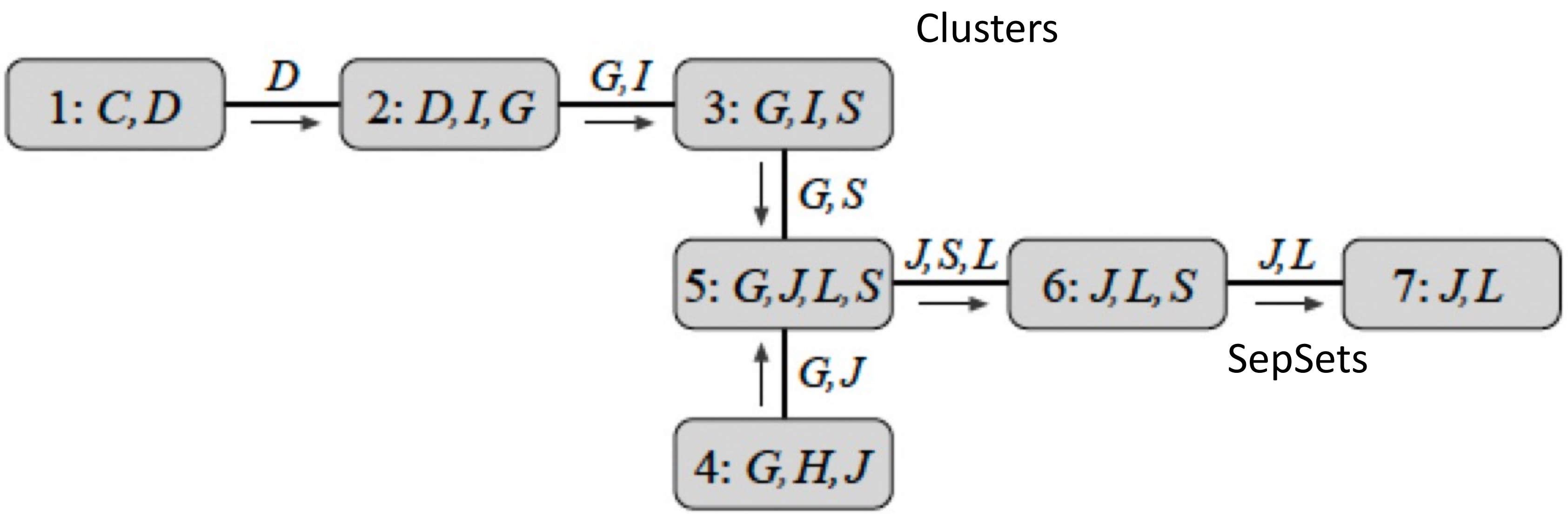
Probabilistic Graphical Models

Belief Propagation

Variable Elimination

Step	Variable eliminated	Factors used	Variables involved	New factor
1	C	$\phi_C(C), \phi_D(D, C)$	D, C	$\tau_1(D)$
2	D	$\tau_1(D), \phi_G(G, I, D)$	D, I, G	$\tau_2(G, I)$
3	I	$\tau_2(G, I), \phi_I(I), \phi_S(S, I)$	G, I, S	$\tau_3(G, S)$
4	H	$\phi_H(H, G, J)$	G, H, S	$\tau_4(G, J)$
5	G	$\tau_3(G, S), \tau_4(G, J)\phi_L(L, G)$	G, J, L, S	$\tau_5\ (J, L, S)$
6	S	$\tau_5\ (J, L, S)\ \phi_J(J, L, S)$	J, L, S	$\tau_6\ (J, L)$
7	L	$\tau_6\ (J, S)$	J, L	$\tau_7\ (J)$

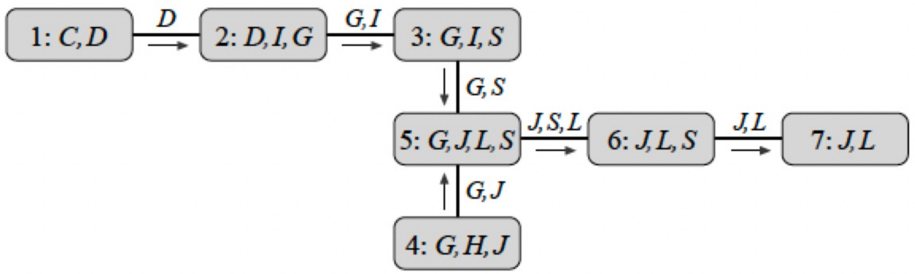
Cluster Trees



Variable Elimination as message passing

Message passing

$$\delta_{12}(D) = \sum_{D} \phi(C)\phi(C,D)$$



$$\delta_{23}(G,I) = \sum_{D} \phi_G(D,I,G)\delta_{12}(D)$$

$$\delta_{35}(G,S) = \sum_{I} \phi_I(I), \phi_S(S,I)\delta_{35}(G,I)$$

$$\delta_{67}(J,L) = \sum_{S} \delta_{56}(J,S,L)$$

$$\delta_{45}(G,J) = \sum_{H} \phi(G,H,J)$$

$$\delta_{56}(J,S,L) = \sum_{G} \delta_{45}(G,J)\delta_{35}(G,S)\phi(L,G)$$

Clique-Tree Message Passing

1. Pick a node to be your root.
2. For each node i, initialize the potential of the node

$$\psi_i = \prod_i \phi_i$$

3. Start from a leaf and send message to all neighbors

$$\delta_{i\to j} = \sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \psi_i \cdot \prod_{k\in(\mathrm{Nb}_i - \{j\})} \delta_{k\to i}$$

4. Repeat for every node that is ready to transmit a message (i.e., has received messages from every neighbor)

Properties of Cluster Trees

Family Preservation:

For each factor $\phi_k \in \Phi$, there exists a cluster C_i s.t. Scope $[\phi_k] \subseteq \boldsymbol{C}_i$

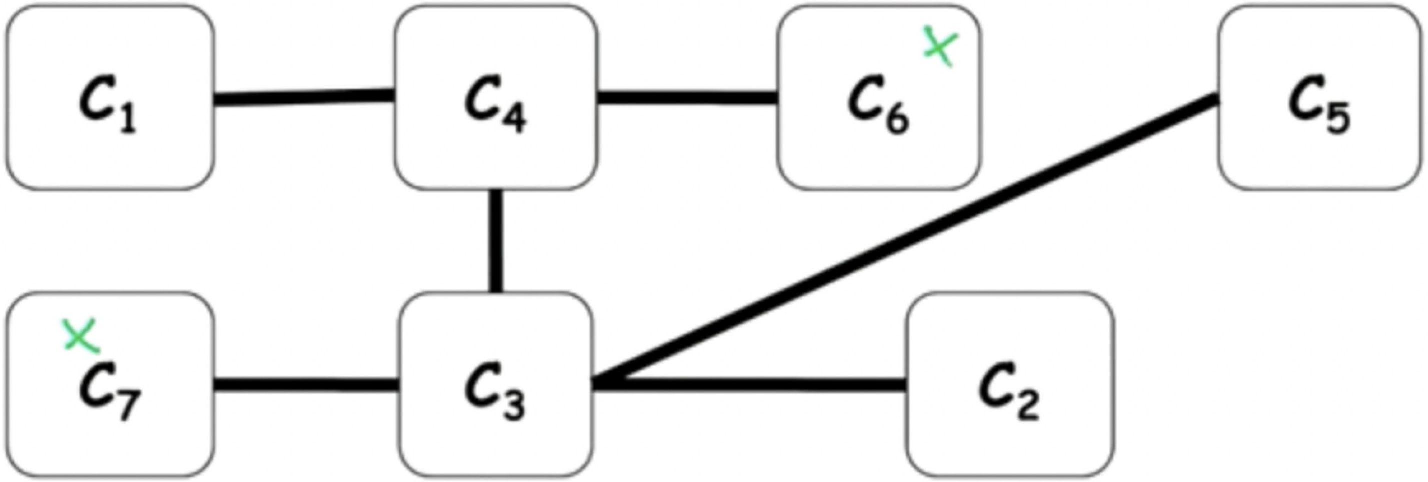
every factor has a node that can accommodate it

Running Intersection:

For each pair of clusters $C_{\mathrm{i}}, C_{\mathrm{j}}$ and variable $X \in C_i \cap C_j$, in the unique path between C_i and C_j, all clusters and sepsets contain X.

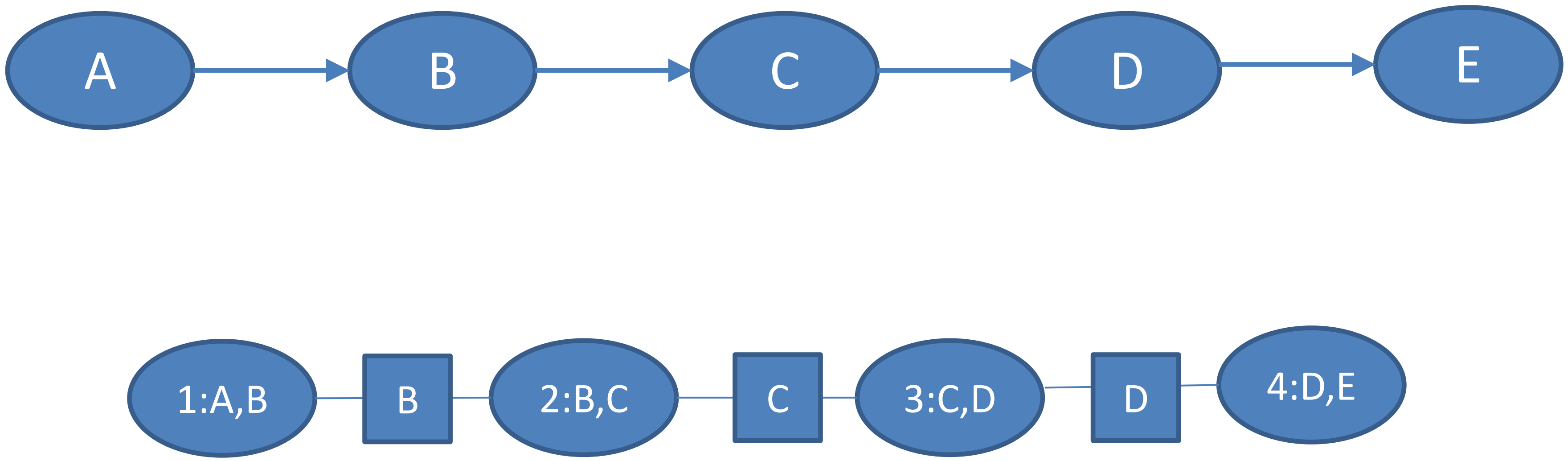
clusters that include the same variable need to communicate for consistency

Running Intersection

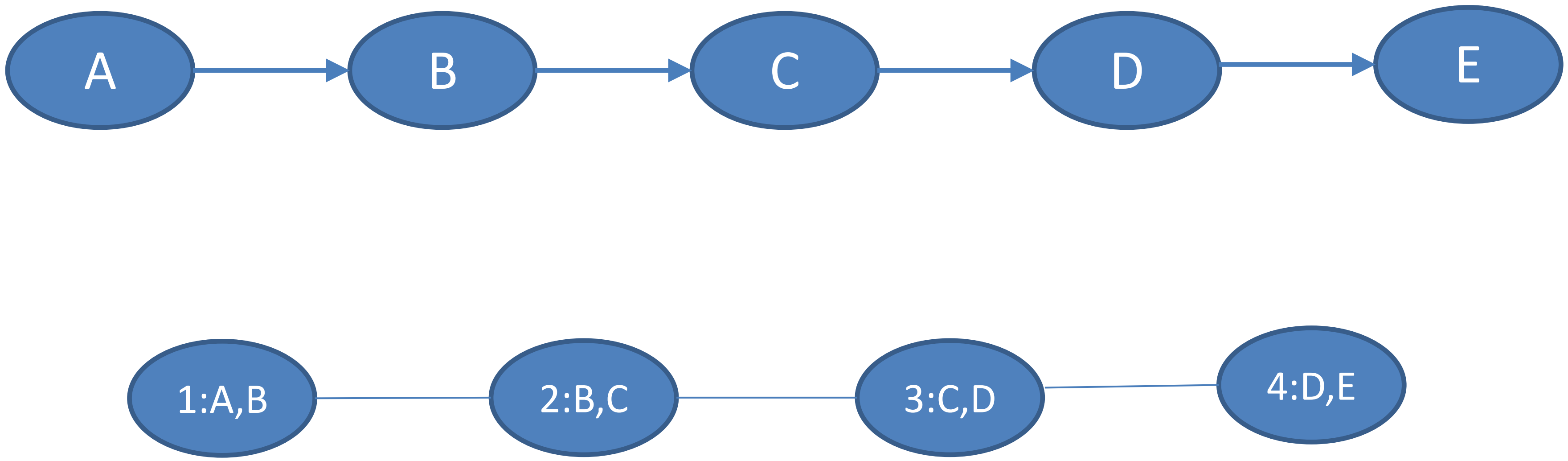


Which clusters need to include X?

Cluster Tree Inference

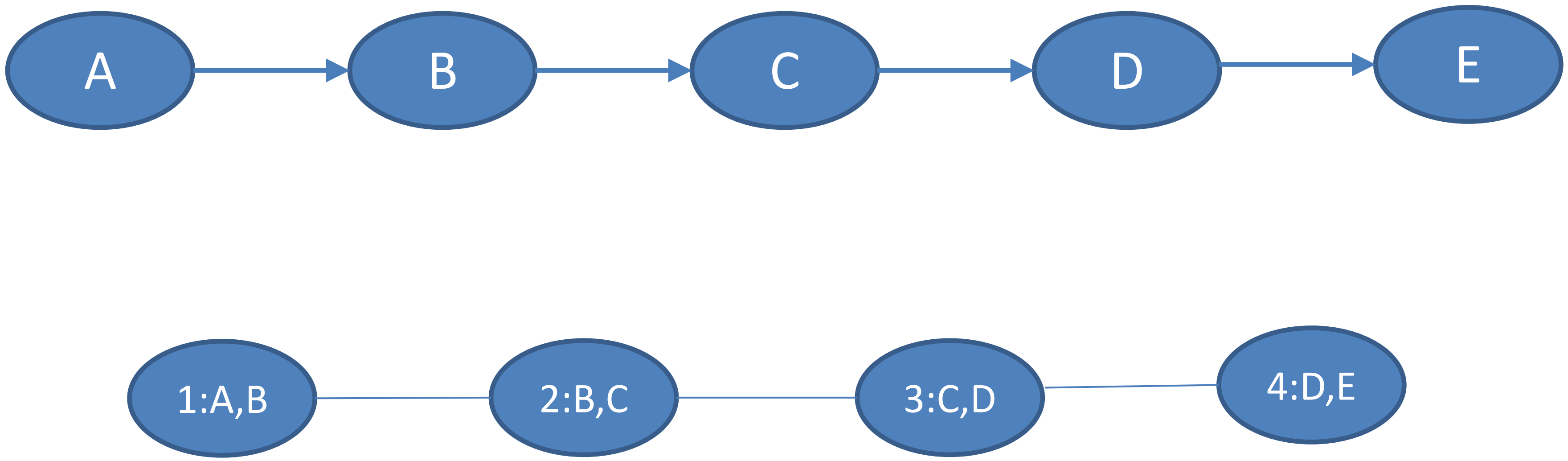


Cluster Tree Inference



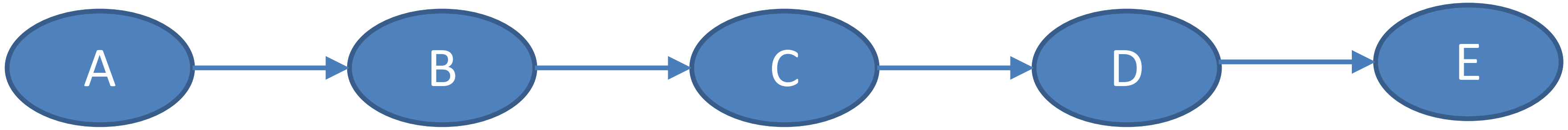
Compute P(E)

Cluster Tree Inference



Compute P(C)

Correctness



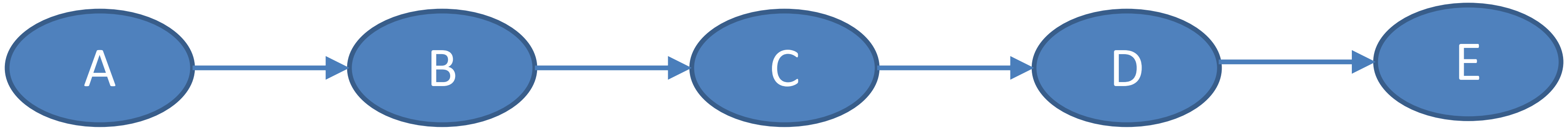
Compute P(C)

$$\beta_3(C,D) = \psi_3 \times \delta_{2\to3} \times \delta_{4\to3} =$$

$$\psi_3 \times \left(\sum_B \psi_2 \delta_{2\to1} \right) \times \sum_E \psi_4 =$$

$$\psi_3 \times (\sum_B \psi_2 \sum_A \psi_1) \times \sum_E \psi_4$$

Correctness



Compute P(C)

VE is correct as long as:
When X is eliminated,
all factors with X in their
scope have been
multiplied in.

$$\beta_3(C,D) = \psi_3 \times \delta_{2\to3} \times \delta_{4\to3} =$$

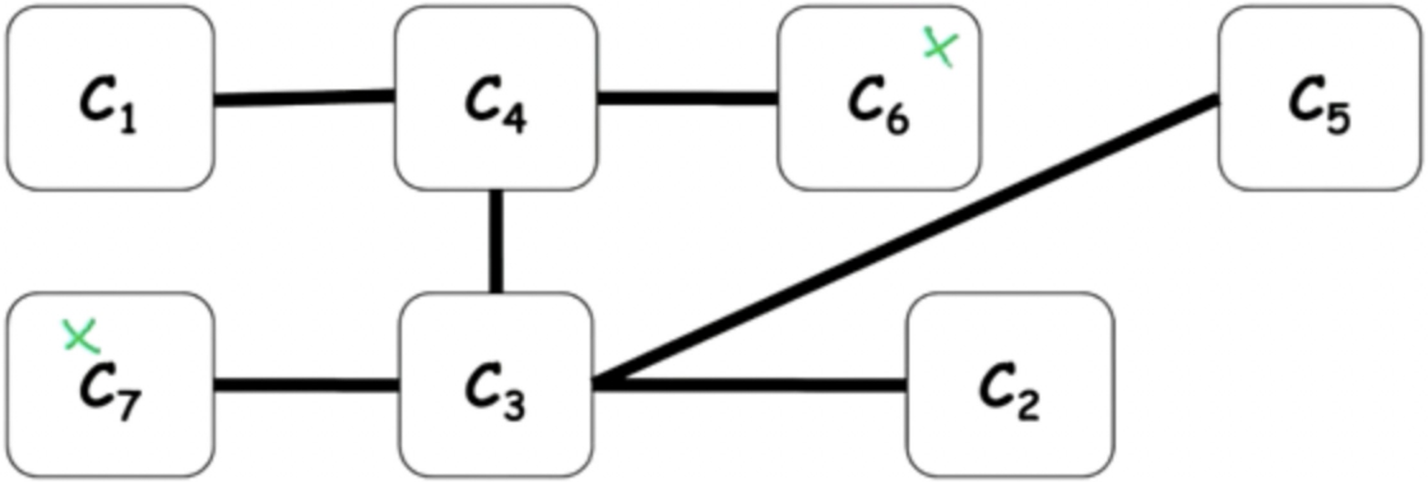
$$\psi_3 \times \left(\sum_B \psi_2 \delta_{2\to1}\right) \times \sum_E \psi_4 =$$

$$\psi_3 \times (\sum_B \psi_2 \sum_A \psi_1) \times \sum_E \psi_4$$

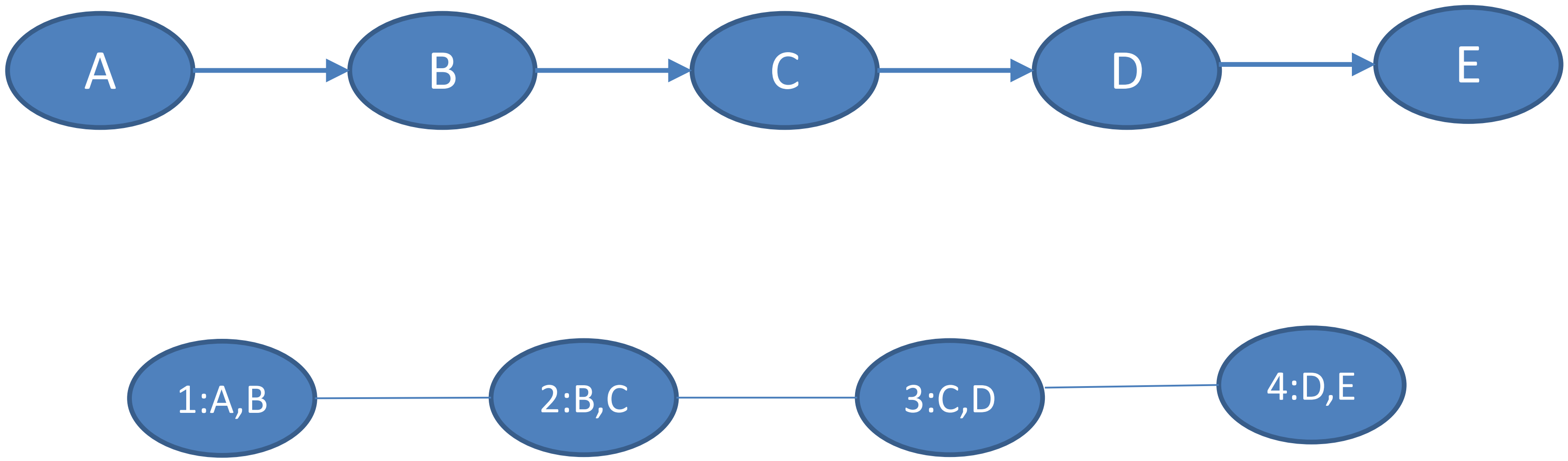
Correctness

Running Intersection $\Rightarrow$ Correctness

- If X is eliminated when we pass the message $C_i \rightarrow C_j$
- Then X does not appear in the C_j side of the tree



Cluster Tree Inference



Compute P(D)

Calibrating Cluster Trees

Perform one forward and one backward pass of the tree

Store all messages

You can compute any marginal distribution

Calibrating Cluster Trees

Algorithm 10.2 Calibration using sum-product message passing in a clique tree

```
Procedure CTree-SP-Calibrate (
    Φ,    // Set of factors
    T     // Clique tree over Φ
  )
  Initialize-Cliques
  while exist i, j such that i is ready to transmit to j
    δ_{i→j}(S_{i,j}) ← SP-Message(i, j)
  for each clique i
    β_i ← ψ_i · ∏_{k∈Nb_i} δ_{k→i}
  return {β_i}
```

Two adjacent cliques C_i *and* C_j *are said to be* calibrated *if*

$$\sum_{C_i - S_{i,j}} \beta_i(C_i) = \sum_{C_j - S_{i,j}} \beta_j(C_j).$$

Types of Queries

- Posterior distribution queries on variables that appear together in clique
- Sum out irrelevant variables from any clique containing those variables and renormalize
- Introducing new evidence $Z = z$ and querying
  - If X appears in clique with Z, multiply clique that contains X and Z with indicator function $1(Z = z)$ (reduction)
  - Sum out irrelevant variables and renormalize

Types of Queries

Introducing new evidence $Z = z$ and querying X if X does not share a clique with Z

- Multiply $1(Z = z)$ into some clique containing Z (reduction)

- Propagate messages along path to clique containing X

Belief propagation is correct for trees

Thm: The Message Passage guarantees obtaining all marginals in the tree

$$m_{ji}(x_i) = \sum_{x_j} \left(\psi(x_j)\psi(x_i, x_j) \prod_{k \in N(j) \setminus i} m_{kj}(x_j) \right)$$

How about non-trees?

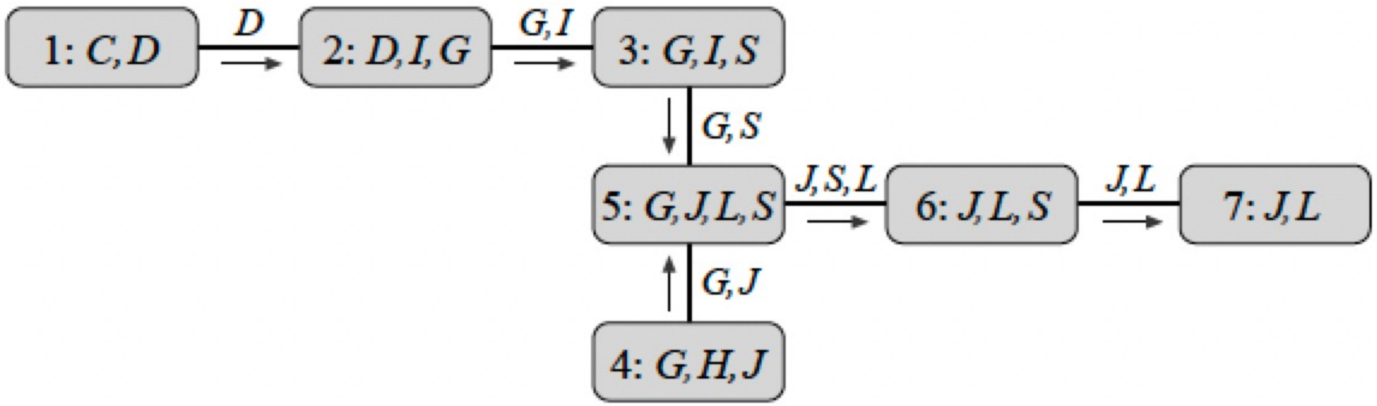
Running Intersection & Independence

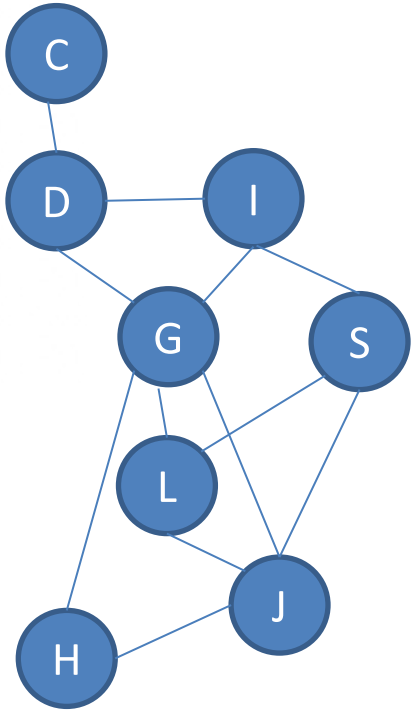
For an edge (i, j) in T, let:
$-W_{<\text{(i,j)}}$ = all variables that appear only on C_i side of T
$-W_{<\text{(j,i)}}$ = all variables that appear only on C_{j} side

- Variables on both sides are in the sepset $s_{i,j}$

- Theorem: T satisfies RIP if and only if, for every (i, j)
- $P_\Phi \models \left(\boldsymbol{W}_{<(i,j)} \perp \boldsymbol{W}_{<(j,i)} \mid \boldsymbol{S}_{i,j}\right)$

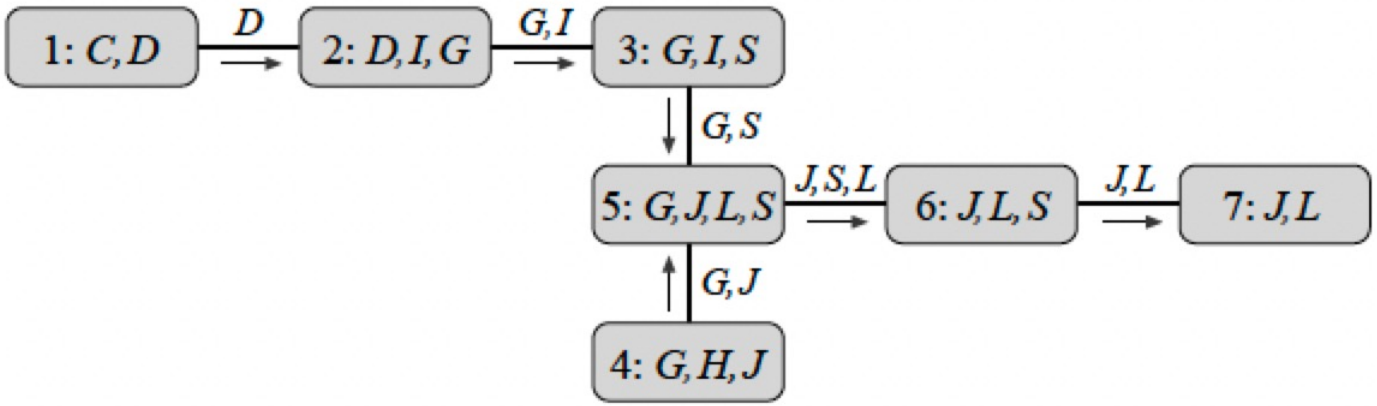
Separation

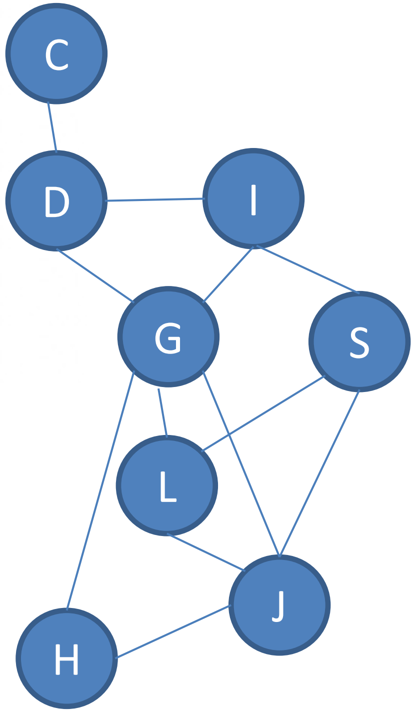




- $P_\Phi \vDash \left(\boldsymbol{W}_{<(i,j)} \perp \boldsymbol{W}_{<(j,i)} \mid \boldsymbol{S}_{i,j}\right)$

Separation

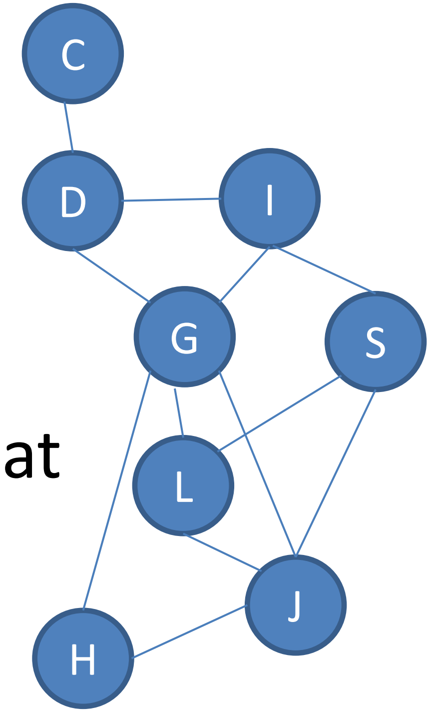




- $P_\Phi \models \left(\boldsymbol{W}_{<(i,j)} \perp \boldsymbol{W}_{<(j,i)} \mid \boldsymbol{S}_{i,j}\right)$
- $P_\Phi \models (C \perp G, I, S, L, J, H \mid D)$
- $P_\Phi \models (C, D \perp S, L, J, H \mid G, I)$
- $P_\Phi \models (C, D, I \perp L, J, H \mid G, S)$

Theorem

- Theorem: T satisfies RIP if and only if, for every (i, j)
- $P_\Phi \vDash \left(\boldsymbol{W}_{<(i,j)} \perp \boldsymbol{W}_{<(j,i)} \mid \boldsymbol{S}_{i,j}\right)$



Proof sketch (⇒)

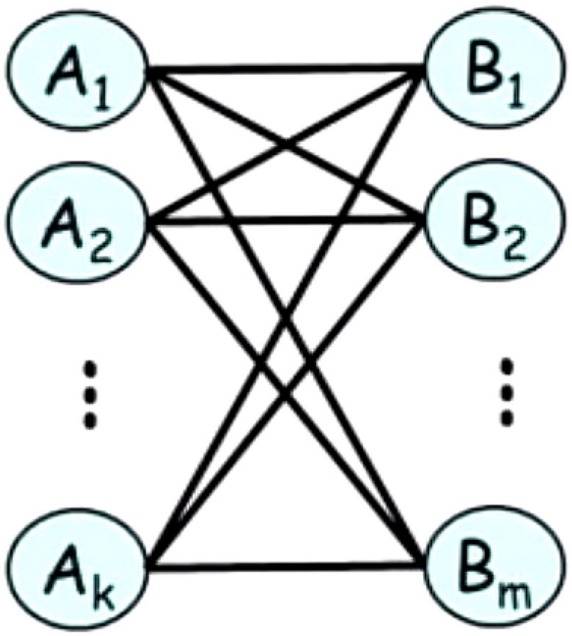
∃ path in G_I between V ∈ $\boldsymbol{W}_{<(i,j)}$ and X ∈ $\boldsymbol{W}_{<(j,i)}$ that does not go through $S_{i,j}$

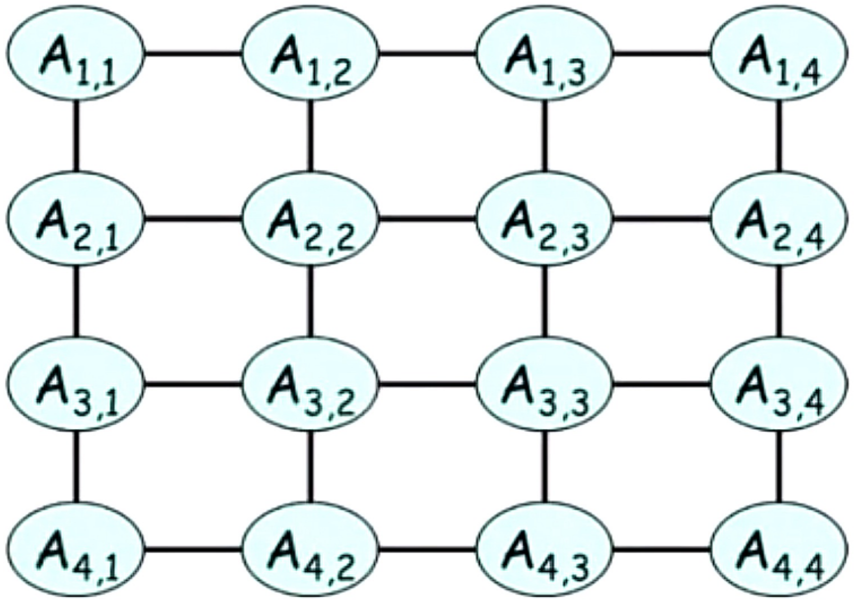
∃ factor in the induced graph that includes V, X
If the factor is on the $\boldsymbol{W}_{<(j,i)}$ side, it should be in every intermediate $S_{i,j}$ otherwise RIP does not hold

Implications

Each sepset needs to separate graph into two conditionally independent parts

Implies minimal complexity





Summary

Belief propagation on trees generates beliefs that are guaranteed to be correct marginals

In clique tree with K cliques, if messages are passed starting at leaves 2*(K-1) messages suffice to compute all beliefs

Can compute marginals over all variables at only twice the cost of variable elimination

Summary (2)

By storing messages, inference can be reused in incremental queries

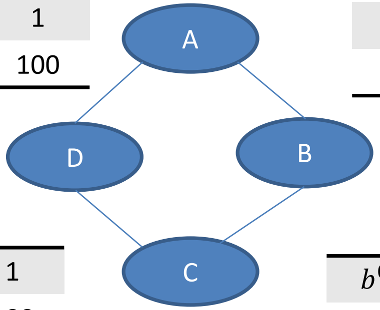
Correctness of clique tree inference relies on running intersection property

Running intersection property implies separation in original distribution
-Implies minimal complexity

Example: Misconception

d^0	α^0	100
d^0	a^1	1
d^1	a^0	1
d^1	a^1	100

α^0	b^0	30
α^0	b^1	5
α^1	b^0	1
α^1	b^1	100



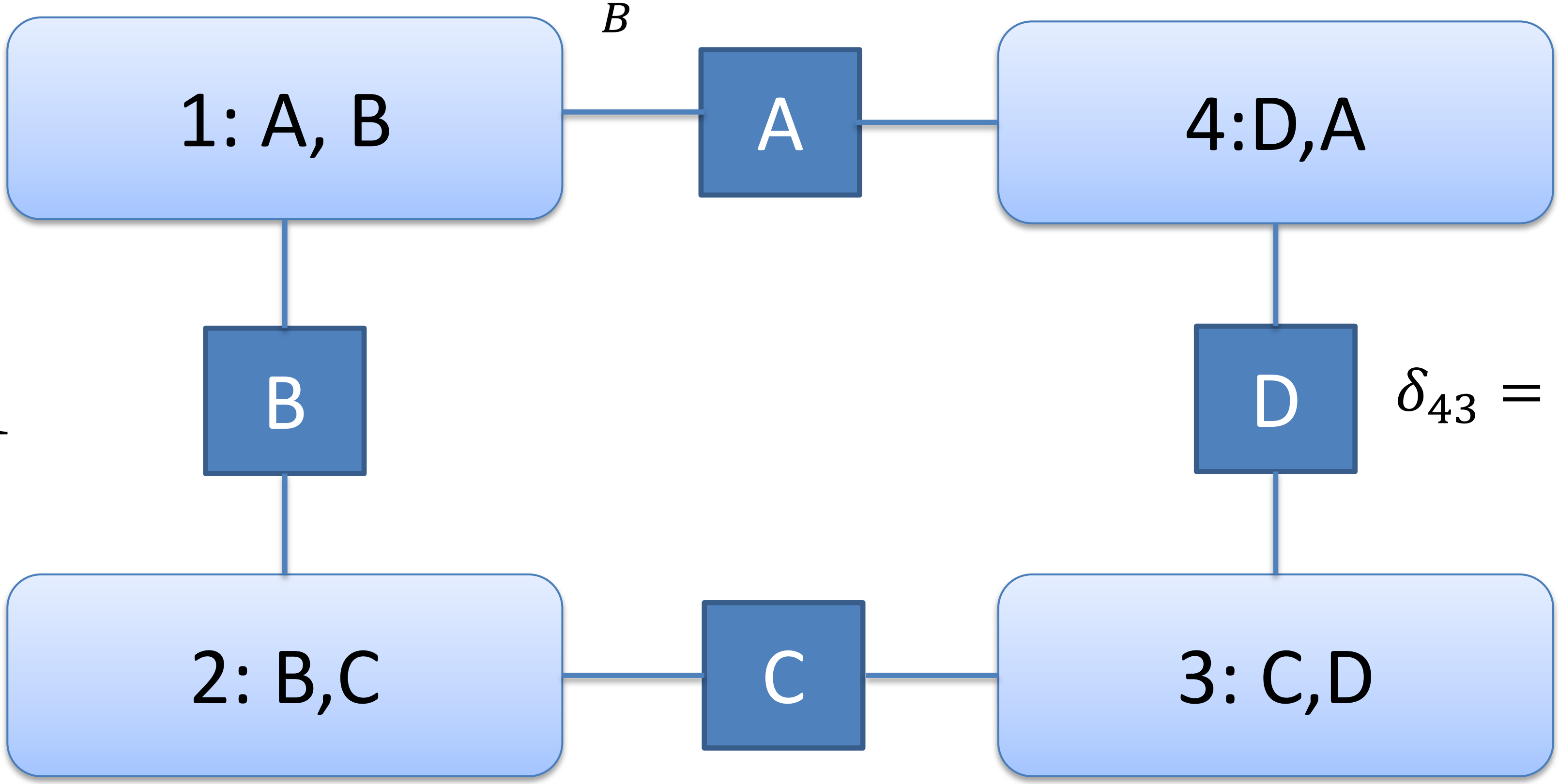
c^0	d^0	1
c^0	d^1	100
c^1	d^0	100
c^1	d^1	1

b^0	c^0	100
b^0	c^1	1
b^1	c^0	1
b	c^1	100

Example: Misconception

$$\delta_{14} = \sum_{B} \delta_{21}(B)\, \psi_1(A, B)$$

$$\delta_{21} = 1$$



$$\delta_{43} = \sum_{A} \delta_{14}(A)\, \psi_3(D, A)$$

$$\delta_{32} = \sum_{D} \delta_{43}(A)\, \psi_3(C, d)$$

Cluster Graphs

- Undirected graph such that:
- nodes are clusters $C_i \subseteq \{X_1, \ldots, X_n\}$
- edge between C_i and C_j associated with sepset

$$S_{i,j} \subseteq C_i \cap C_j$$

- Given set of factors Φ, we assign each ϕ_k to a single cluster $\boldsymbol{C}_{\alpha(\mathrm{k})}$ s.t. Scope $[\phi_{\mathrm{k}}] \subseteq \boldsymbol{C}_{\alpha(\mathrm{k})}$

- Initial belief of a cluster: $\psi_i(C_i) = \prod_{k:a(k)=i} \phi_k$

Properties of Cluster Graphs

Family Preservation:

For each factor $\phi_k \in \Phi$, there exists a cluster C_i s.t. Scope $[\phi_k] \subseteq \boldsymbol{C}_i$

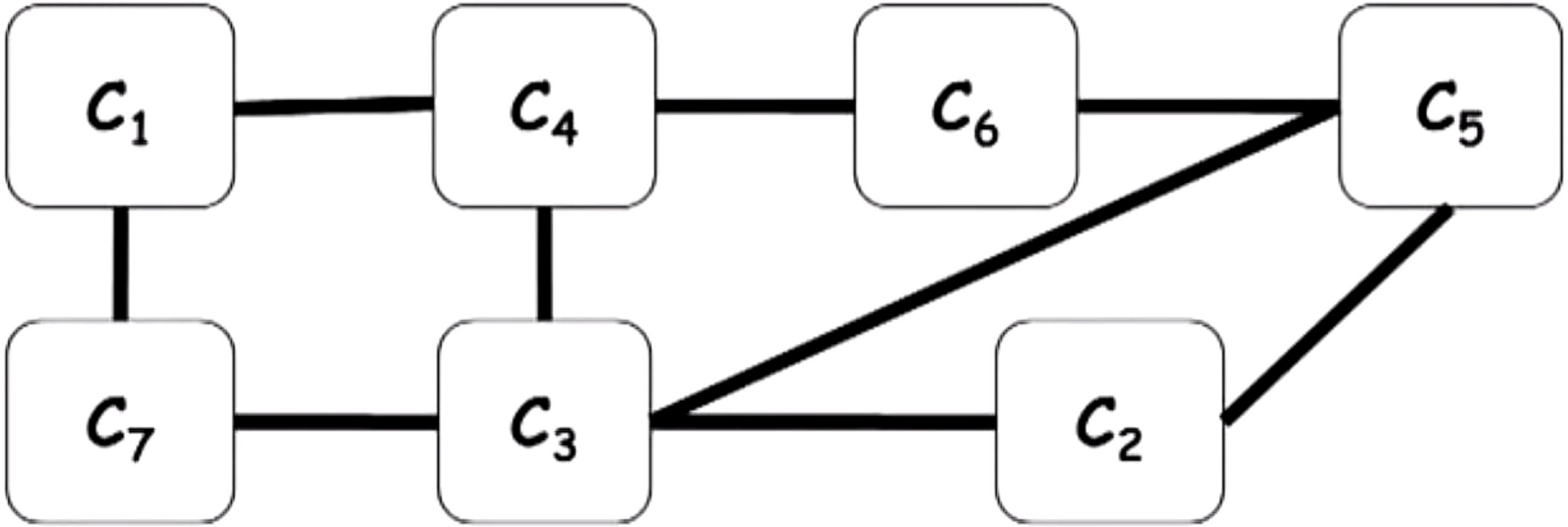
every factor has a node that can accommodate it

Running Intersection:

For each pair of clusters $C_{\mathrm{i}}, C_{\mathrm{j}}$ and variable $X \in C_i \cap C_j$, there exists a unique path between C_i and C_j, all clusters and sepsets contain X.

clusters that include the same variable need to communicate for consistency

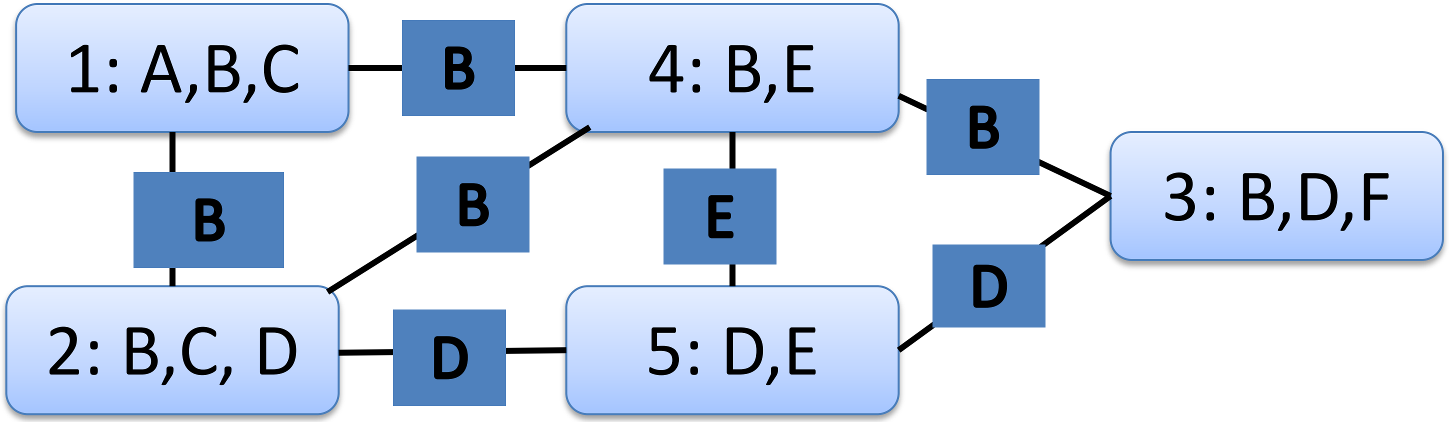
RIP: Existence/Uniqueness



Example: X in C_7, C_5

Example Cluster Graph

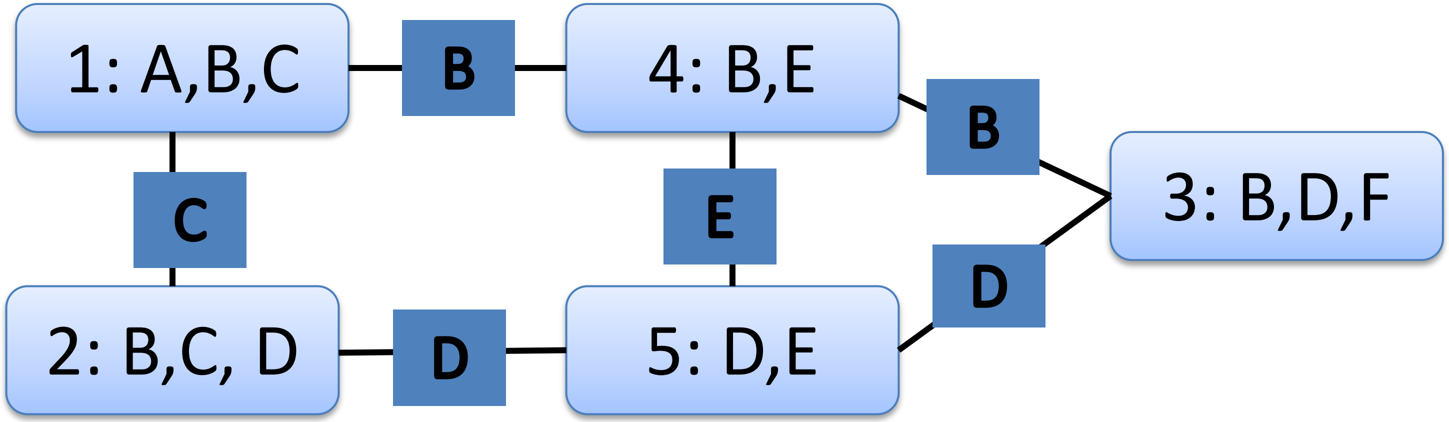
$\phi_1(\mathrm{A, B, C}), \phi_2(\mathrm{B, D}), \phi_3(D, E)\ \phi_4(B, E), \phi_5(B, D, F)$



Assign factors to clusters

Illegal Cluster Graph

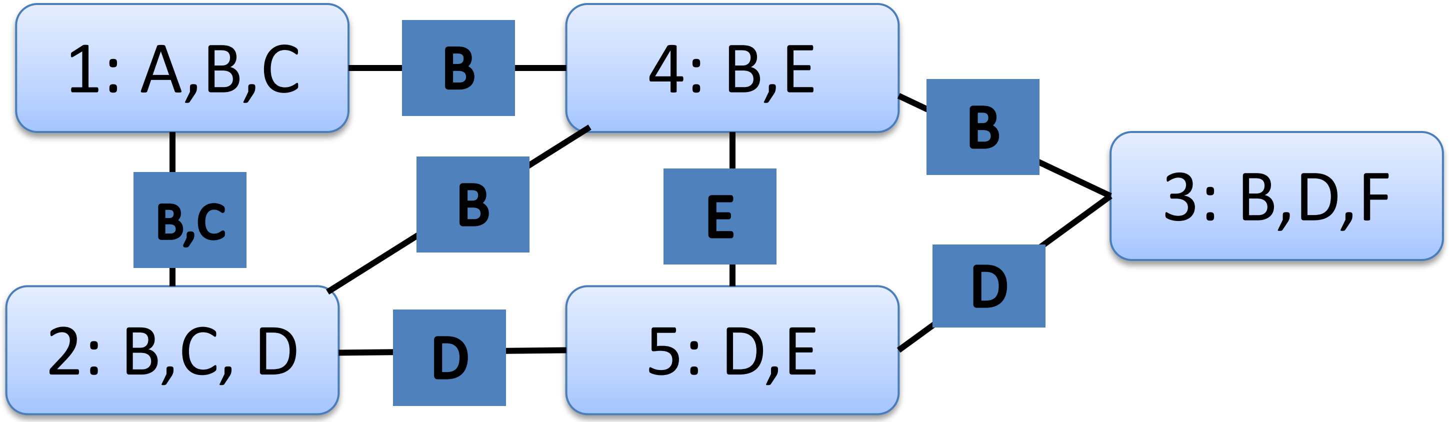
$\phi_1(\mathrm{A,B,C}), \phi_2(\mathrm{B,D}), \phi_3(D,E)\ \phi_4(B,E), \phi_5(B,D,F)$



Violates existence

Illegal Cluster Graph

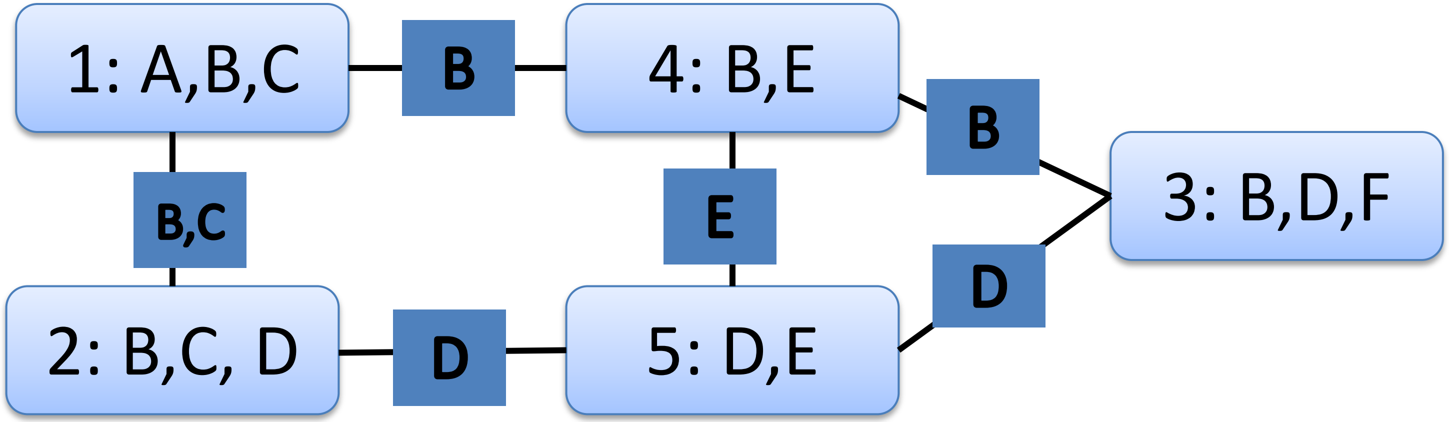
$\phi_1(\mathrm{A, B, C}), \phi_2(\mathrm{B, D}), \phi_3(D, E)\ \phi_4(B, E), \phi_5(B, D, F)$



Violates uniqueness

Legal Cluster Graph

$$\phi_1(\mathrm{A,B,C}), \phi_2(\mathrm{B,D}), \phi_3(D,E)\ \phi_4(B,E), \phi_5(B,D,F)$$



Bethe Cluster Graph

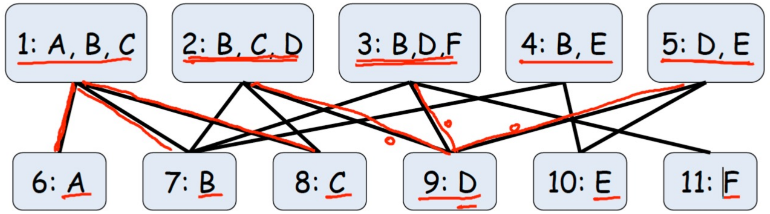
$\phi_1(\mathrm{A, B, C}), \phi_2(\mathrm{B, D}), \phi_3(D, E)\ \phi_4(B, E), \phi_5(B, D, F)$

- For each $\phi_k \in \Phi$, a factor cluster $C_k = \mathrm{Scope}[\phi_k]$
- For each $\mathrm{X_i}$ a singleton cluster $\{\mathrm{X_i}\}$
- Edge $C_k - X_i$ if $\mathrm{X_i} \in C_k$

Bethe Cluster Graph

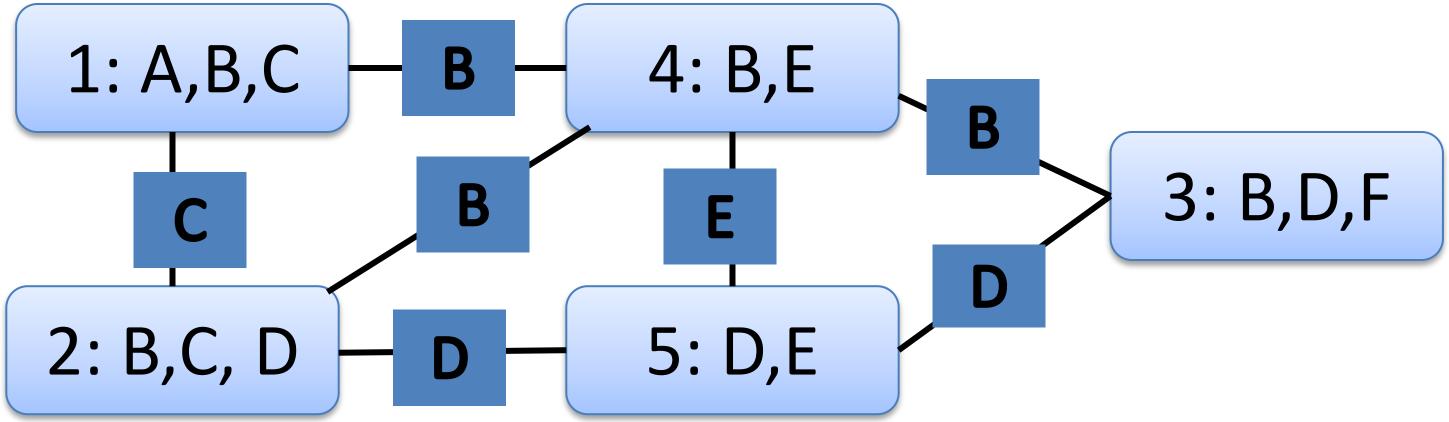
$\phi_1(\mathrm{A, B, C}), \phi_2(\mathrm{B, D}), \phi_3(D, E)\ \phi_4(B, E), \phi_5(B, D, F)$

- For each $\phi_k \in \Phi$, a factor cluster $C_k = \mathrm{Scope}[\phi_k]$
- For each $\mathrm{X_i}$ a singleton cluster $\{\mathrm{X_i}\}$
- Edge $C_k - X_i$ if $\mathrm{X_i} \in C_k$



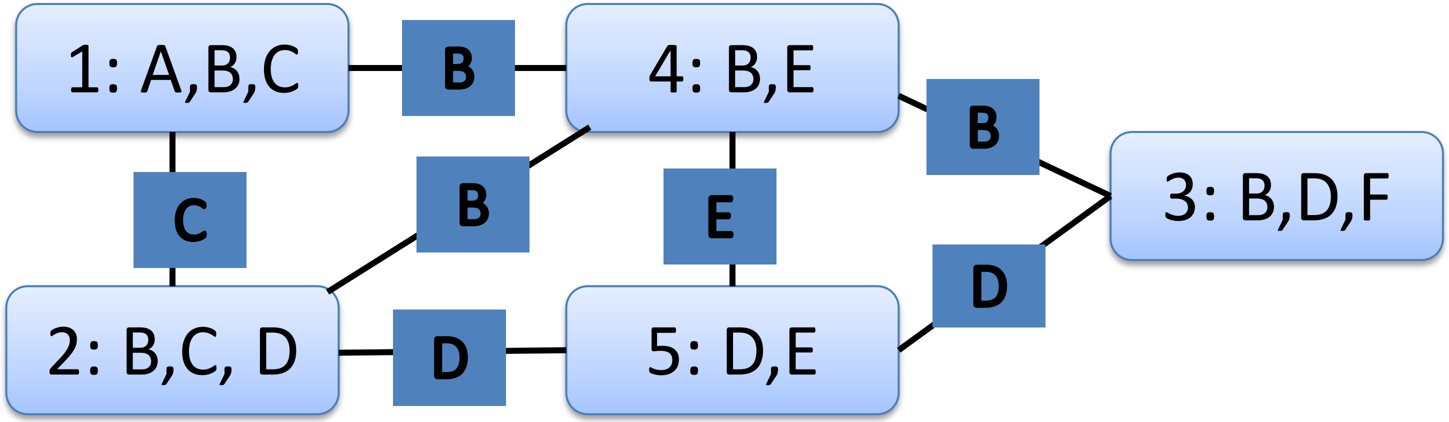
Message Passing

$$\delta_{1\to4}(B) = \sum_{A,C} \psi_1(A,B,C)\delta_{2\to1}(C)$$



Message Passing

$$\delta_{4\to1}(B) = \sum_{\mathrm{E}} \psi_4(B, E)\delta_{2\to4}(\mathrm{B})\delta_{5\to4}(\mathrm{E})\delta_{3\to4(B)}$$



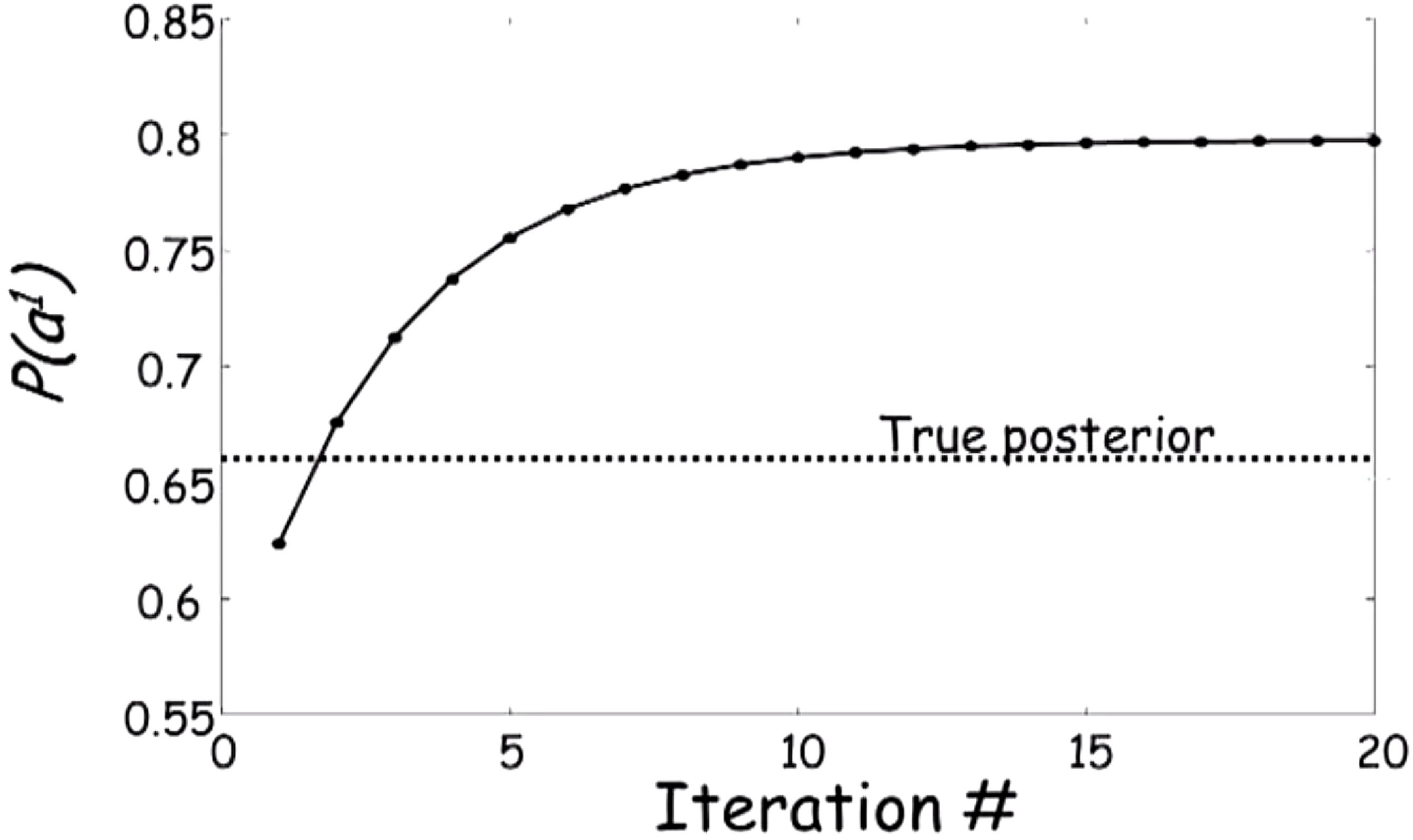
Loopy Belief Propagation

- Assign each factor $\phi_k \in \Phi$ to a cluster $C_{\alpha(k)}$
- Construct initial potentials $\psi_i(C_i) = \prod_{k:\alpha(k)=i} \phi_k$
- Initialize all messages to be 1.
- Select edge (i,j) and pass message

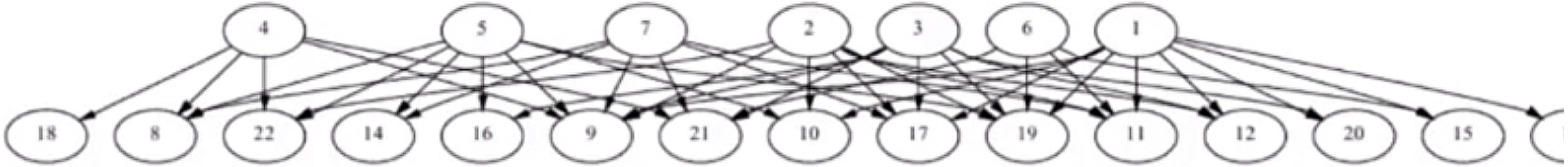
$$\delta_{i\to j}(\boldsymbol{S}_{i,j}) = \sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \psi_i \times \prod_{k\in(\mathcal{N}_i - \{j\})} \delta_{k\to i}$$

- Compute $\beta_i(C_i) = \psi_i \times \prod_{k\in\mathcal{N}_i} \delta_{k\to i}$

Loopy Belief Propagation - Properties

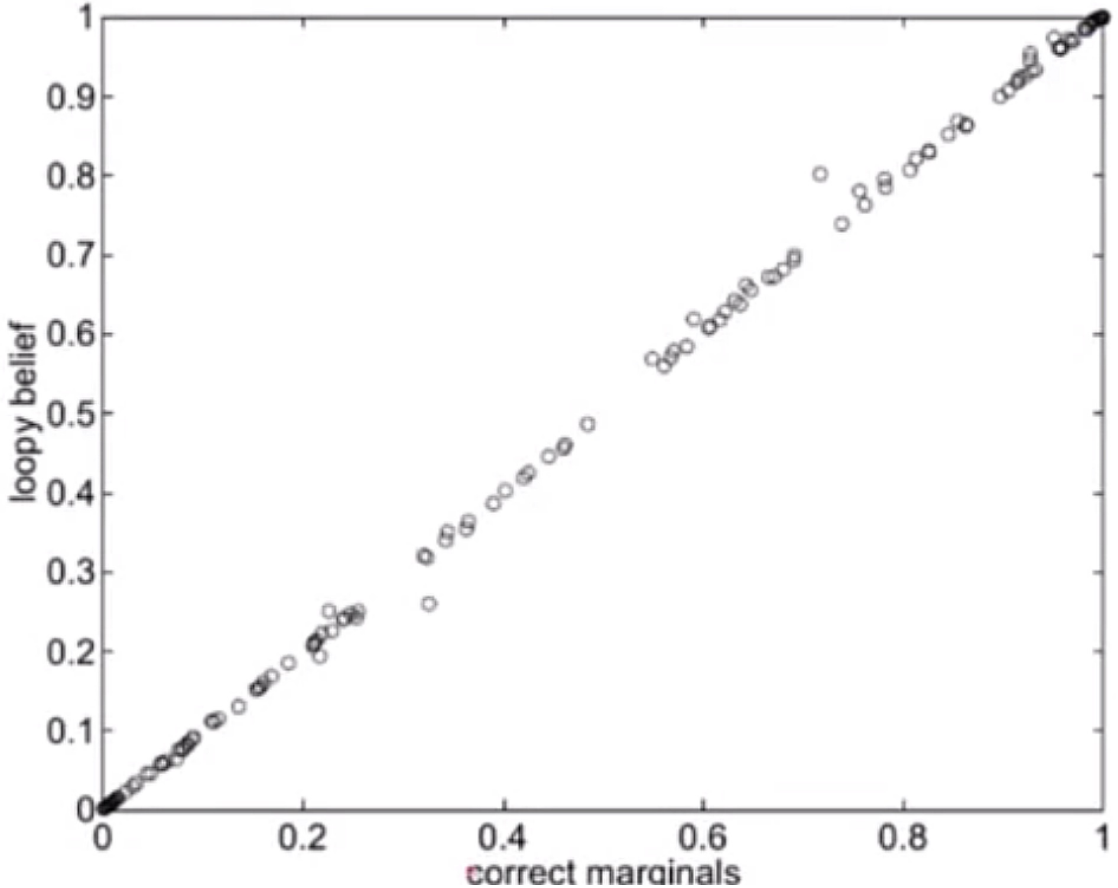


Example

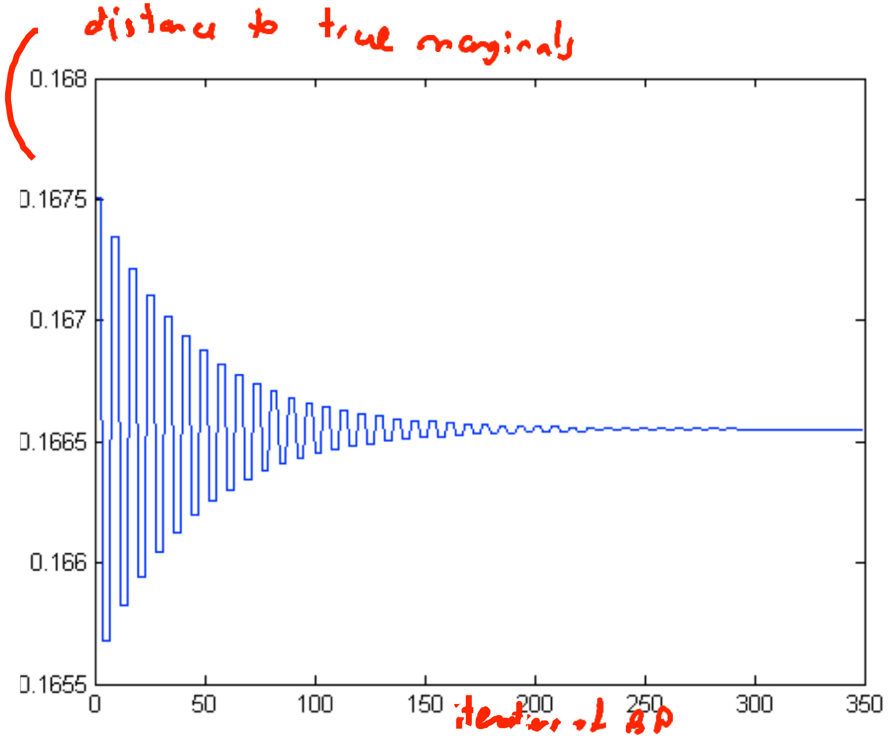


Sample Results

Murphy, Weiss, Jordan
UAI 99



Misconception Revisited

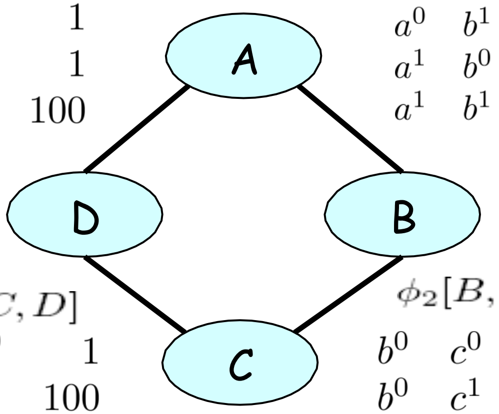


$\phi_4[D, A]$

d^0	a^0	100
d^0	a^1	1
d^1	a^0	1
d^1	a^1	100

$\phi_1[A, B]$

a^0	b^0	100
a^0	b^1	2
a^1	b^0	1
a^1	b^1	100



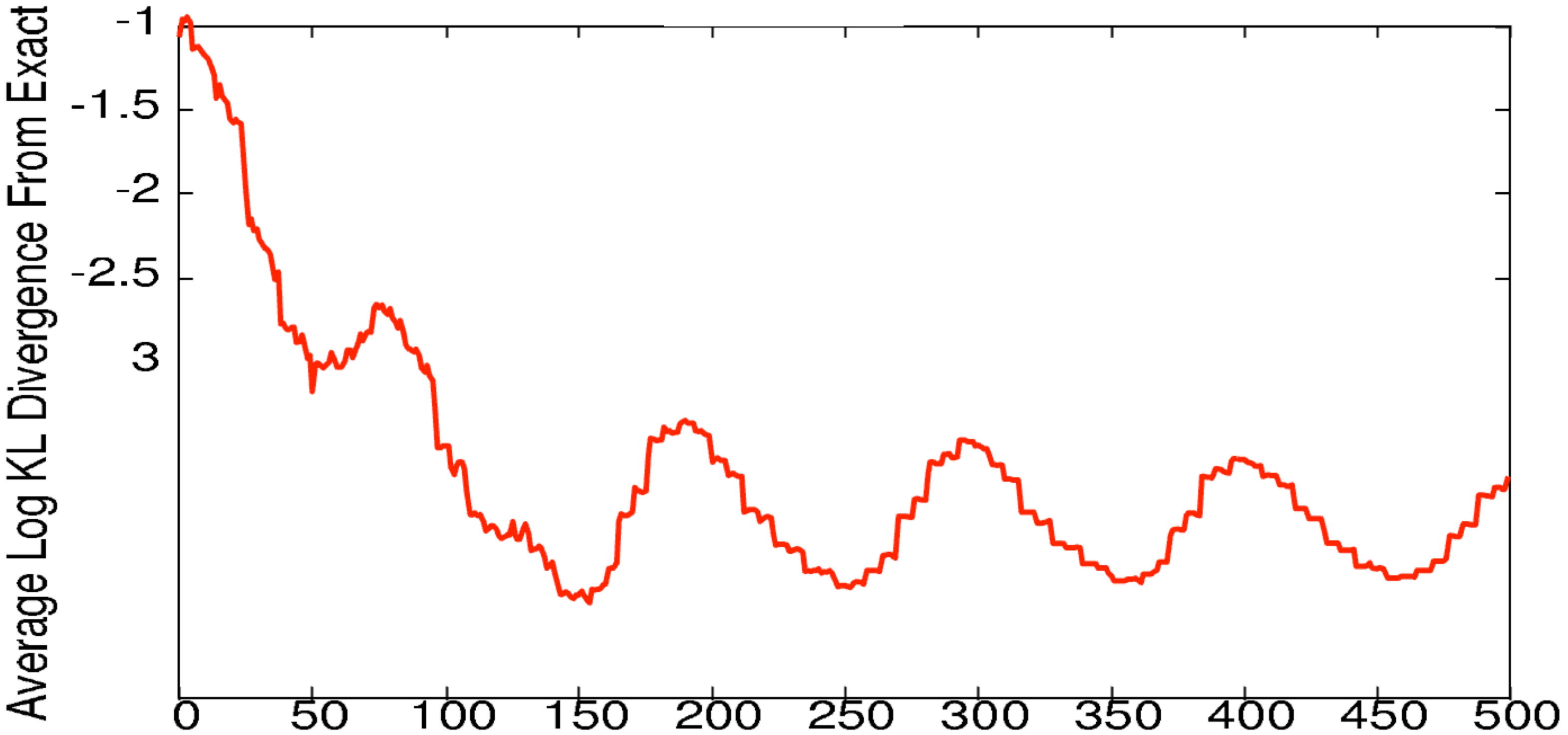
$\phi_3[C, D]$

c^0	d^0	1
c^0	d^1	100
c^1	d^0	100
c^1	d^1	1

$\phi_2[B, C]$

b^0	c^0	100
b^0	c^1	1
b^1	c^0	1
b^1	c^1	100

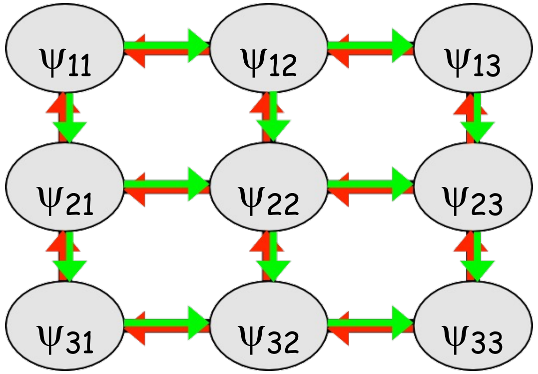
Nonconvergent BP Run

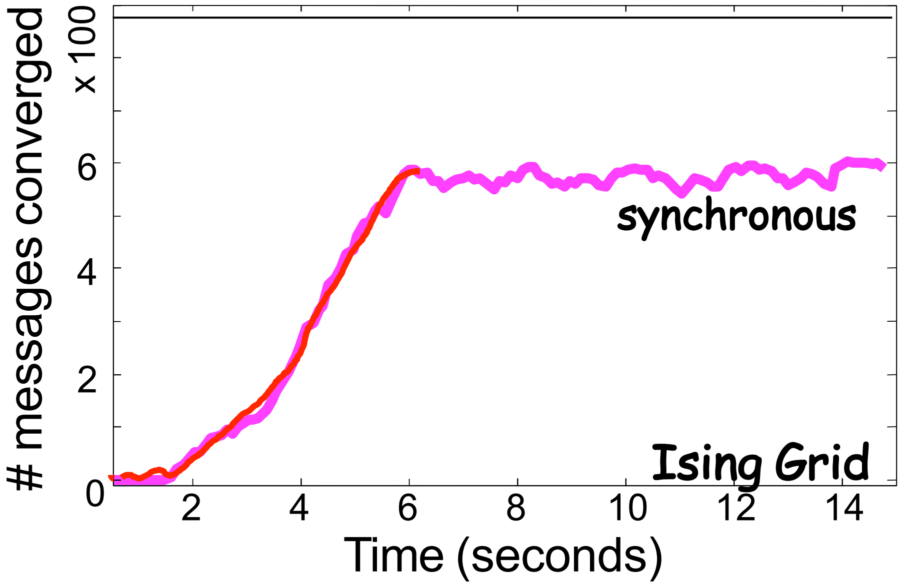


Daphne Koller

Different Variants of BP

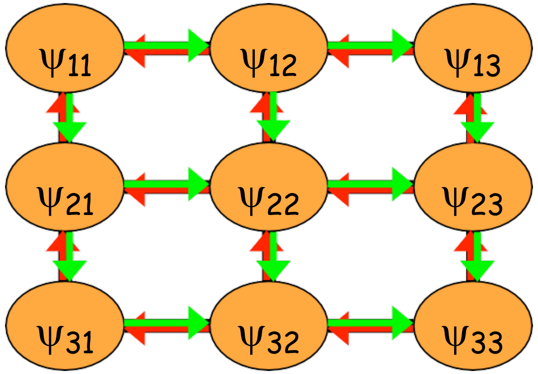
Synchronous BP:
all messages are updated in parallel

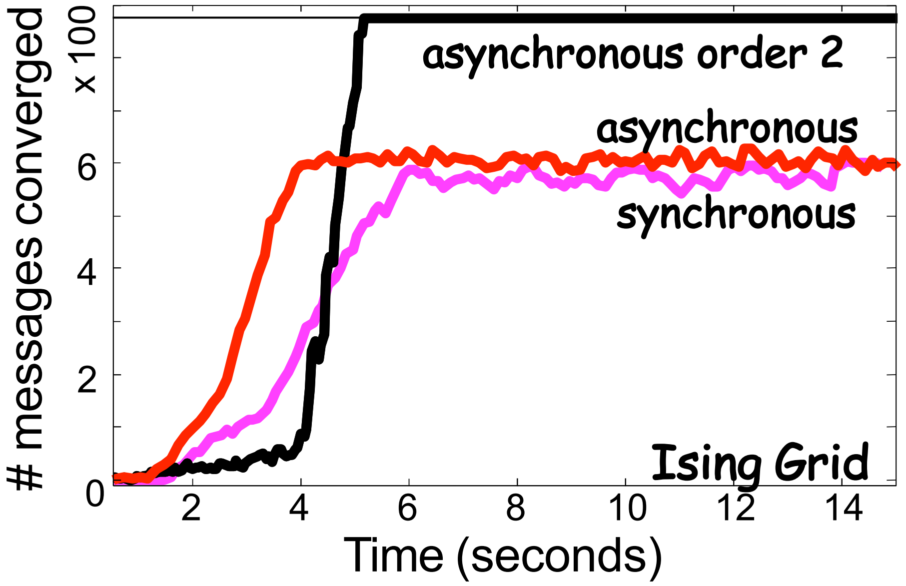




Different Variants of BP

Asynchronous BP:
Messages are updated one at a time



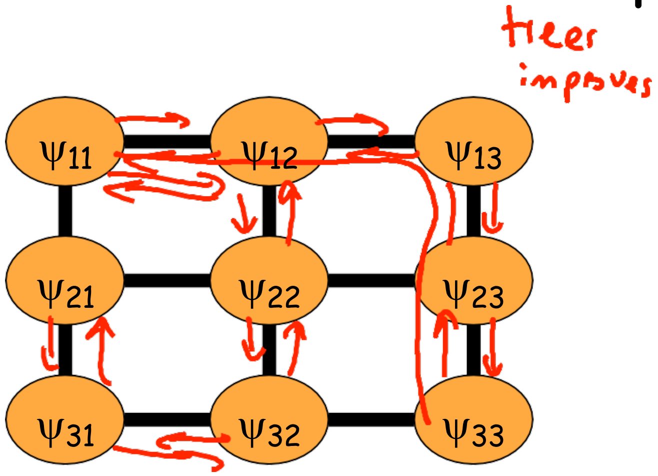


Observations

- Convergence is a local property:
  - some messages converge soon
  - others may never converge
- Synchronous BP converges considerably worse than asynchronous
- Message passing order makes a difference to extent and rate of convergence

Informed Message Scheduling

- Tree reparameterization (TRP)
  - Pick a tree and pass messages to calibrate



Informed Message Scheduling

- Tree reparameterization (TRP)
  - Pick a tree and pass messages to calibrate
- Residual belief propagation (RBP)
  - Pass messages between two clusters whose beliefs over the sepset disagree the most

Smoothing (Damping) Messages

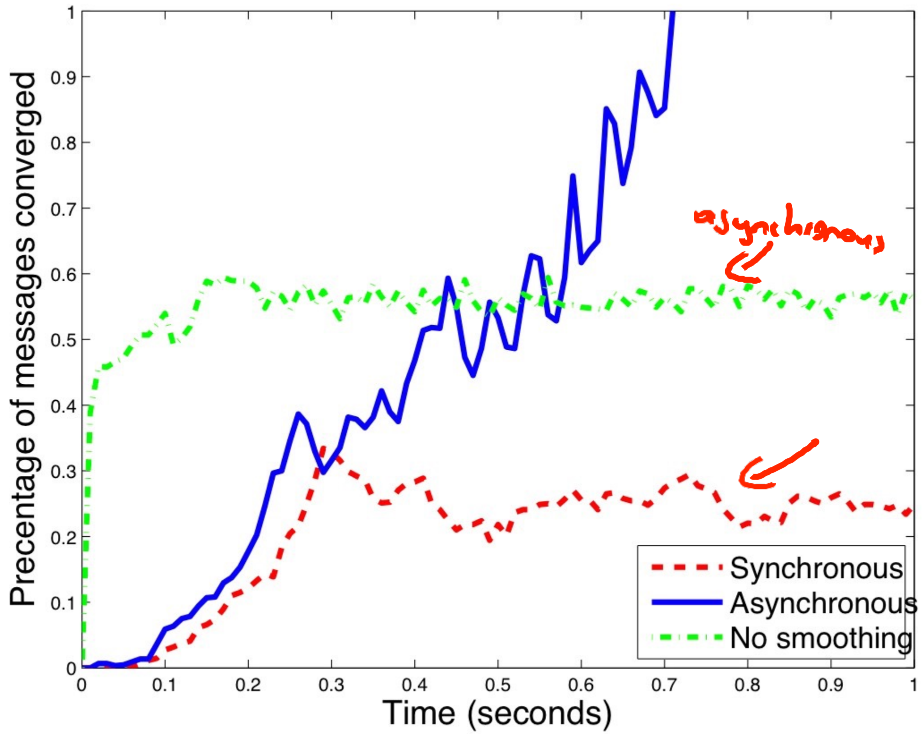
$$\delta_{i\to j} \leftarrow \sum_{C_i - S_{i,j}} \psi_i \prod_{k\neq j} \delta_{k\to i}$$

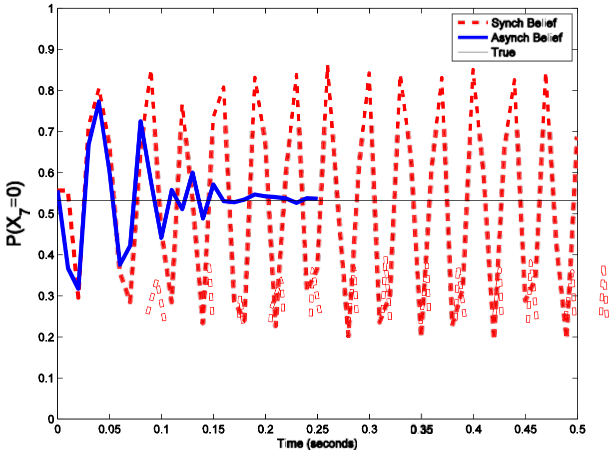
$$\delta_{i\to j} \leftarrow \lambda \left(\sum_{C_i - S_{i,j}} \psi_i \prod_{k\neq j} \delta_{k\to i} \right) + (1-\lambda)\delta^{\text{old}}_{i\to j}$$

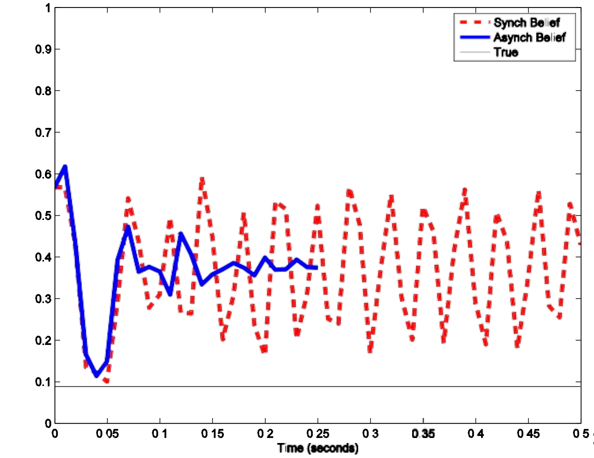
new msg

old msg

- Dampens oscillations in messages


Precentage of messages converged
Time (seconds)
Synchronous
Asynchronous
No smoothing


P(X_7=0)
Time (seconds)
Synch Belief
Asynch Belief
True


Time (seconds)
Synch Belief
Asynch Belief
True

Summary

- To achieve BP convergence, two main tricks
  - Damping
  - Intelligent message ordering
- Convergence doesn't guarantee correctness
- Bad cases for BP – both convergence & accuracy:
  - Strong potentials pulling in different directions
  - Tight loops
- Some new algorithms have better convergence:
  - Optimization-based view to inference