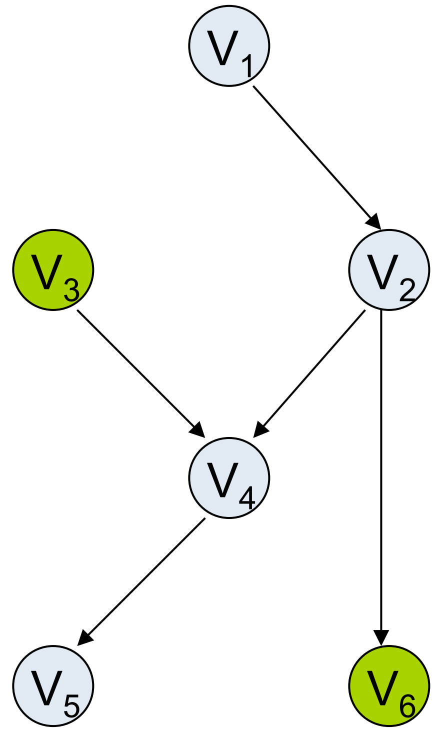
π Messages capture information passed from parent to child

THE EVIDENCE

- Evidence – values of observed nodes
  - $V_3 = T, V_6 = 3$
- Our belief in what the value of V_i 'should' be changes.
- This belief is propagated
- As if the CPTs became

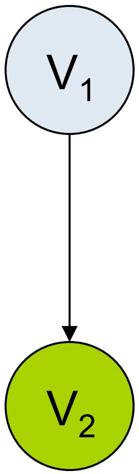
V_3=T	1.0
V_3=F	0.0

P	V_2=T	V_2=F
V_6=1	0.0	0.0
V_6=2	0.0	0.0
V_6=3	1.0	1.0



THE Λ MESSAGES

- We know what the π messages are
- What about λ?



Assume E = { V_2 } and compute by Bayes rule:

$$P(V_1 \mid V_2) = \frac{P(V_1)P(V_2 \mid V_1)}{P(V_2)} = \alpha P(V_1)P(V_2 \mid V_1)$$

The information not available at V_1 is the $P(V_2|V_1)$. To be passed upwards by a λ-message. Again, this is not in general exactly the CPT, but the *belief* based on evidence down the tree.

- The messages are $\pi(V)=P(V|E^+)$ and $\lambda(V)=P(E^-|V)$

COMBINATION OF EVIDENCE

- Let $E_V = E_V^+ \cup E_V^-$ and let us compute

$$P(V \mid E) = P(V \mid E_V^+, E_V^-) = \alpha' P(E_V^+, E_V^- \mid V) P(V) =$$

$$\alpha' P(E_V^- \mid V) P(E_V^+ \mid V) P(V) = \alpha P(E_V^- \mid V) P(V \mid E_V^+) =$$

$$\alpha \lambda(V) \pi(V) = BEL(V)$$

- α is the normalization constant
- normalization is not necessary (can do it at the end)
- but may prevent numerical underflow problems

MESSAGES

- Assume *X* received λ-messages from neighbors
- How to compute $\lambda(X) = p(E^- | X)$?
- Let $Y_1, \ldots, Y_c$ be the children of *X*
- $\lambda_{XY}(x)$ denotes the λ-message sent between X and Y

$$\lambda(X) = \prod_{j=1}^{c} \lambda_{Y_j X}(X)$$

MESSAGES

- Assume X received π -messages from neighbors
- How to compute $\pi(X) = p(X|E^{+})$?
- Let $U_1, \ldots, U_p$ be the parents of X
- $\pi_{XY}(x)$ denotes the π-message sent between X and Y
- summation over the CPT

$$\pi(X) = \sum_{u_1,\ldots,u_p} P(X \mid U_1,\ldots,U_p) \prod_{j=1}^{p} \pi_{U_j X}(U_j)$$

MESSAGES TO PASS

- We need to compute $\pi_{XY}(x)$

$$\pi_{XY_J}(x) = \alpha\pi_X(x)\prod_{k \neq j} \lambda_{Y_k X}(x)$$

- Similarly, $\lambda_{XY}(x)$, X is parent, Y child
- Symbolically, group other parents of Y into V = V_1, … , V_q

$$\lambda_{Y_j X}(x) = \sum_{y_j} \lambda_{Y_j}(y_j) \sum_{v_1,\ldots,v_q} p(y \mid v_1,\ldots,v_q) \prod_{k=1}^{q} \pi_{V_k Y_j}(v_k)$$

PEARL'S BP ALGORITHM

- Initialization
  - For nodes with evidence *e*
    - $\lambda(x_i) = 1$ wherever $x_i = e_i$; 0 otherwise
    - $\pi(x_i) = 1$ wherever $x_i = e_i$; 0 otherwise
  - For nodes without parents
    - $\pi(x_i) = p(x_i)$ - prior probabilities
  - For nodes without children
    - $\lambda(x_i) = 1$ uniformly (normalize at end)

THE PEARL BELIEF PROPAGATION ALGORITHM

- Iterate until no change occurs
  - (For each node X) if X has received all the π messages from its parents, calculate $\pi(x)$
  - (For each node X) if X has received all the λ messages from its children, calculate $\lambda(x)$
  - (For each node X) if $\pi(x)$ has been calculated and X received all the λ-messages from all its children (except Y), calculate $\pi_{XY}(x)$ and send it to Y.
  - (For each node X) if $\lambda(x)$ has been calculated and X received all the π-messages from all parents (except U), calculate $\lambda_{XU}(x)$ an d send it to U.
- Compute Belief BEL(X) = $\lambda(x)\pi(x)$
- and normalize

PROPERTIES OF BP

- Exact for polytrees
  - Each node separates Graph into 2 disjoint components
- On a polytree, the BP algorithm converges in time proportio nal to diameter of network – at most linear
- Work done in a node is proportional to the size of CPT
  - Hence BP is linear in number of network parameters
- For general BBNs
  - Exact inference is NP-hard
  - Approximate inference is NP-hard

LOOPY BELIEF PROPAGATION

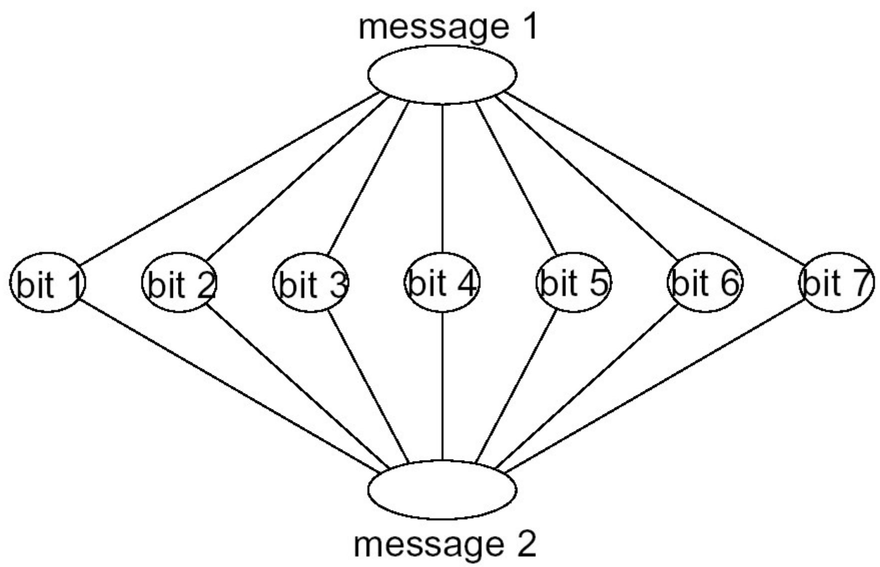
- Most graphs are not polytrees
  - Cutset conditioning
  - Clustering
    - Join Tree Method
  - Approximate Inference
    - Loopy BP

LOOPY BELIEF PROPAGATION

- If BP is used on graphs with loops, messages may circulate indefinitely
- 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

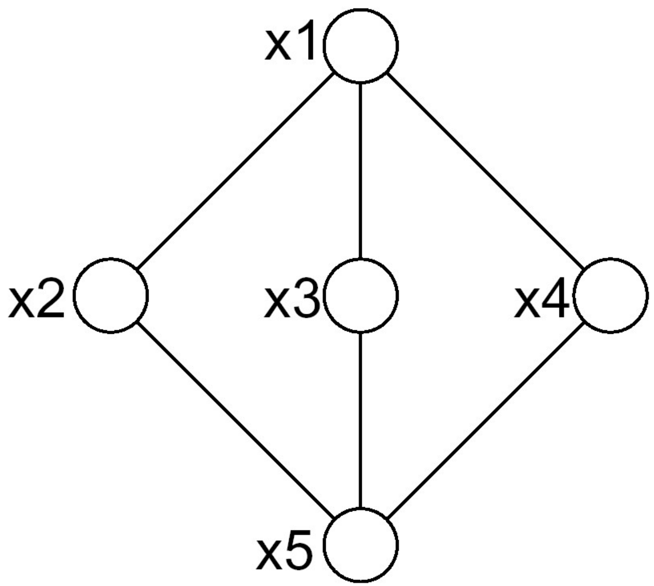
LOOPY BELIEF PROPAGATION

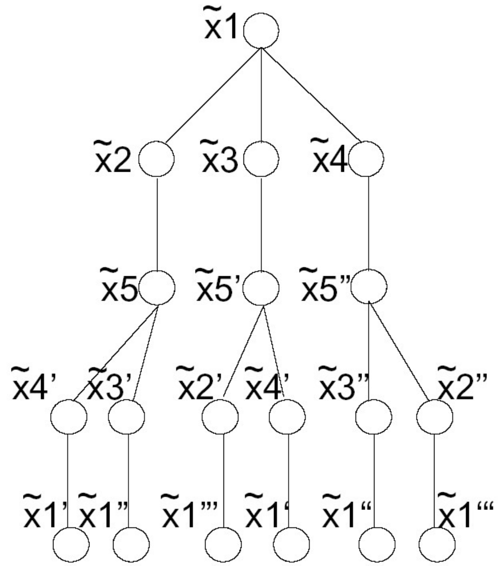
- Just apply BP rules in spite of loops
- In each iteration, each node sends all messages in parallel
- Seems to work for some applications



TROUBLE WITH LBP

- May not converge
  - A variety of tricks can help
- Cycling Error – old information is mistaken as new
- Convergence Error – unlike in a tree, neighbors need not be independent. However, LBP treats them as if they were.





Bolt & Gaag “On the convergence error in loopy propagation” (2004).

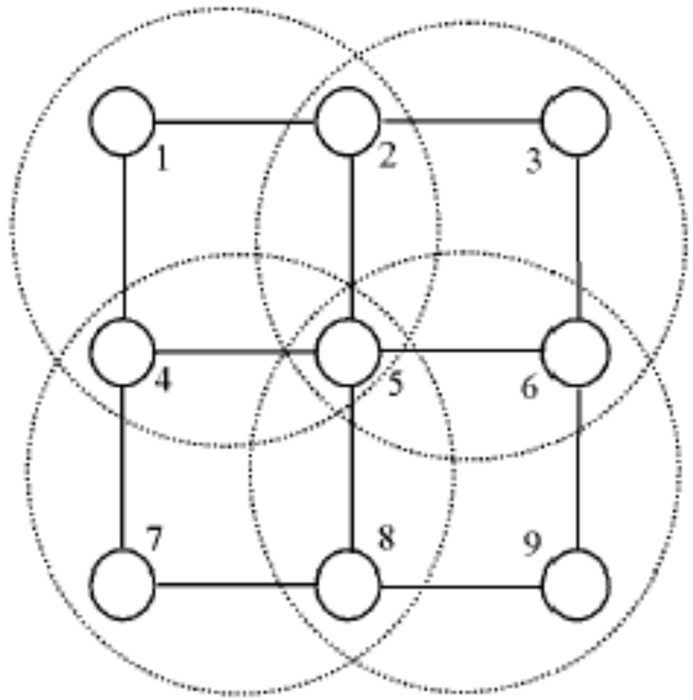
GENERALIZED BP

- We can try to improve inference by taking into accoun t higher-order interactions among the variables
- An intuitive way to do this is to define messages that propagate between groups of nodes rather than just s ingle nodes
- This is the intuition in Generalized Belief Propagation (GPB)

GBP ALGORITHM

1) Split the graph into basic clusters

[1245],[2356],
[4578],[5689]

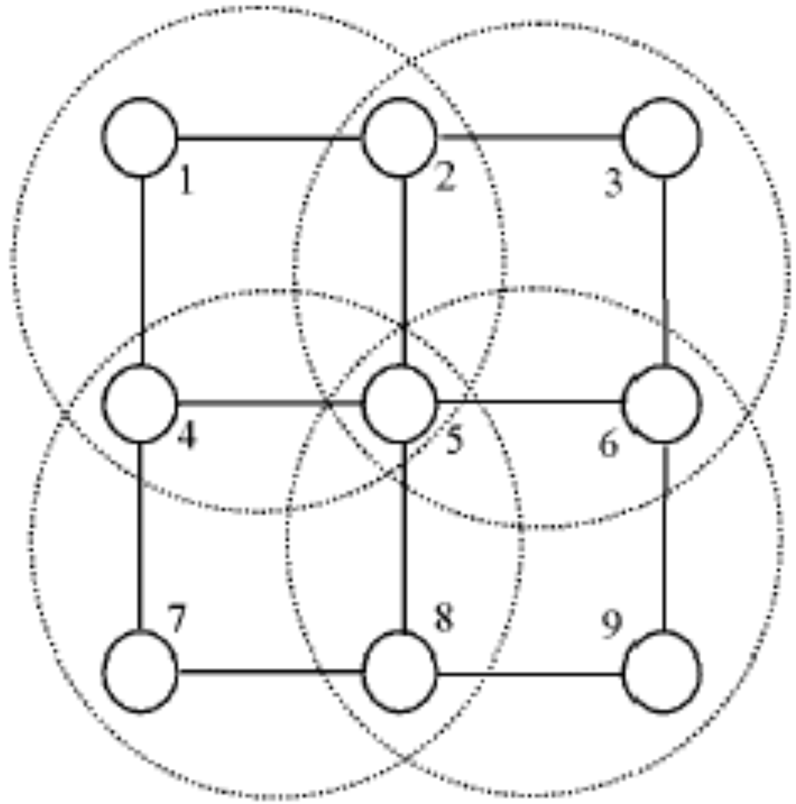


GBP ALGORITHM

2) Find all intersection regions of the basic clusters, and all their intersections

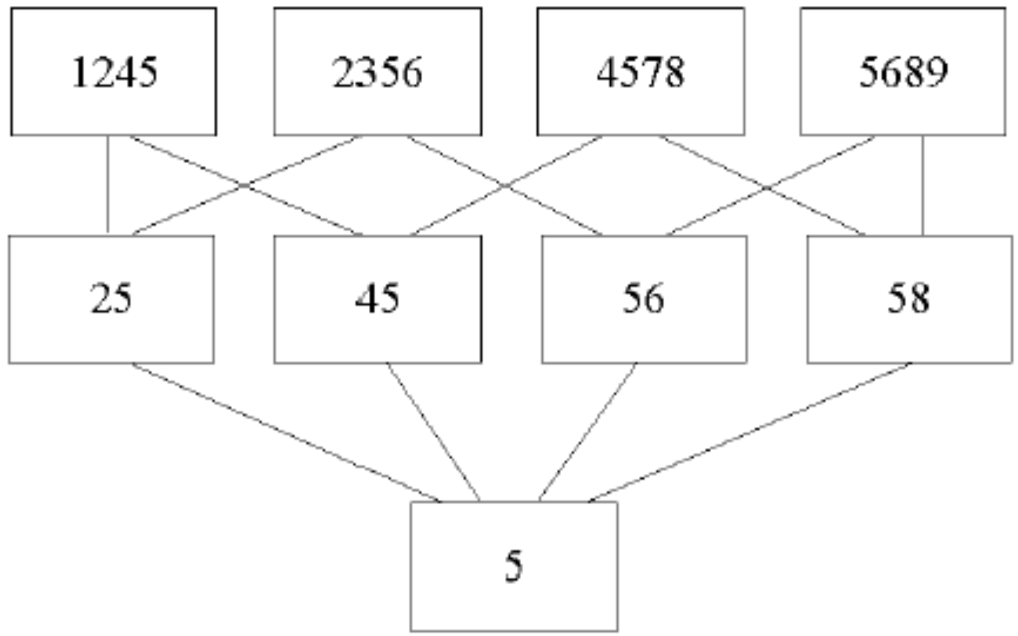
[25], [45], [56], [58],

[5]



GBP ALGORITHM

3) Create a hierarchy of regions and their direct sub-reg ions



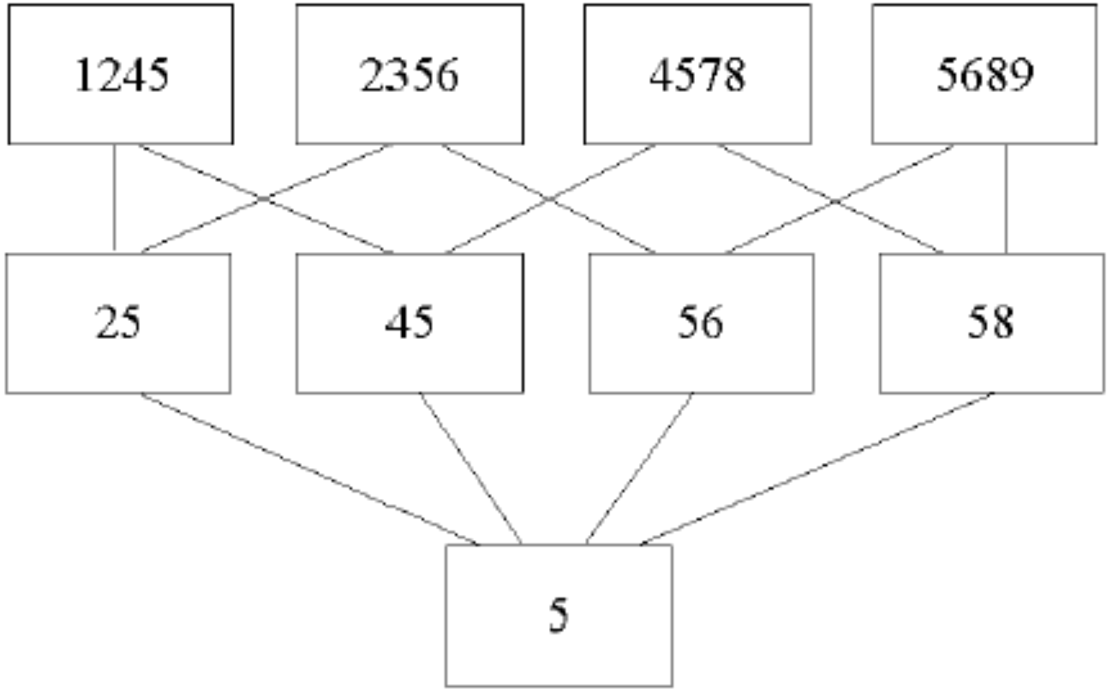
GBP ALGORITHM

4) Associate a message with each line in the graph

e.g. message from

[1245]->[25]:

$m_{14->25}(x_2,x_5)$



GBP ALGORITHM

5) Setup equations for beliefs of regions

- remember from earlier:

$$b_i(x_i) = k\phi_i(x_i) \prod_{j \in N(i)} m_{ji}(x_i)$$

- So the belief for the region containing [5] is:

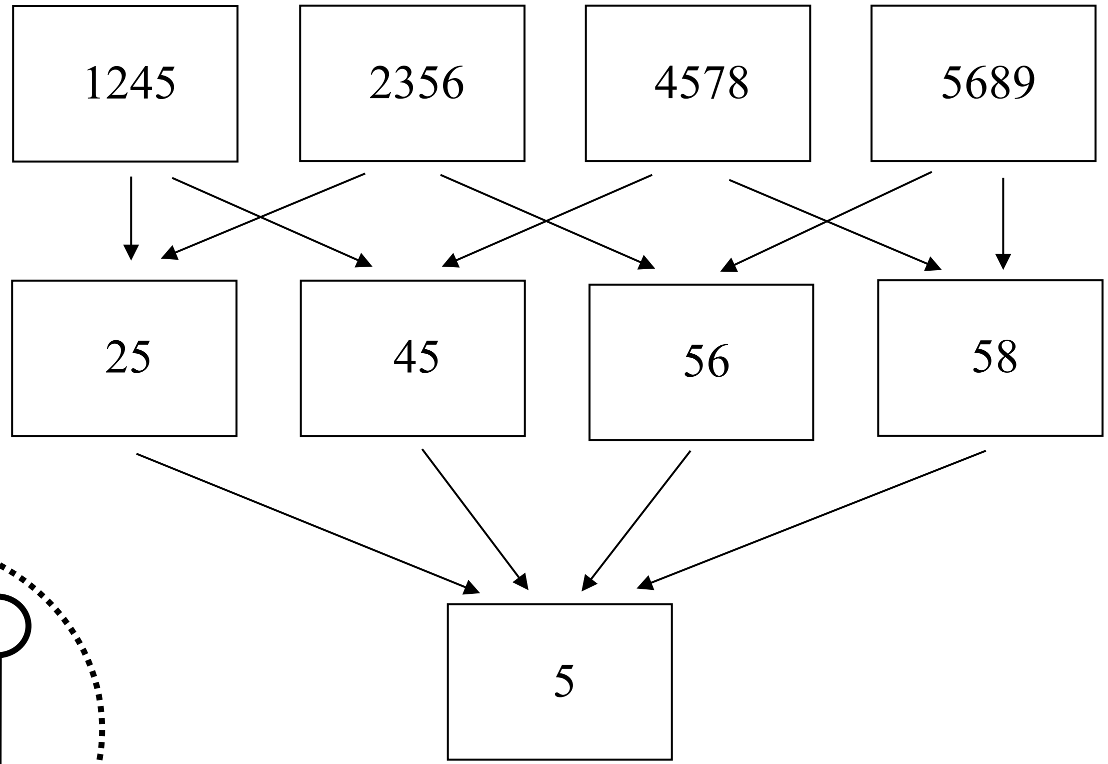
$$b_5 = k\,[\phi_5]\,[m_{2\rightarrow 5} m_{4\rightarrow 5} m_{6\rightarrow 5} m_{8\rightarrow 5}]$$

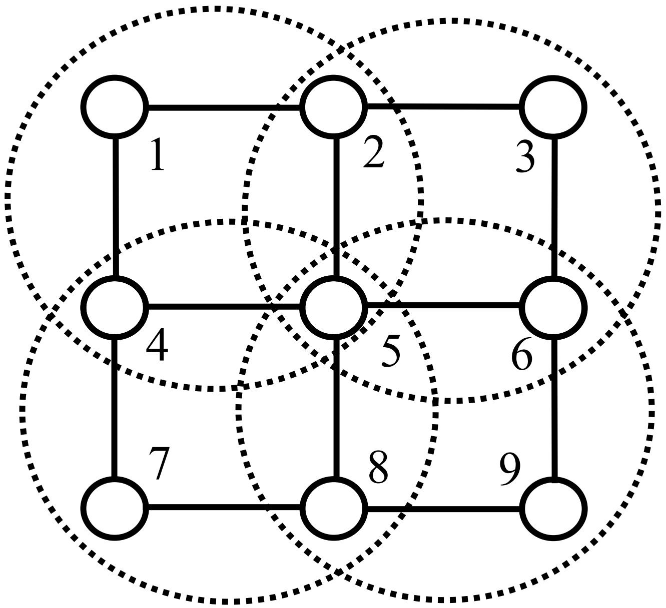
- for the

$$b_{45} = k\,[\phi_4 \phi_5 \psi_{45}]\,[m_{12\rightarrow 45} m_{78\rightarrow 45} m_{2\rightarrow 5} m_{6\rightarrow 5} m_{8\rightarrow 5}]$$

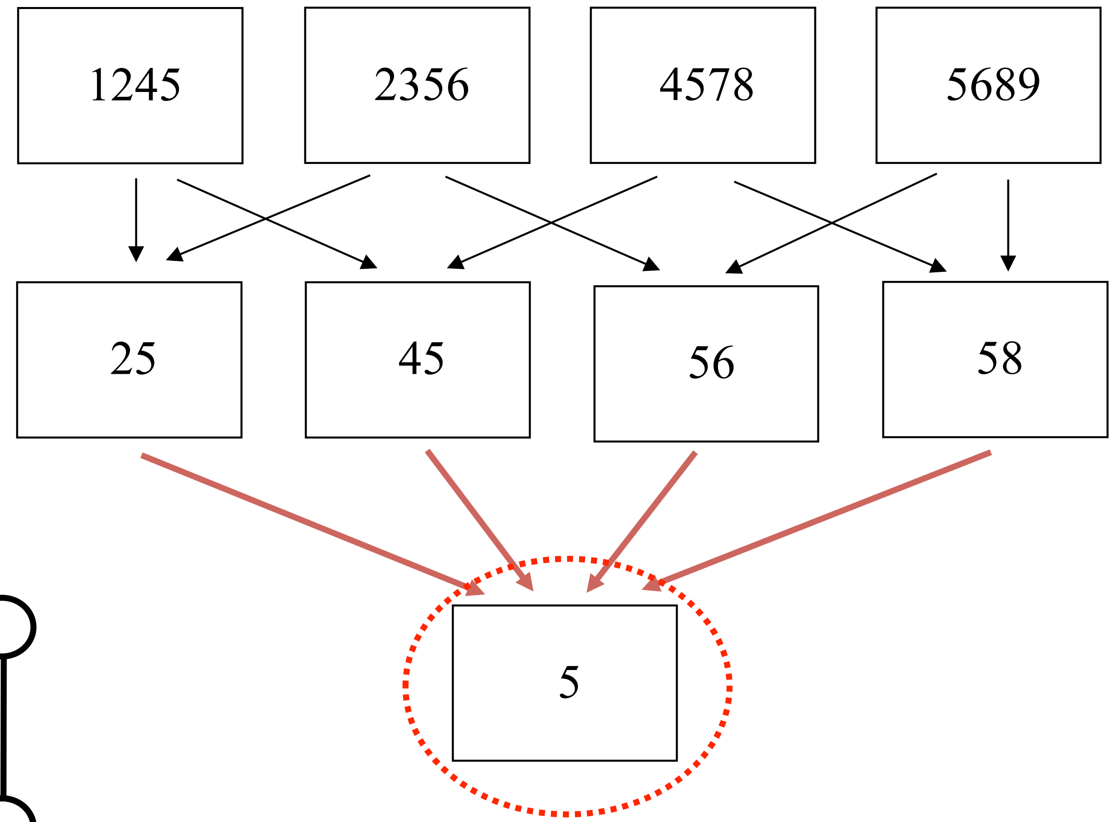
- etc.

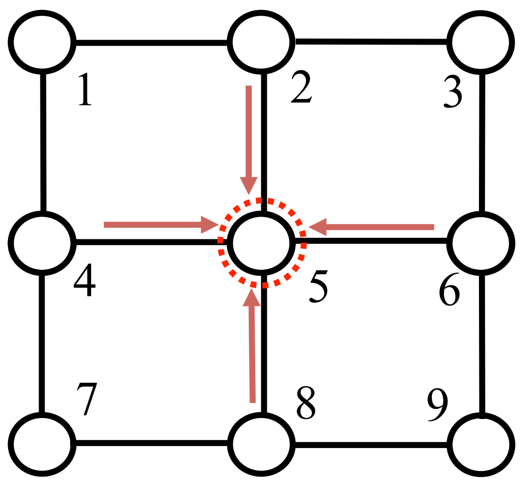
Generalized Belief Propagation





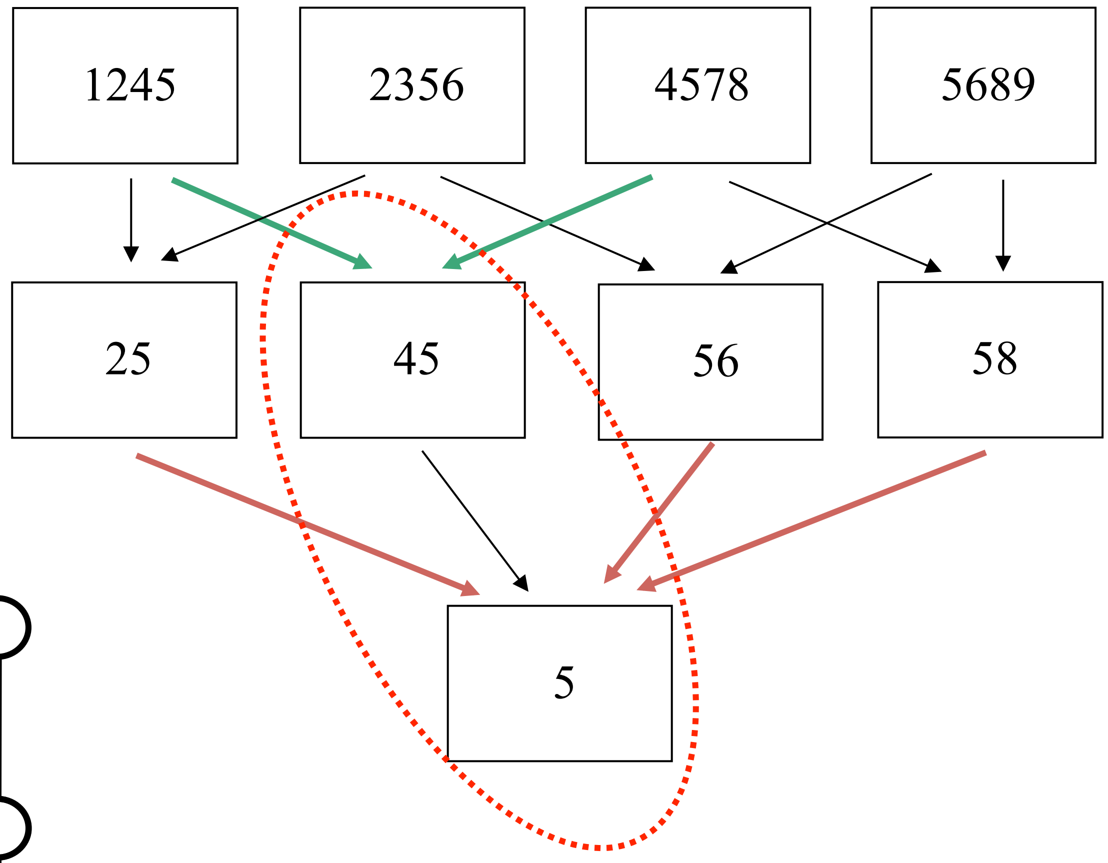
Generalized Belief Propagation

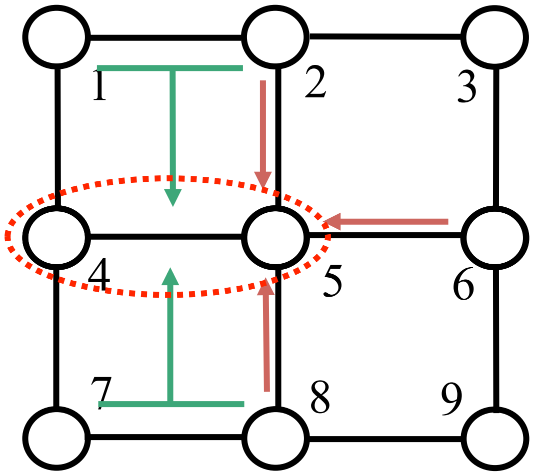




$$b_5 \propto m_{2\to5} m_{4\to5} m_{6\to5} m_{8\to5}$$

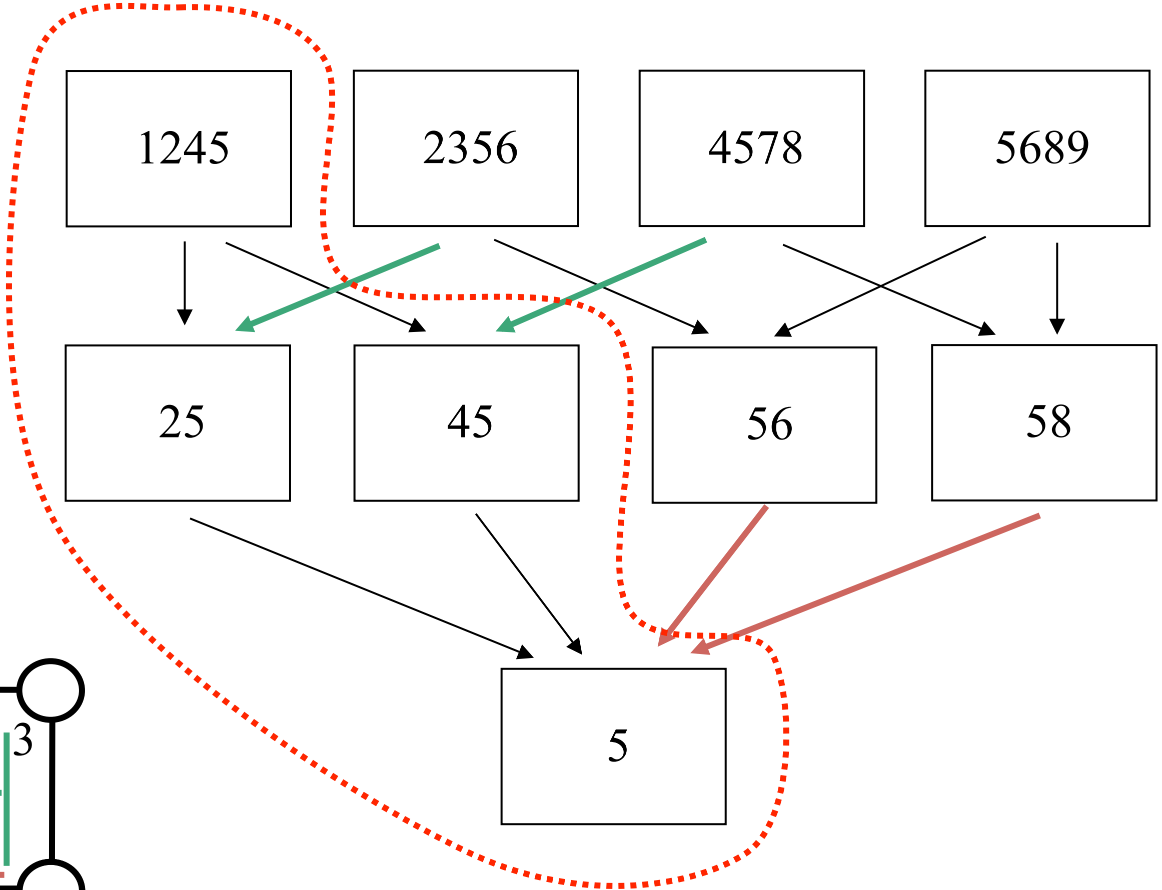
Generalized Belief Propagation

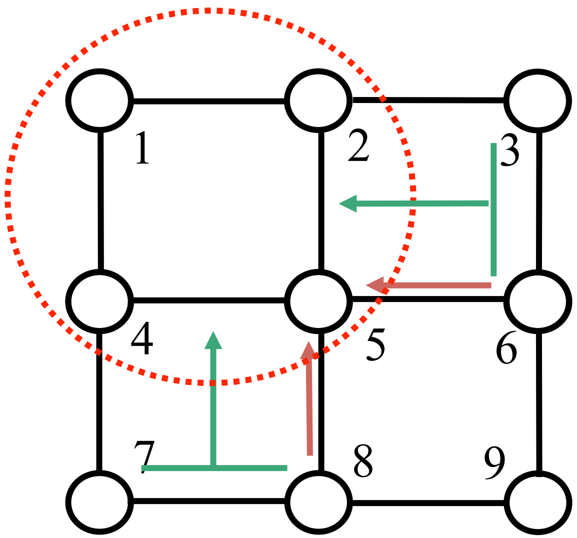




$$b_{45} \propto [f_{45}][m_{12\to 45} m_{78\to 45} m_{2\to 5} m_{6\to 5} m_{8\to 5}]$$

Generalized Belief Propagation

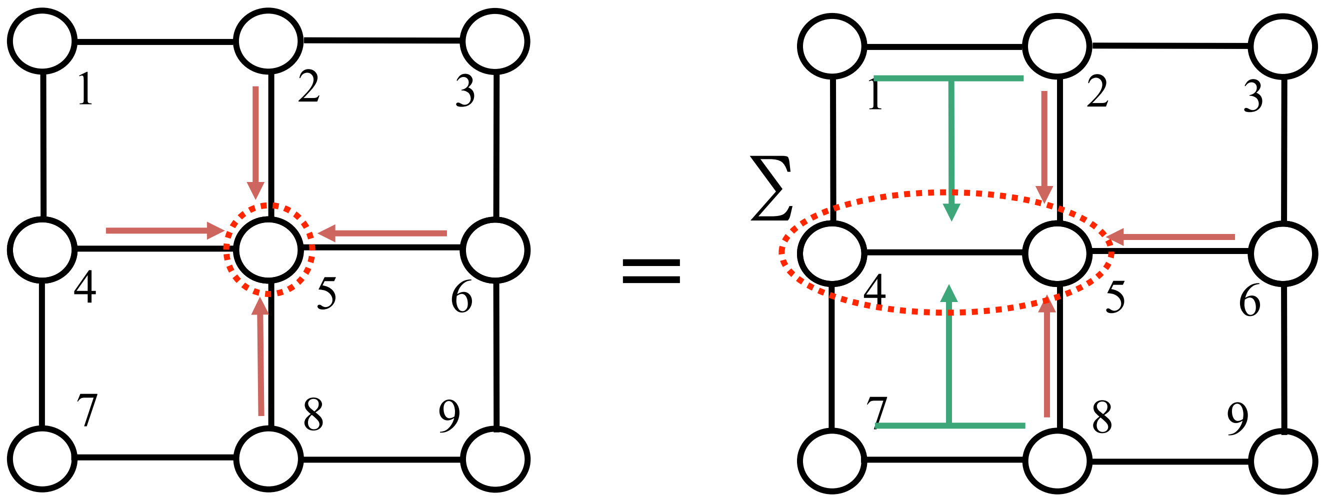




$$b_{1245} \propto [f_{12} f_{14} f_{25} f_{45}][m_{36\to 25} m_{78\to 45} m_{6\to 5} m_{8\to 5}]$$

Generalized Belief Propagation

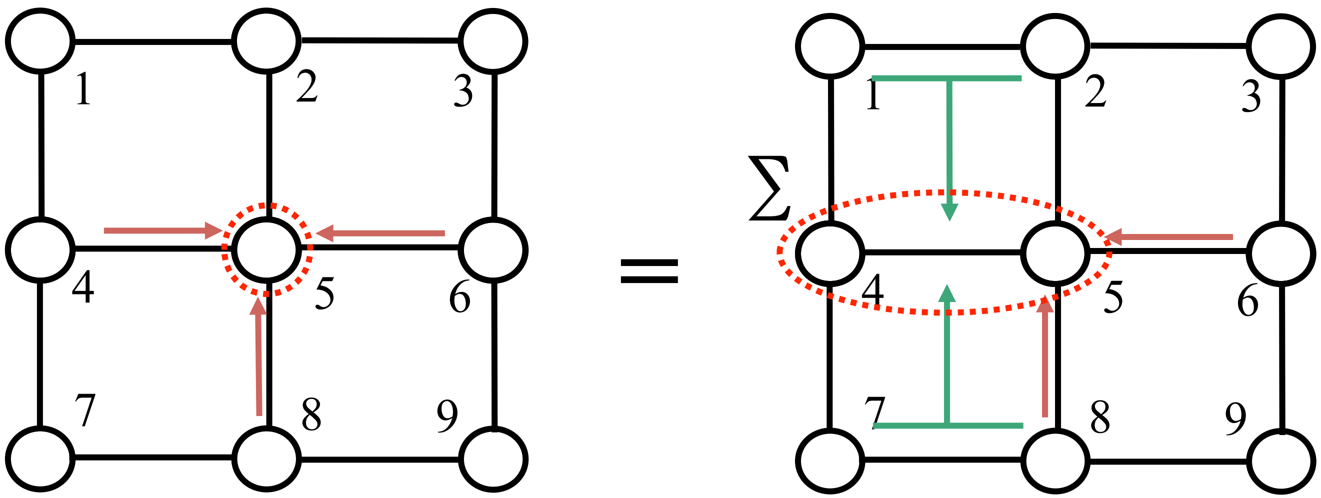
Use Marginalization Constraints to Derive Message-Update Rules



$$b_5(x_5) = \sum_{x_4} b_{45}(x_4, x_5)$$

Generalized Belief Propagation

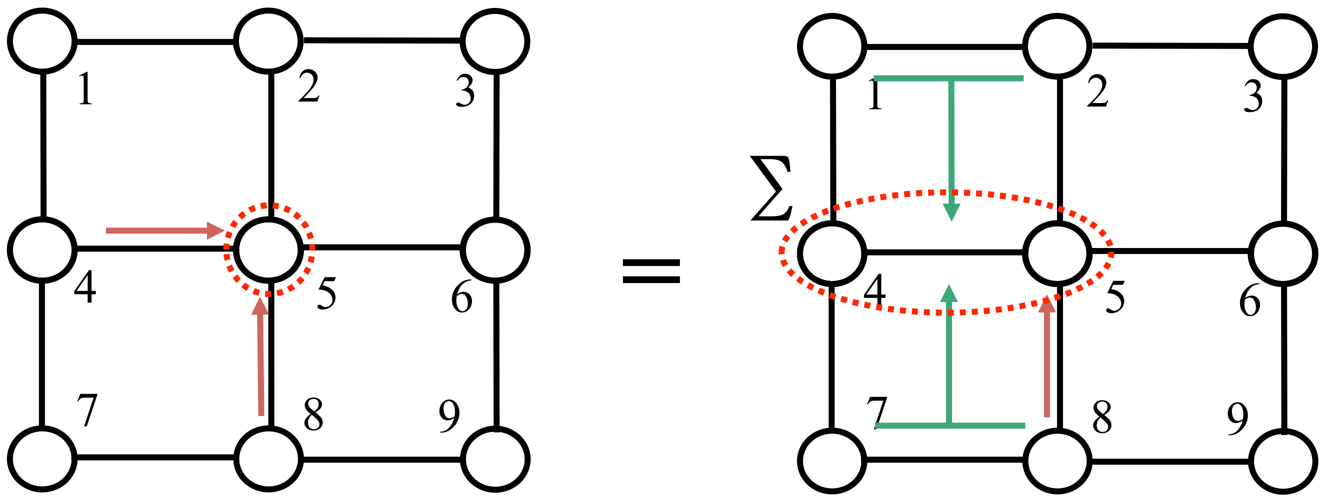
Use Marginalization Constraints to Derive Message-Update Rules



$$b_5(x_5) = \sum_{x_4} b_{45}(x_4, x_5)$$

Generalized Belief Propagation

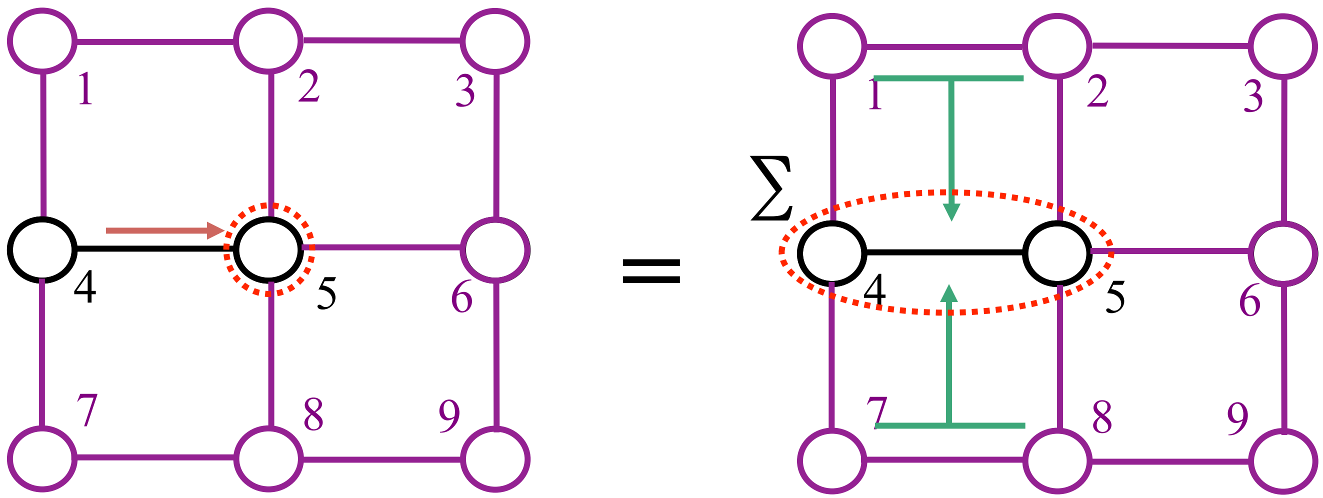
Use Marginalization Constraints to Derive Message-Update Rules



$$b_5(x_5) = \sum_{x_4} b_{45}(x_4, x_5)$$

Generalized Belief Propagation

Use Marginalization Constraints to Derive Message-Update Rules



$$m_{4\to5}(x_5) \propto \sum_{x_4} f_{45}(x_4, x_5) m_{12\to45}(x_4, x_5) m_{78\to45}(x_4, x_5)$$

GBP ALGORITHM

6) Setup equations for updating messages by enforcing marginalization conditions and combining them with the belief equations:

e.g. condition $b_5(x_5) = \sum_{x_4} b_{45}(x_4, x_5)$ yields, with the previous two bel sage update r ule

$$m_{4\rightarrow 5}(x_5) \leftarrow k \sum_{4_2} \phi_4(x_4)\psi_{45}(x_4, x_5) m_{12\rightarrow 45}(x_4, x_5) m_{78\rightarrow 25}(x_2, x_5)$$

REFERENCES

- Pearl, J. : Probabilistic reasoning in intelligent systems – Networks of plausible inference, Morgan – Kaufmann 1988
- Castillo, E., Gutierrez, J. M., Hadi, A. S. : Expert Systems and Probabilistic Network Models, Springer 1997
  - Derivations shown in class are from this book, except that we worked with π instead of ρ messages. They are related by factor of $p(e^+)$.
- www.cs.kun.nl/~peterl/teaching/CS45CI/bbn3-4.ps.gz
- Murphy, K.P., Weiss, Y., Jordan, M. : Loopy belief propagation for approximate inference – an empirical study, UAI 99
- reason.cs.uiuc.edu/eyal/classes/.../lec18-**BeliefPropagation**.ppt
- www.cs.pitt.edu/~**tomas**/cs3750/pearl.ppt