Belief Propagation

Machine Learning
CSx824/ECEx242
Bert Huang
Virginia Tech

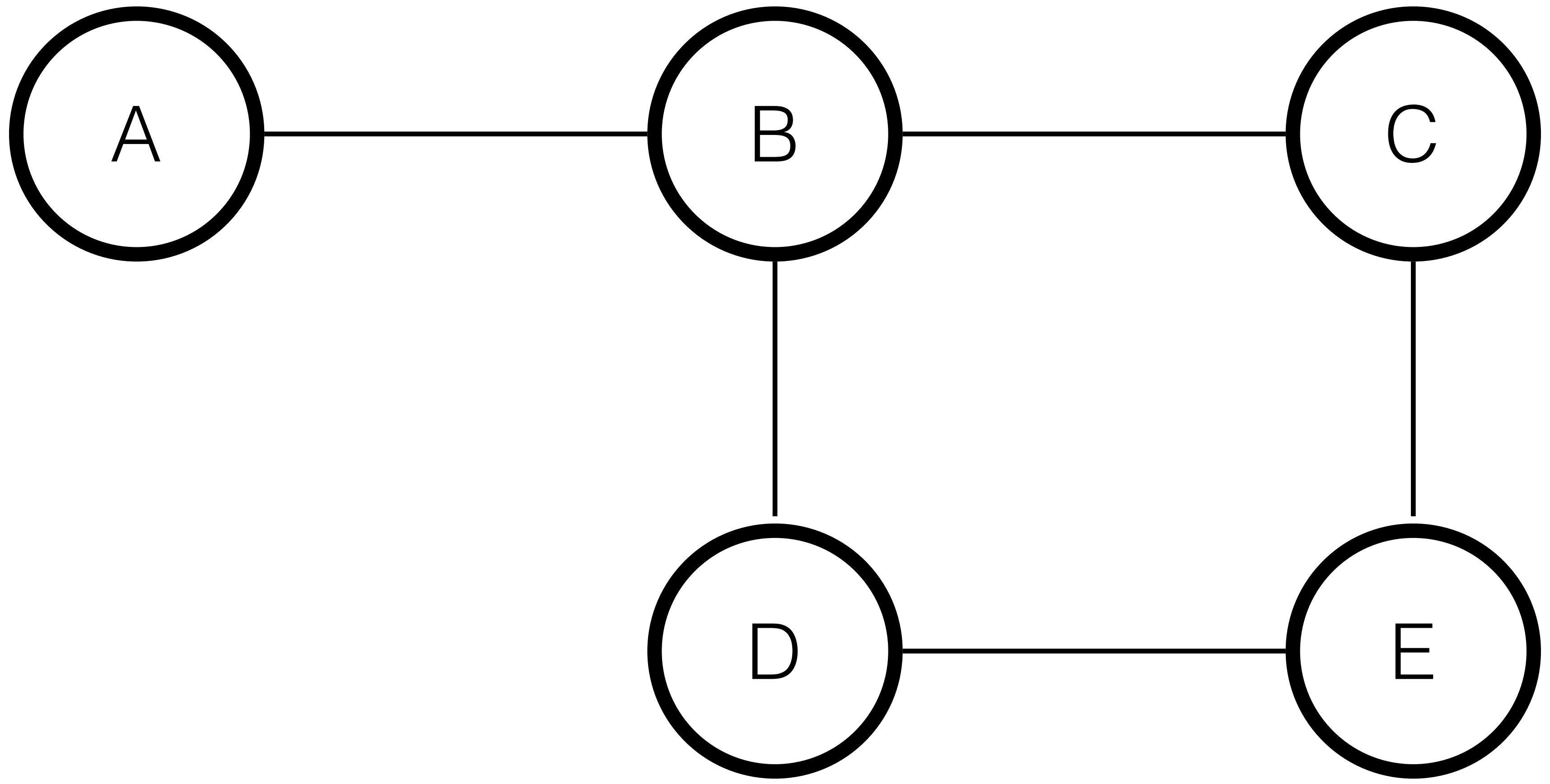
A Very Brief Sampling and Rough Introduction of Belief Propagation

Machine Learning
CSx824/ECEx242
Bert Huang
Virginia Tech

Outline

- Review: Markov random fields
- Sum-product belief propagation (BP)
- BP in trees
- BP in HMMs

Markov Random Fields



$$P(A, B, C, D, E) \propto \phi(A, B)\phi(B, C)\phi(B, D)\phi(C, E)\phi(D, E)$$

$$P(X) = \frac{1}{Z} \prod_{c \in \text{cliques}(G)} \phi_C(x_c)$$

potential functions

Inference in MRFs

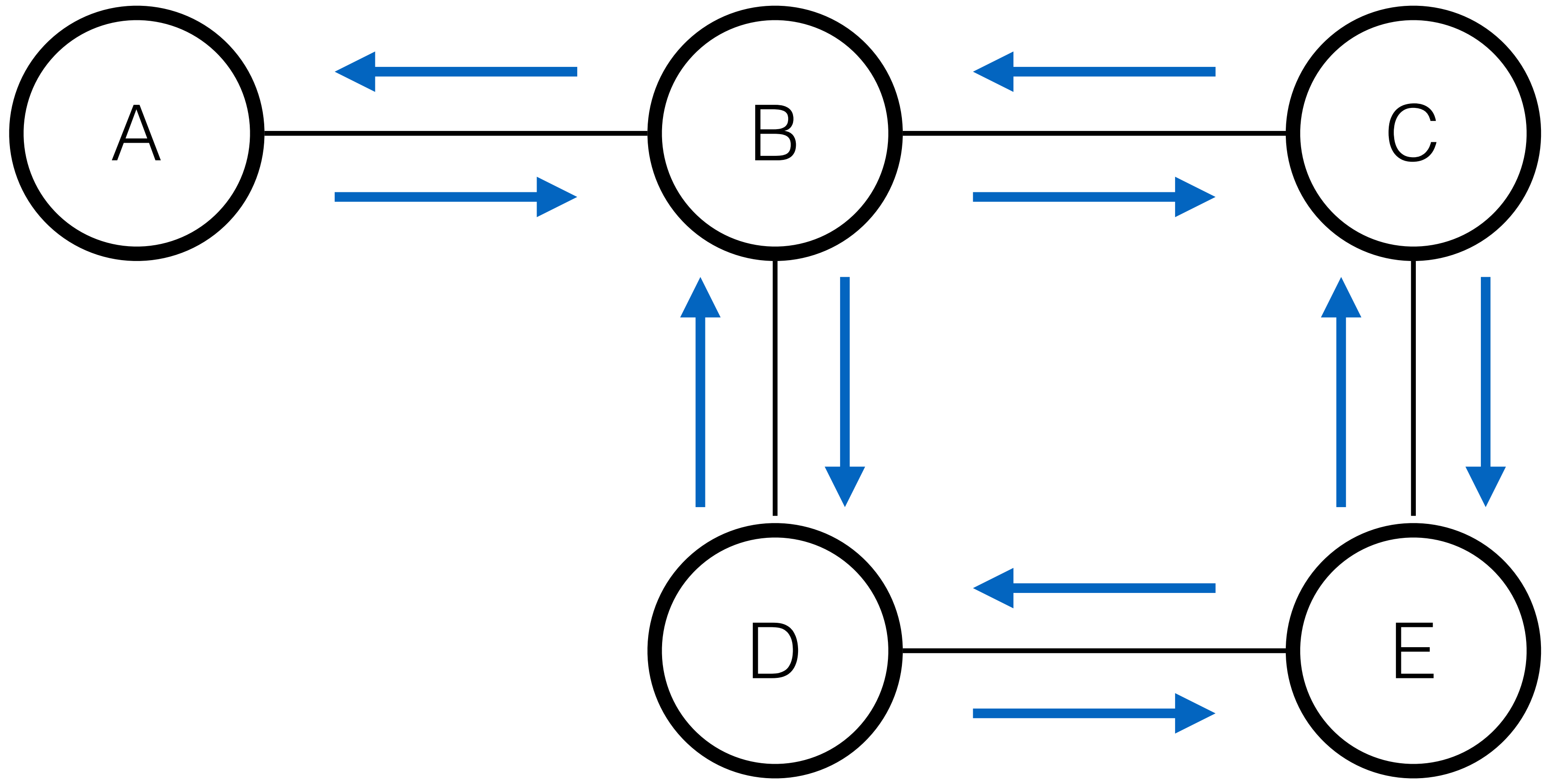
$$P(X) = \frac{1}{Z} \prod_{c \in \text{cliques}(G)} \phi_C(x_c) \qquad p(X_s = x_s) = \sum_{x : X_s = x_s} \frac{1}{Z_s} \prod_{c \in \text{cliques}(G)} \phi_c(x_c)$$

$$P(A, B, C, D, E) \propto \phi(A, B)\phi(B, C)\phi(B, D)\phi(C, E)\phi(D, E)$$

$$p(A = a) \propto \sum_{b,c,d,e} \phi(a, b)\phi(b, c)\phi(b, d)\phi(c, e)\phi(d, e)$$

$$\phi(a, b) := \phi(A = a, B = b)$$

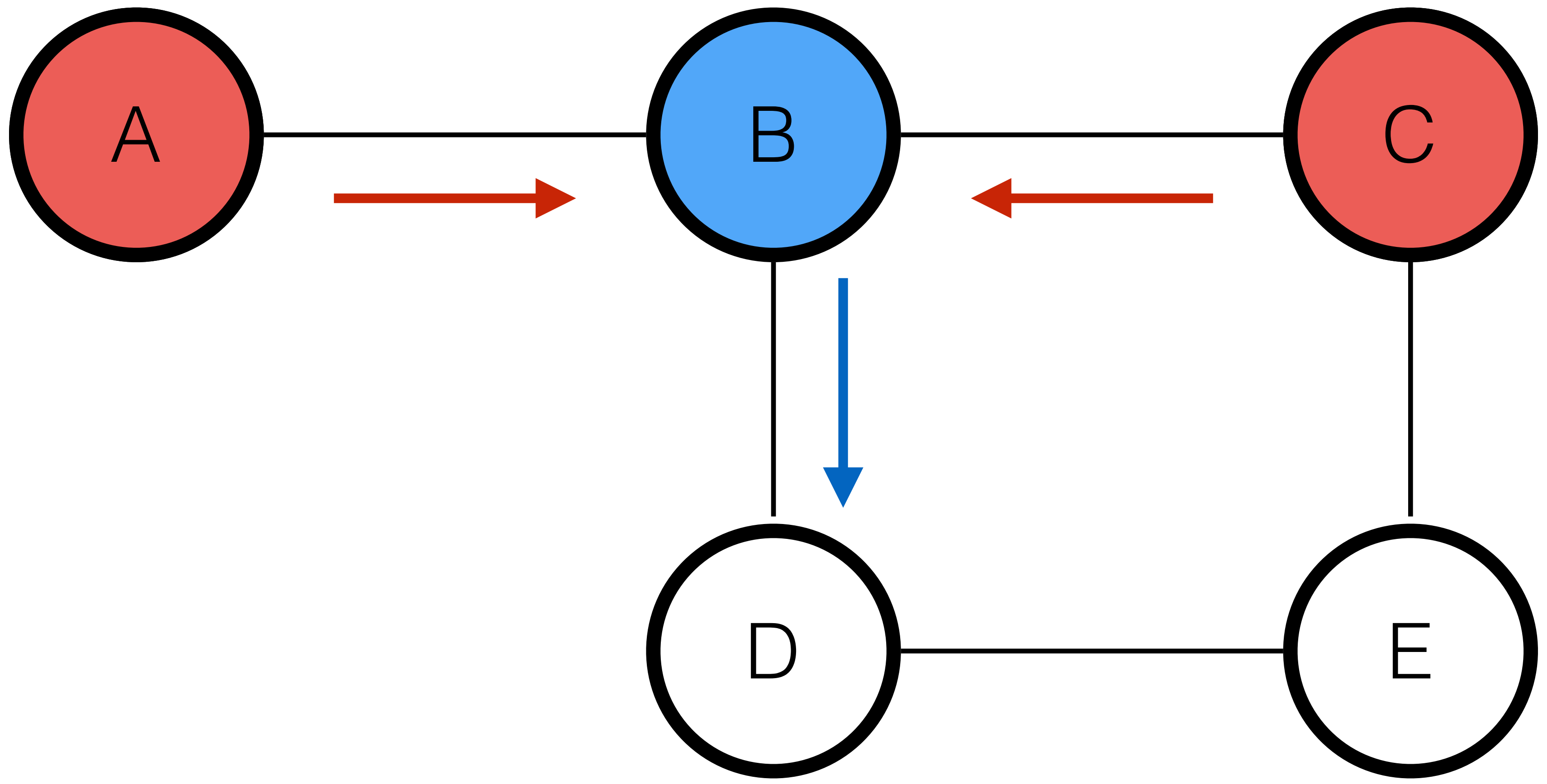
Sum-Product Belief Propagation



$$b_t(x_t) \propto \prod_{s \in \text{neighbors}(t)} m_{t \to s}(x_s)$$

$$m_{s \to t}(x_t) := \sum_{x_s} \left(\phi_{st}(x_s, x_t) \prod_{u \in \text{neighbors}(s) \setminus t} m_{u \to s}(x_s) \right)$$

Sum-Product Belief Propagation

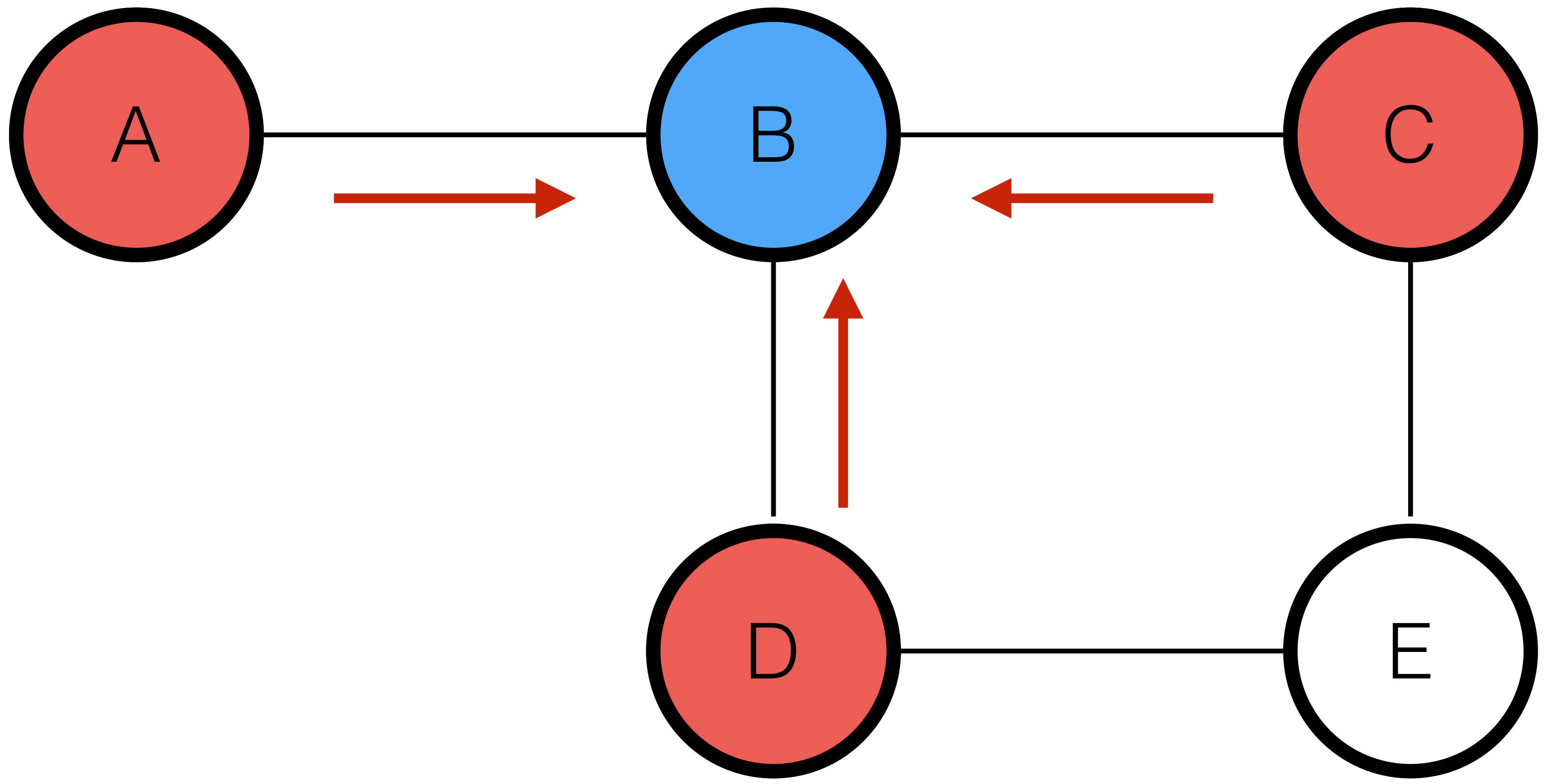


$$b_t(x_t) \propto \prod_{s \in \text{neighbors}(t)} m_{t \to s}(x_s)$$

$$m_{s \to t}(x_t) := \sum_{x_s} \left(\phi_{st}(x_s, x_t) \prod_{u \in \text{neighbors}(s) \setminus t} m_{u \to s}(x_s) \right)$$

$$m_{B \to D}(x_D) = \sum_{x_B} \phi(x_B, x_C) \times m_{A \to B}(x_B) \times m_{C \to B}(x_B)$$

Sum-Product Belief Propagation



$$b_t(x_t) \propto \prod_{s \in \text{neighbors}(t)} m_{t \to s}(x_s)$$

$$b_B(x_B) \propto (m_{A \to B}(x_B))(m_{C \to B}(x_B))(m_{D \to B}(x_B))$$

$$m_{s \to t}(x_t) := \sum_{x_s} \left(\phi_{st}(x_s, x_t) \prod_{u \in \text{neighbors}(s) \setminus t} m_{u \to s}(x_s) \right)$$

$$m_{B \to D}(x_D) = \sum_{x_B} \phi(x_B, x_C) \times m_{A \to B}(x_B) \times m_{C \to B}(x_B)$$

Sum-Product Messages

- Functions of receiving node's variable
- Vectors for discrete variables
- Can (should) be normalized
- Alternate form explicitly encodes **unary** and **edge** potentials:

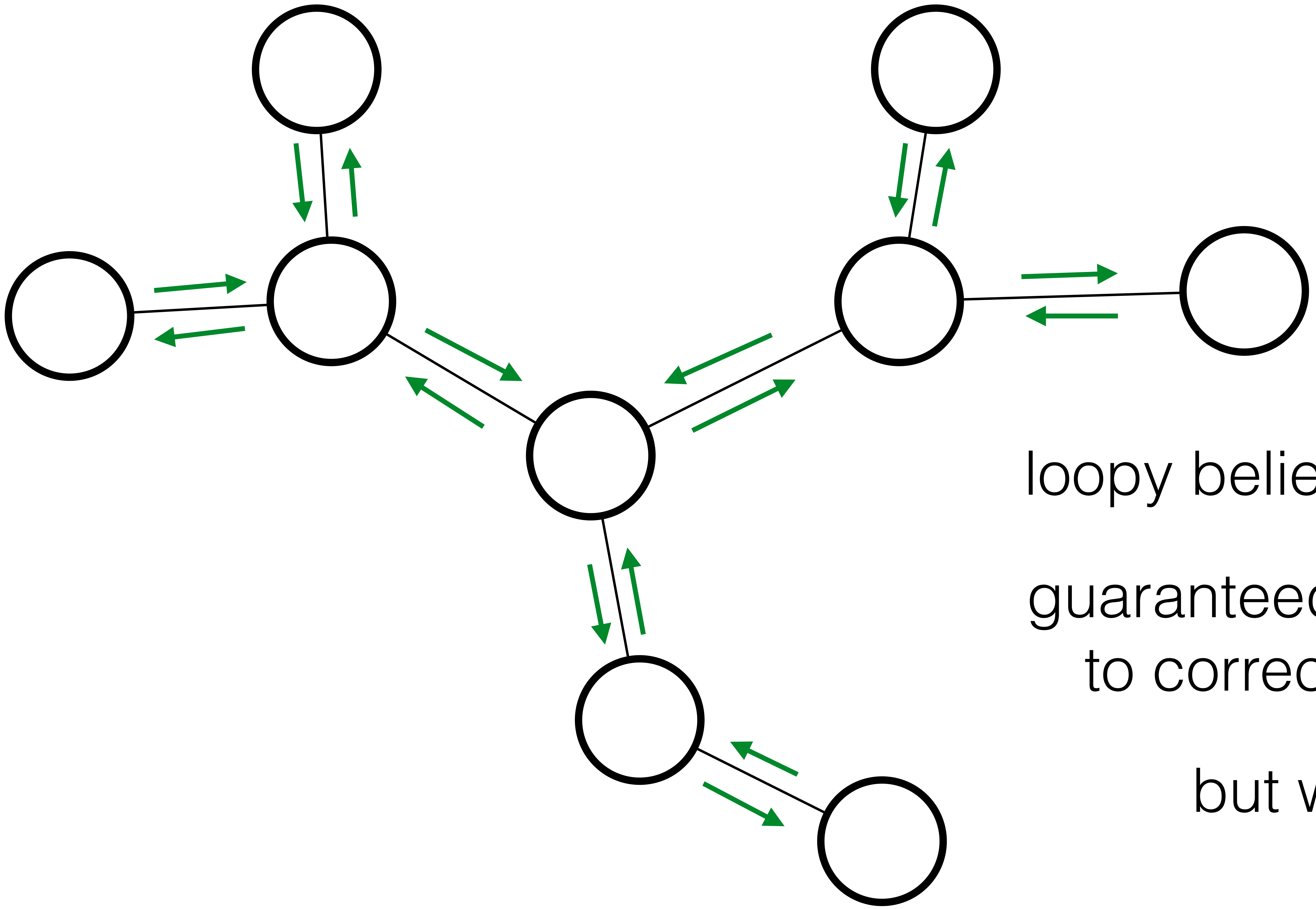
$$p(\mathbf{x}) \propto \prod_{s \in \mathcal{V}} \phi_s(x_s) \prod_{(s,t) \in \mathcal{E}} \phi_{st}(x_s, x_t)$$

previous form absorbs unary potentials into edge potentials

$$b_t(x_t) \propto \phi_t(x_t) \prod_{s \in \text{neighbors}(t)} m_{t \to s}(x_s)$$

$$m_{s \to t}(x_t) := \sum_{x_s} \left(\phi_s(x_s) \phi_{st}(x_s, x_t) \prod_{u \in \text{neighbors}(s) \setminus t} m_{u \to s}(x_s) \right)$$

Scheduling Message Updates

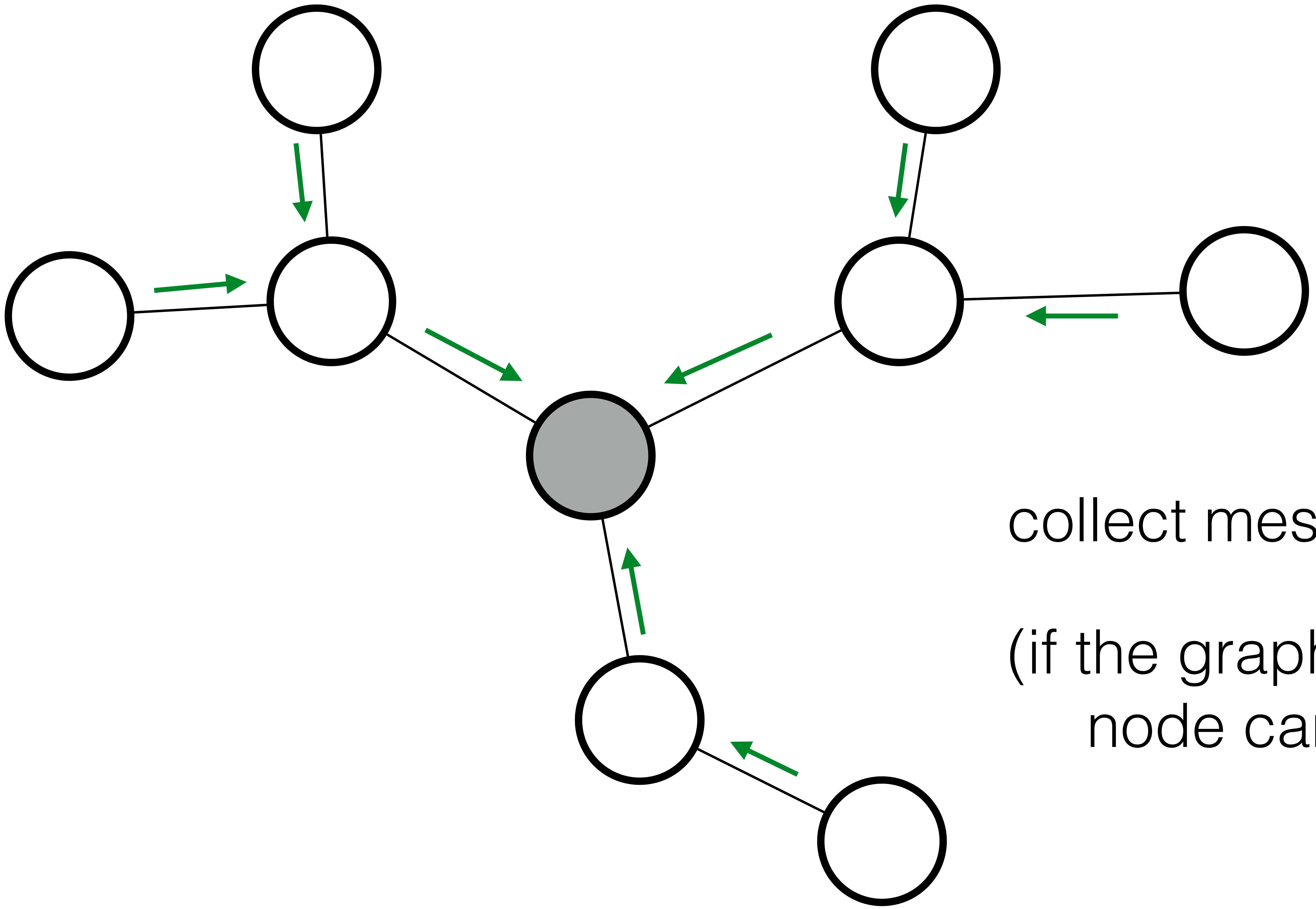


loopy belief propagation

guaranteed to converge
to correct marginals

but wasteful

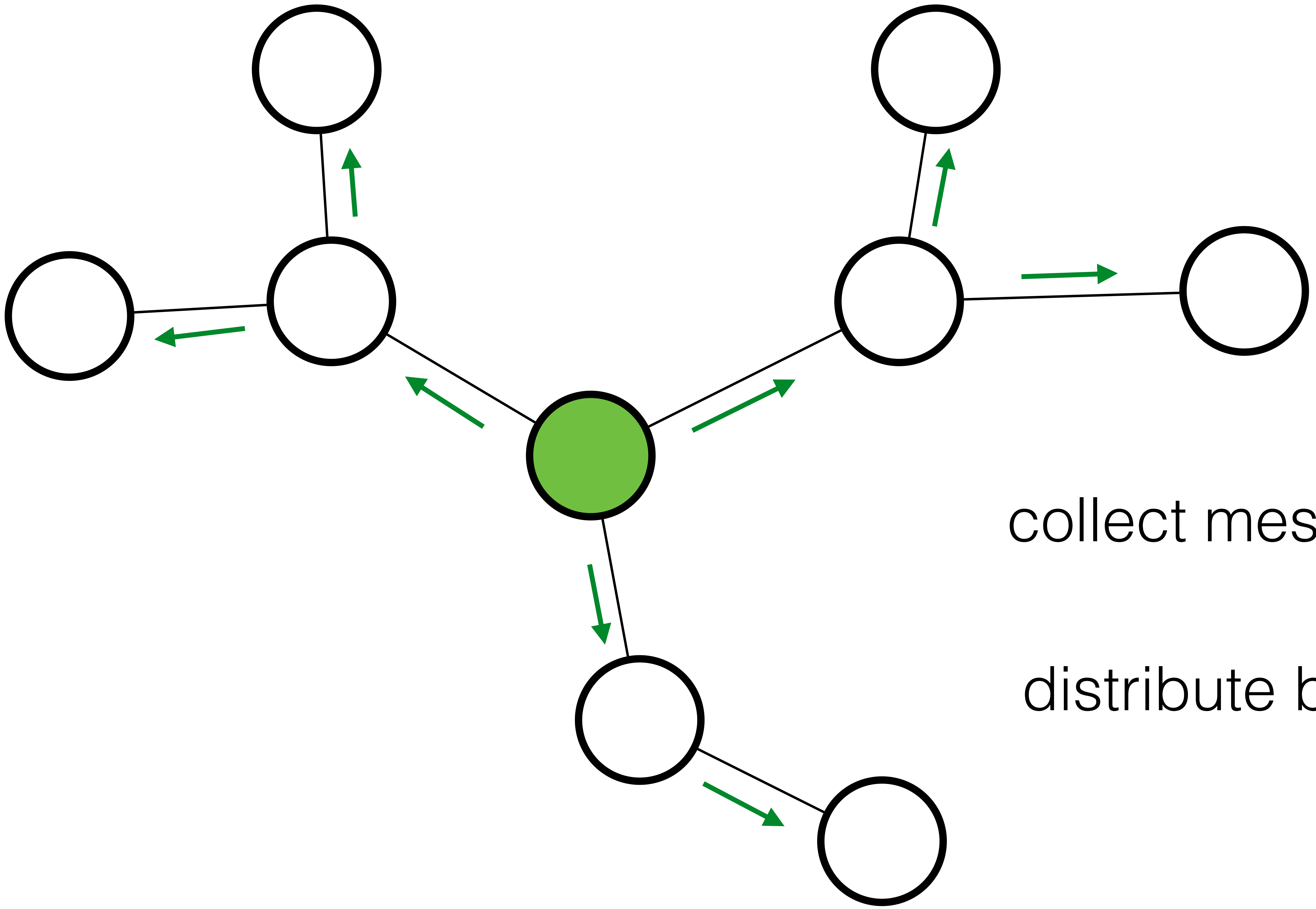
Scheduling Message Updates



collect messages to root

(if the graph is a tree, any node can be a root)

Scheduling Message Updates



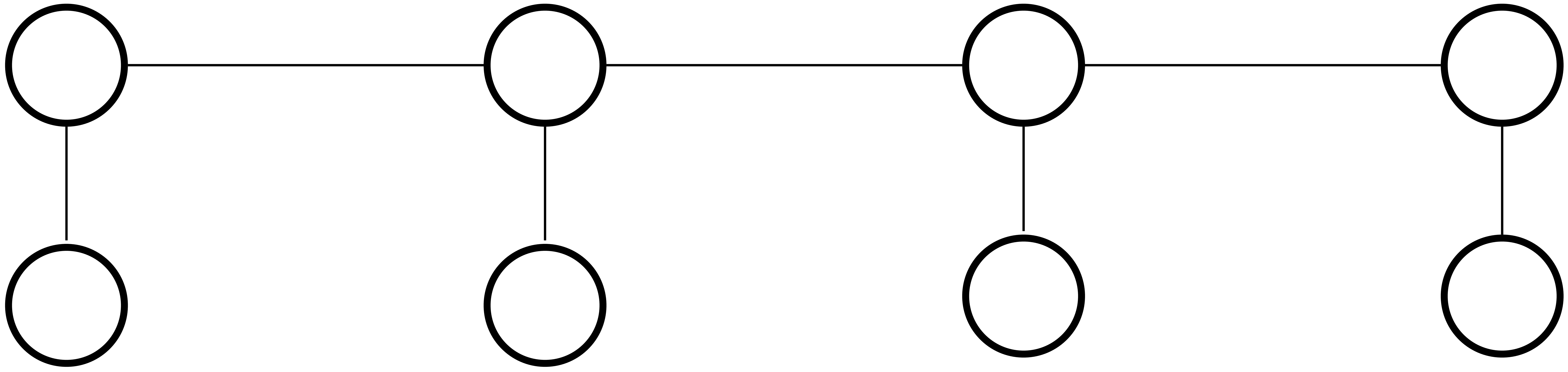
collect messages to root

distribute back to leaves

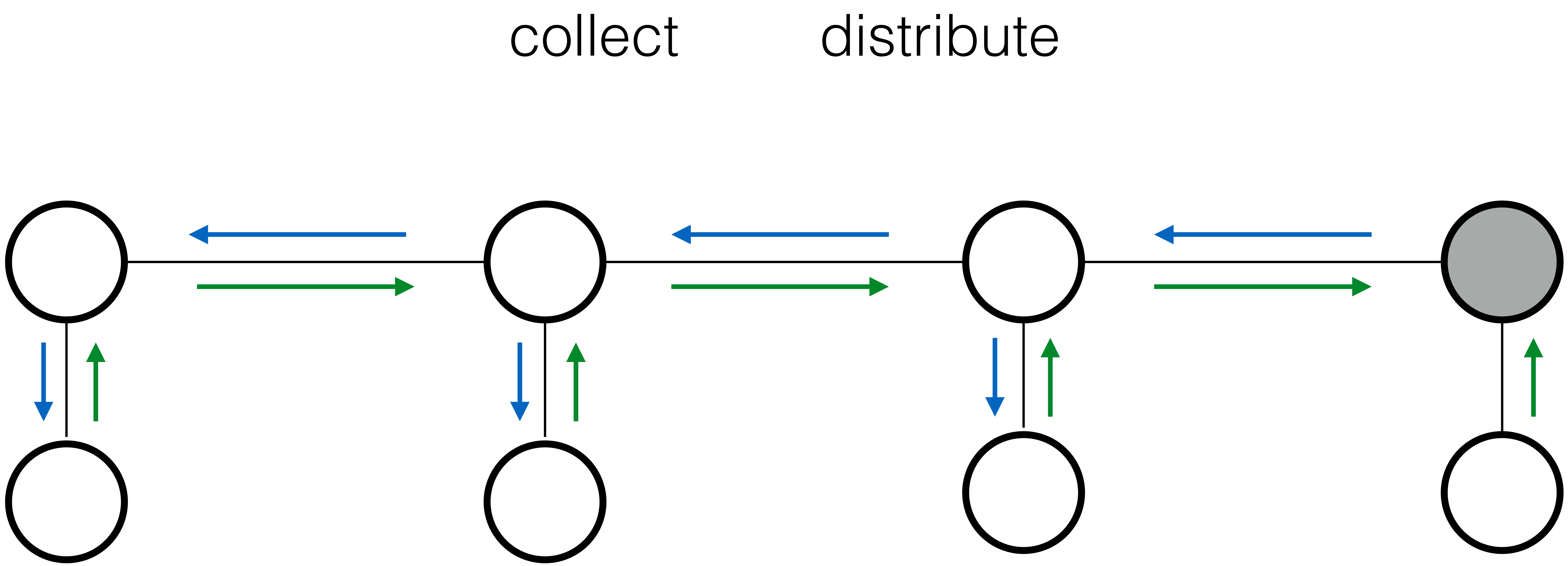
BP in Trees

- Guaranteed to converge to marginals
- Collect-distribute “converges” with just one update per message
- Guarantees not as nice in non-trees
    - rule-of-thumb: treat loopy BP as approximate inference

Belief Propagation in an HMM



Belief Propagation in an HMM



Summary and Notes

- Belief propagation passes messages between neighboring nodes
- Fuse incoming messages from all neighbors **except receiving node**
- Collect evidence and distribute (serial order) in trees: exact marginals
- Iterate in loopy graphs: approximate marginals
- Messages are related to Lagrange multipliers for enforcing **consistency constraints** on estimated marginals (lots of interesting research on this relationship)