

Probabilistic Graphical Models

Clique trees & Belief Propagation

Siamak Ravanbakhsh

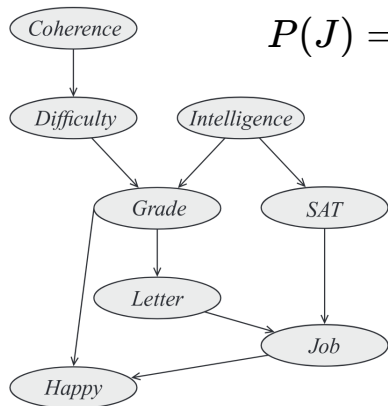
Winter 2021

Learning objectives

- message passing on clique trees
- its relation to variable elimination
- two different forms of belief propagation

Recap: variable elimination (VE)

- marginalize over a subset - e.g., $P(J) = \sum_{C,D,I,G,S,L,H} P(C, D, I, G, S, L, J, H)$
- expensive to calculate (why?)
- use the **factorized form** of P

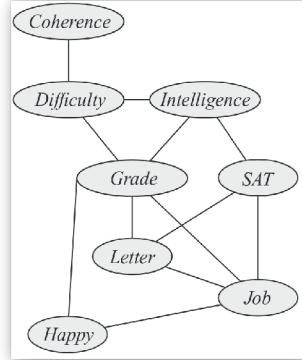
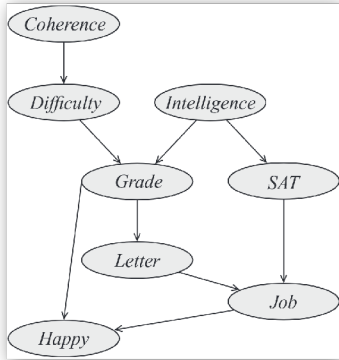


$$P(J) = \sum_{C,D,I,G,S,L,H} P(D|C)P(G|D,I)P(S|I)P(L|G)P(J|L,S)P(H|G,J)$$

Recap: variable elimination (VE)

- use the factorized form of P

$$P(J) = \sum_{C,D,I,G,S,L,H} P(D|C)P(G|D,I)P(S|I)P(L|G)P(J|L,S)P(H|G,J)$$



note that they do not encode the same CIs

$$\phi_2(H, G, J)$$

think of this as a factor/potential
same treatment of

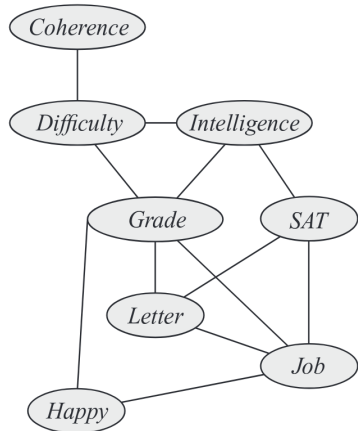
- Bayes-nets
- Markov nets

for inference

Recap: variable elimination (VE)

- use the factorized form of P

$$P(J) \propto \sum_{C,D,I,G,S,L} \phi_1(D, C) \phi_2(G, D, I) \phi_3(S, I) \phi_4(L, G) \phi_5(J, L, S) \phi_6(H, G, J)$$



$$= \dots \sum_I \phi_3(S|I) \sum_D \phi_2(G, D, I) \sum_C \phi_1(D, C)$$

- repeat this

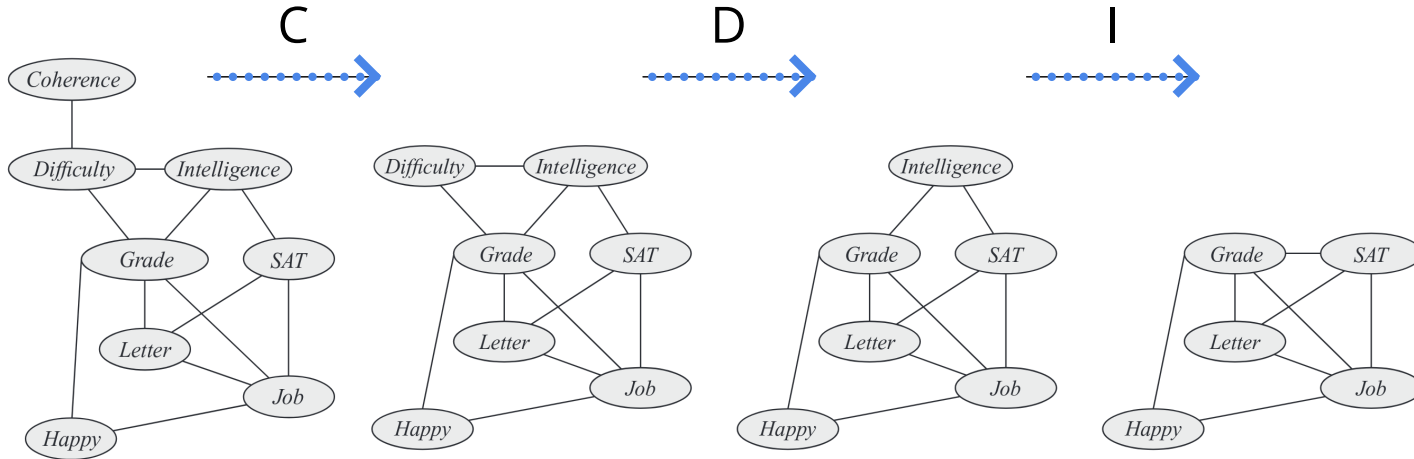
$$\psi'_1(D) \quad \psi_1(D, C)$$

$$= \dots \sum_I \phi_3(S, I) \sum_D \phi_2(G, D, I) \psi'_1(D)$$

$$\psi'_2(G, I) \quad \psi_2(G, I, D)$$

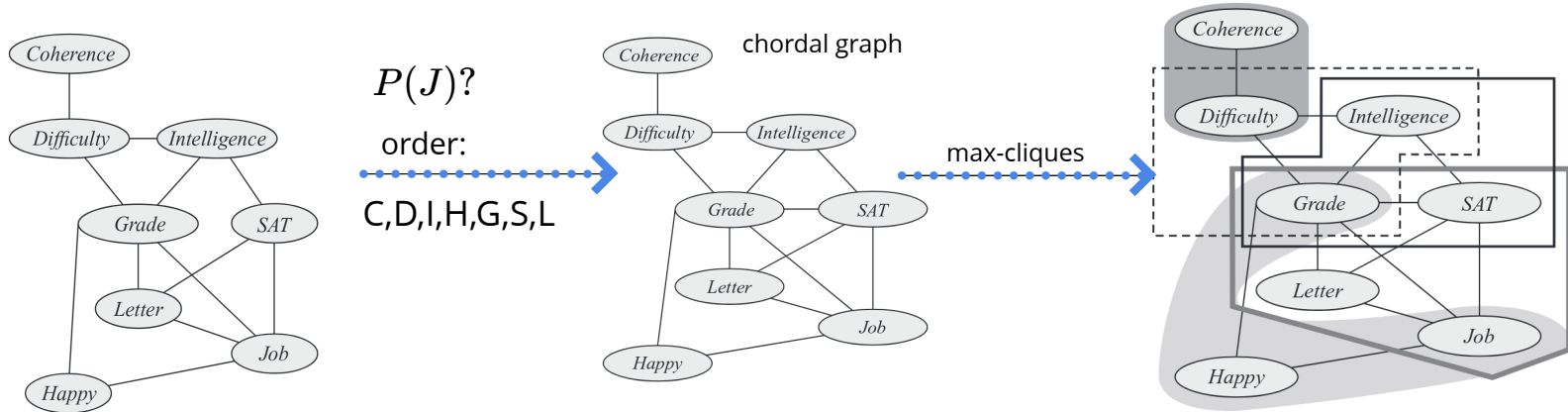
Recap: variable elimination (VE)

- eliminate variables in some order (order of factors in the summation)



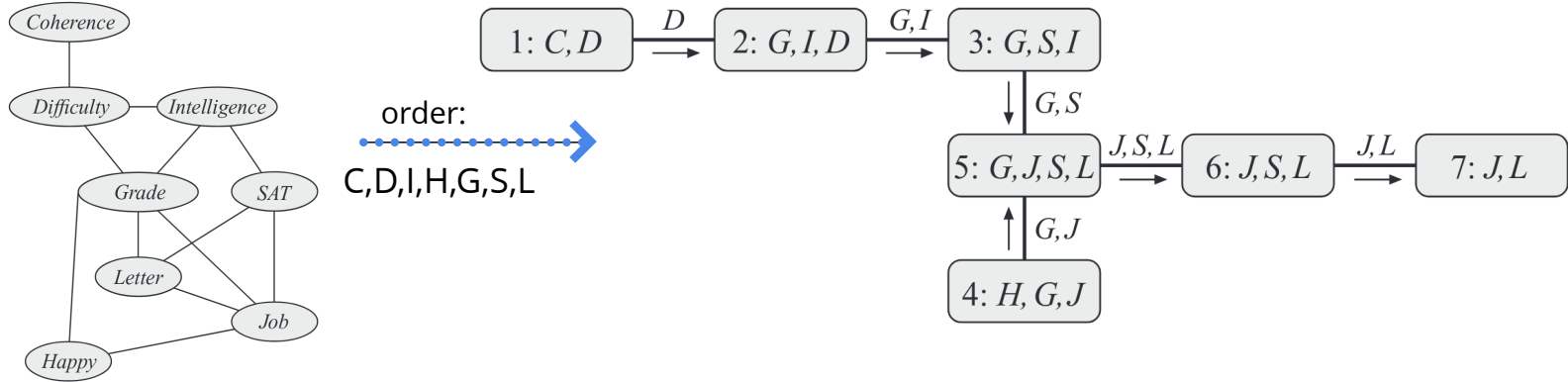
Recap: variable elimination (VE)

- eliminate variables in some order
- creates a **chordal graph**
- **maximal cliques** are factors created during VE (ψ_t)



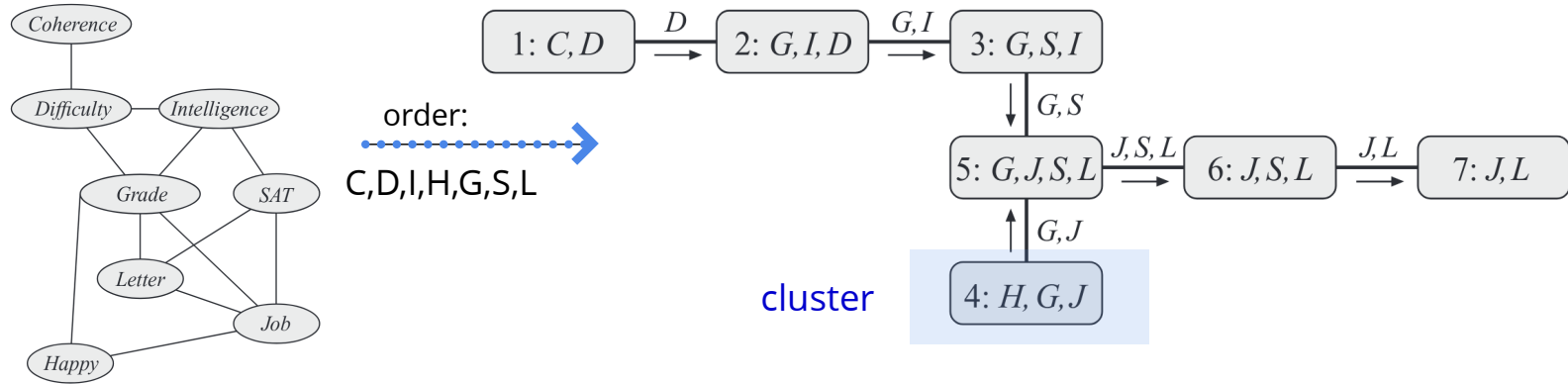
Clique-tree

- summarize the *VE* computation using a **clique-tree**



Clique-tree

- summarize the *VE computation* using a **clique-tree**

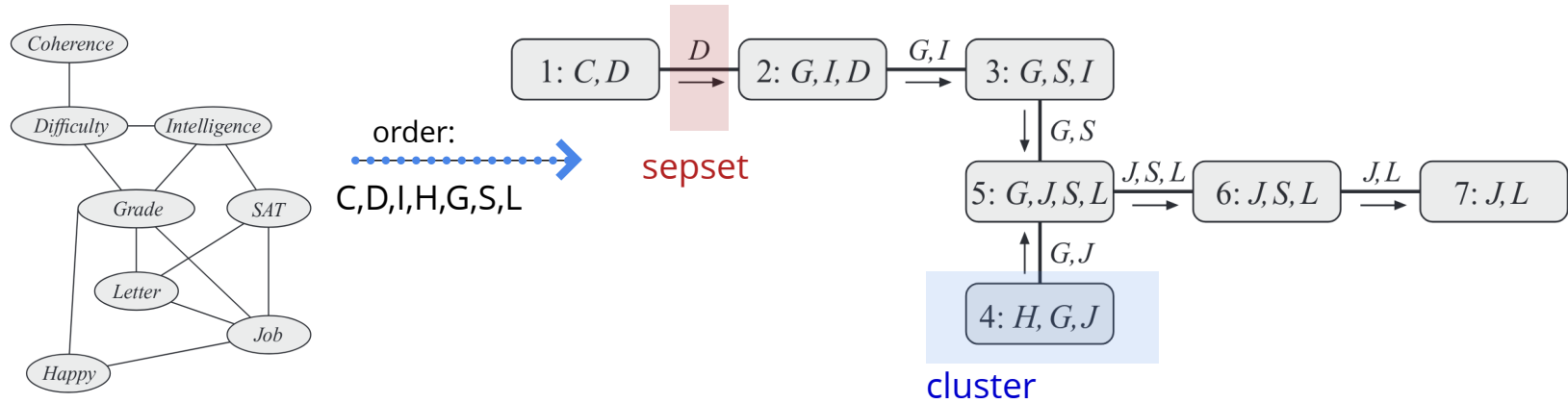


- clusters** are maximal cliques (scope of factors created during VE)

$$C_i = \text{Scope}[\psi_i]$$

Clique-tree

- summarize the *VE computation* using a **clique-tree**



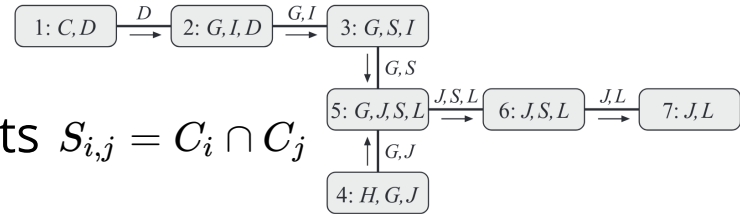
- clusters** are maximal cliques (factors that are marginalized)
- sepsets** are the result of marginalization over cliques

$$C_i = \text{Scope}[\psi_i]$$

$$S_{i,j} = \text{Scope}[\psi'_i]$$

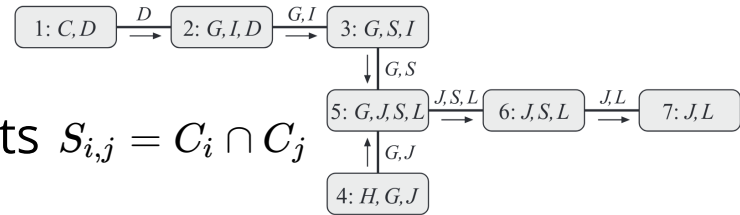
$$S_{i,j} = C_i \cap C_j$$

Clique-tree: **properties**



- a tree \mathcal{T} from clusters C_i and sepsets $S_{i,j} = C_i \cap C_j$
- **family-preserving property:** $\alpha(\phi) = j$
 - each factor ϕ is associated with a cluster C_j s.t. $Scope[\phi] \subseteq C_j$

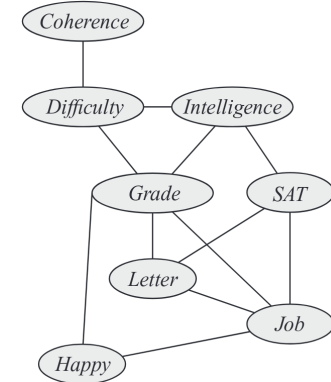
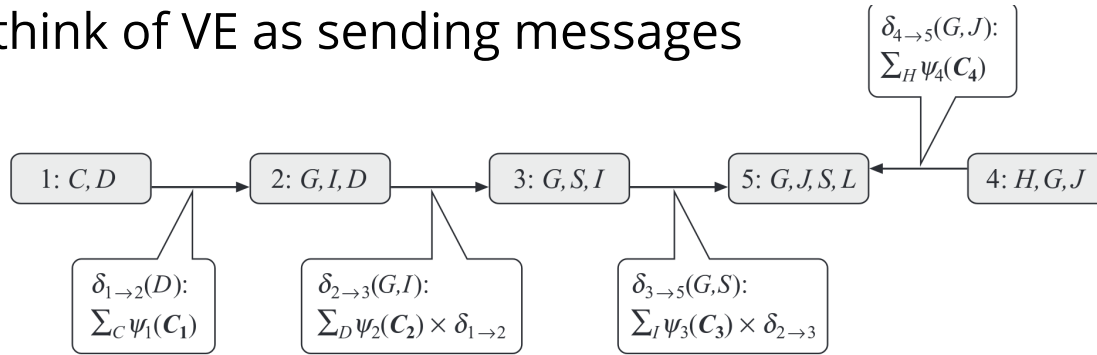
Clique-tree: **properties**



- a tree \mathcal{T} from clusters C_i and sepsets $S_{i,j} = C_i \cap C_j$
- **family-preserving property:** $\alpha(\phi) = j$
 - each factor ϕ is associated with a cluster C_j s.t. $Scope[\phi] \subseteq C_j$
- **running intersection property:**
 - if $X \in C_i, C_j$ then $X \in C_k$ for C_k in the path $C_i \rightarrow \dots \rightarrow C_j$

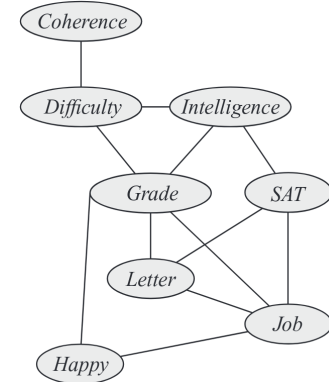
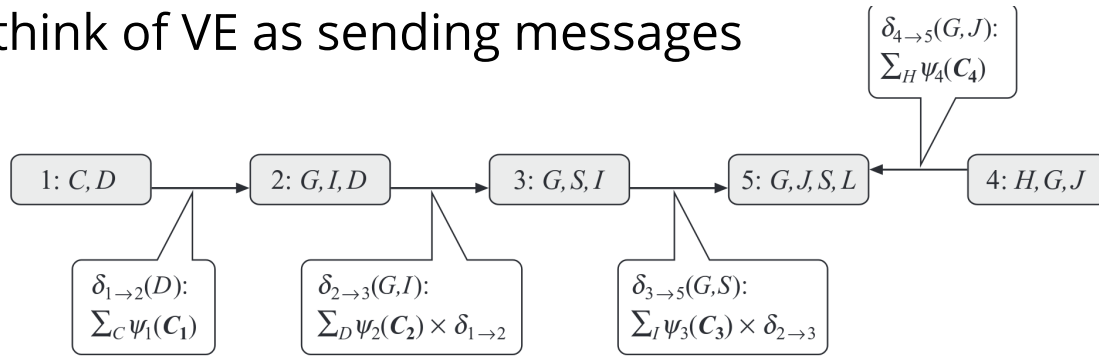
VE as message passing

think of VE as sending messages



VE as message passing

think of VE as sending messages



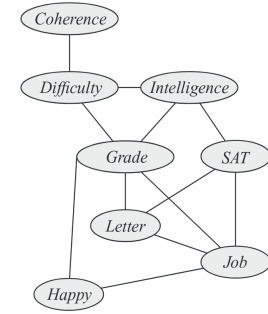
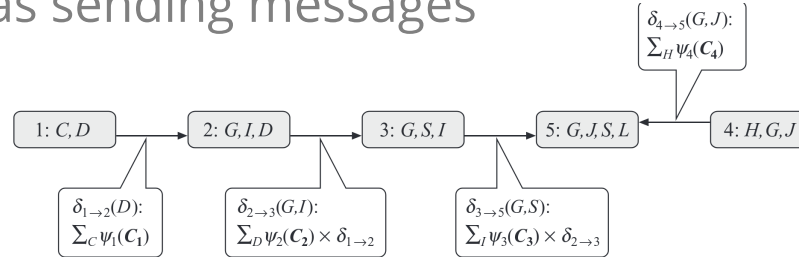
calculate the product of factors in each clique $\psi_i(C_i) \triangleq \prod_{\phi: \alpha(\phi)=i} \phi$

send messages from the leaves towards a root:

$$\delta_{i \rightarrow j}(S_{i,j}) = \sum_{C_{i-S_{i,j}}} \psi_i(C_i) \prod_{\substack{k \in Nb_{i-j} \\ \text{neighbours}}} \delta_{k \rightarrow i}(S_{i,k})$$

message passing

- think of VE as sending messages



- send messages from the leaves towards a root:

$$\begin{aligned} \delta_{i \rightarrow j}(S_{i,j}) &= \sum_{C_{i-S_{i,j}}} \psi_i(C_i) \prod_{k \in Nb_{i-j}} \delta_{k \rightarrow i}(S_{i,k}) \\ &= \sum_{\mathcal{V} \prec (i \rightarrow j)} \prod_{\phi \in \mathcal{F} \prec (i \rightarrow j)} \phi \end{aligned}$$

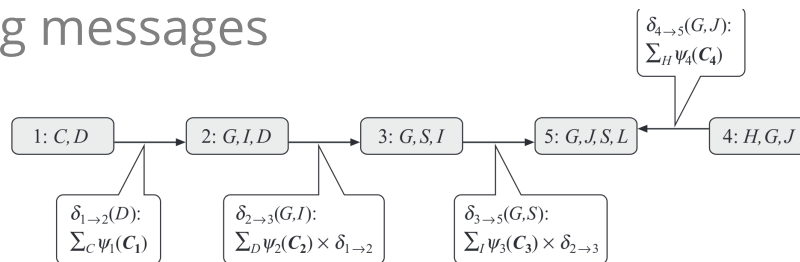
all variable on i side of the tree

all the factors on i side of the tree

- the message is the marginal from one side of the tree

message passing

- think of VE as sending messages



- send messages from the leaves towards a root:

$$\delta_{i \rightarrow j}(S_{i,j}) \triangleq \sum_{C_i - S_{i,j}} \psi_i(C_i) \prod_{k \in Nb_{i-j}} \delta_{k \rightarrow i}(S_{i,k})$$

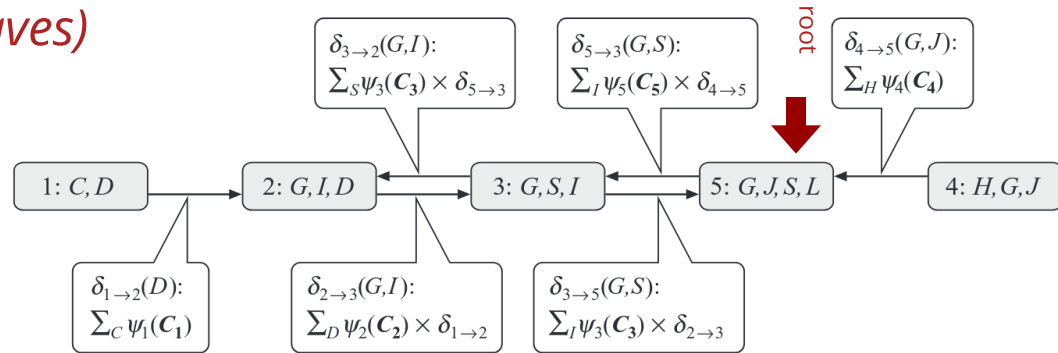
- the **belief** at the root clique is $\beta_r(C_r) \triangleq \psi_r(C_r) \prod_{k \in Nb_r} \delta_{k \rightarrow r}(S_{r,k})$

proportional to the marginal $\beta_r(C_r) \propto \sum_{\mathbf{x} - C_i} P(\mathbf{X})$

message passing: **root-to-leaves**

- what if we continue sending messages?

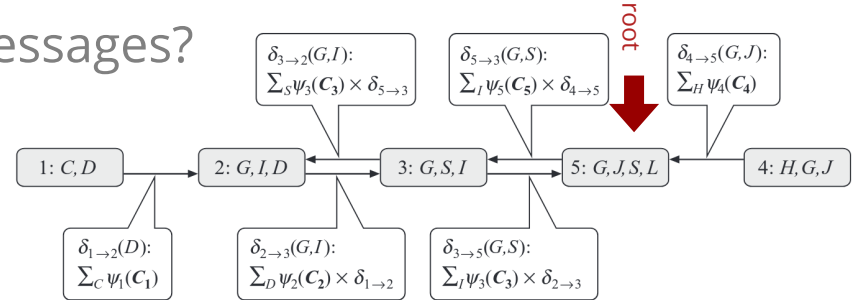
(from the root to leaves)



- clique **i** sends a message to clique **j** when received messages from all the **other** neighbors **k**

message passing: **root-to-leaves**

- what if we continue sending messages?
(from the root to leaves)



- sum-product **belief propagation (BP)**

$$\delta_{i \rightarrow j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \psi_i(C_i) \prod_{k \in Nb_{i-j}} \delta_{k \rightarrow i}(S_{i,k})$$

$$\mu_{i,j}(S_{i,j}) \triangleq \delta_{i \rightarrow j}(S_{i,j}) \delta_{j \rightarrow i}(S_{i,j})$$

$$\beta_i(C_i) \triangleq \psi_i(C_i) \prod_{k \in Nb_i} \delta_{k \rightarrow i}(S_{i,k})$$

marginals

for any clique (not only root)

Summery so far...

- VE creates a chordal induced graph
- maximum cliques in this graph: **clusters**
- **message passing** view of VE:
 - send messages between clusters towards a root
- going **beyond VE**:
 - send messages back from the root
 - produce marginal over all clusters

Clique-tree: calibration

represent P using marginals: $\frac{\prod_i \beta_i}{\prod_{i,j \in \mathcal{E}} \mu_{i,j}} = \frac{\prod_i \psi_i \prod_{k \rightarrow i} \delta_{k \rightarrow i}}{\prod_{i,j \in \mathcal{E}} \delta_{i \rightarrow j} \delta_{j \rightarrow i}} = \prod_i \psi_i = \tilde{P}$

Clique-tree: calibration

represent P using marginals: $\frac{\prod_i \beta_i}{\prod_{i,j \in \mathcal{E}} \mu_{i,j}} = \frac{\prod_i \psi_i \prod_{k \rightarrow i} \delta_{k \rightarrow i}}{\prod_{i,j \in \mathcal{E}} \delta_{i \rightarrow j} \delta_{j \rightarrow i}} = \prod_i \psi_i = \tilde{P}$

an arbitrary assignment to all $\beta_i, \mu_{i,j}$ is calibrated iff

BP produces calibrated beliefs $\mu_{i,j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \beta_i(C_i) = \sum_{C_j - S_{i,j}} \beta_j(C_j)$

Clique-tree: calibration

represent P using marginals: $\frac{\prod_i \beta_i}{\prod_{i,j \in \mathcal{E}} \mu_{i,j}} = \frac{\prod_i \psi_i \prod_{k \rightarrow i} \delta_{k \rightarrow i}}{\prod_{i,j \in \mathcal{E}} \delta_{i \rightarrow j} \delta_{j \rightarrow i}} = \prod_i \psi_i = \tilde{P}$ ★

an arbitrary assignment to all $\beta_i, \mu_{i,j}$ is **calibrated** iff

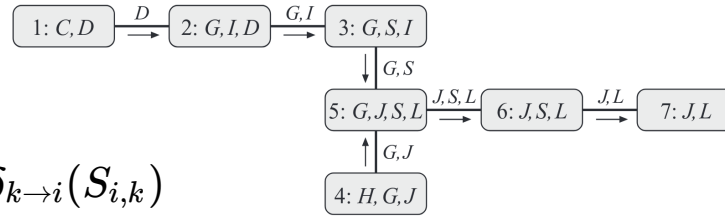
BP produces calibrated beliefs $\mu_{i,j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \beta_i(C_i) = \sum_{C_j - S_{i,j}} \beta_j(C_j)$

being calibrated and ★ means that all $\beta_i, \mu_{i,j}$ are marginals

$$\tilde{P}(\mathbf{X}) \propto \frac{\prod \beta_i(C_i)}{\prod_{i,j \in \mathcal{E}} \mu_{i,j}(S_{i,j})} \iff \beta_i(C_i) \propto P(C_i)$$

BP: an alternative update

approach 1. **message update**



$$\delta_{i \rightarrow j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \psi_i(C_i) \prod_{k \in Nb_i - j} \delta_{k \rightarrow i}(S_{i,k})$$

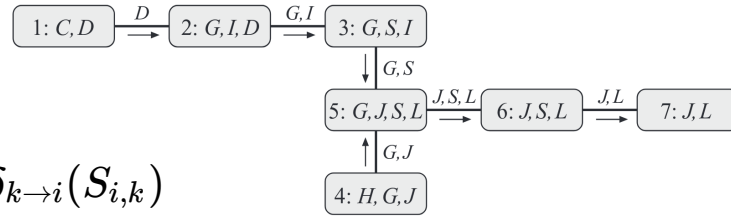
calculate the beliefs **in the end** $\beta_i(C_i) = \psi_i(C_i) \prod_{k \in Nb_i} \delta_{k \rightarrow i}(S_{i,k})$

Update the beliefs so that:

- they are calibrated
- they satisfy

BP: an alternative update

approach 1. **message update**



$$\delta_{i \rightarrow j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \psi_i(C_i) \prod_{k \in Nb_i - j} \delta_{k \rightarrow i}(S_{i,k})$$

calculate the beliefs **in the end** $\beta_i(C_i) = \psi_i(C_i) \prod_{k \in Nb_i} \delta_{k \rightarrow i}(S_{i,k})$

approach 2. **belief update idea**

Update the beliefs so that:

- they are calibrated $\mu_{i,j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \beta_i(C_i) = \sum_{C_j - S_{i,j}} \beta_j(C_j)$
- they satisfy $\frac{\prod_i \beta_i}{\prod_{i,j \in \mathcal{E}} \mu_{i,j}} = \prod_i \psi_i$

BP: an alternative update

belief update

initialize $\beta_i \leftarrow \psi_i = \prod_{\phi:\alpha(\phi)=i} \phi, \quad \mu_{i,j} \leftarrow 1$

until convergence:

pick some $(i, j) \in \mathcal{E}$

$$\hat{\mu}_{i,j} \leftarrow \sum_{C_i - S_{i,j}} \beta_i \quad // \quad \hat{\mu}_{i,j} = \delta_{i \rightarrow j}^{new} \delta_{j \rightarrow i}$$

$$\beta_j \leftarrow \beta_j \frac{\hat{\mu}_{i,j}}{\mu_{i,j}} \quad // \quad \frac{\hat{\mu}_{i,j}}{\mu_{i,j}} = \frac{\delta_{i \rightarrow j}^{new} \delta_{j \rightarrow i}}{\delta_{i \rightarrow j}^{old} \delta_{j \rightarrow i}} = \frac{\delta_{i \rightarrow j}^{new}}{\delta_{i \rightarrow j}^{old}}$$

$$\mu_{i,j} \leftarrow \hat{\mu}_{i,j}$$

at convergence, beliefs are calibrated $\sum_{C_i - S_{i,j}} \beta_i(C_i) = \sum_{C_j - S_{i,j}} \beta_j(C_j)$
and so they are \propto marginals

Clique-tree & queries

What **type of queries** can we answer?

- **marginals** over subset of cliques $P(A) \quad A \subseteq C_i$

Clique-tree & queries

What **type of queries** can we answer?

- **marginals** over subset of cliques $P(A) \quad A \subseteq C_i$
- updating the beliefs after **new evidence** $P(A \mid E^{(t)} = e^{(t)}) \quad A \subseteq C_i, E \subseteq C_j$
 - multiply the (*previously calibrated*) beliefs $\beta(C_i) \mathbb{I}(E^{(t)} = e^{(t)})$
 - propagate to recalibrate (belief update procedure)

Clique-tree & queries

What **type of queries** can we answer?

- **marginals** over subset of cliques $P(A) \quad A \subseteq C_i$
- updating the beliefs after **new evidence** $P(A \mid E^{(t)} = e^{(t)}) \quad A \subseteq C_i, E \subseteq C_j$
 - multiply the (*previously calibrated*) beliefs $\beta(C_i) \mathbb{I}(E^{(t)} = e^{(t)})$
 - propagate to recalibrate (belief update procedure)
- **marginals outside cliques:** $P(A, B) \quad A \subseteq C_i, B \subseteq C_j$
 - define a super-clique that has both A,B
 - a more efficient alternative?

Clique-tree & queries

What **type of queries** can we answer?

- **marginals** over subset of cliques $P(A) \quad A \subseteq C_i$
- updating the beliefs after **new evidence** $P(A \mid E^{(t)} = e^{(t)}) \quad A \subseteq C_i, E \subseteq C_j$
 - multiply the (*previously calibrated*) beliefs $\beta(C_i) \mathbb{I}(E^{(t)} = e^{(t)})$
 - propagate to recalibrate (belief update procedure)
- **marginals outside cliques:** $P(A, B) \quad A \subseteq C_i, B \subseteq C_j$
 - define a super-clique that has both A,B
 - a more efficient alternative?
- **partition function** $Z = \sum_{C_i} \beta_i(C_i)$

Building a clique-tree

how to create it for a given graphical model:

Building a clique-tree

how to create it for a given graphical model:

1. **triangulation**: make a chordal graph

- e.g. *induced graph* in VE
- finding the chordal graph with min max-clique is NP-hard (heuristics we discussed)

Building a clique-tree

how to create it for a given graphical model:

1. **triangulation**: make a chordal graph

- e.g. *induced graph* in VE
- finding the chordal graph with min max-clique is NP-hard (heuristics we discussed)

2. find maximal cliques (**clusters** in the clique-tree)

- in general graphs NP-hard, but easy for chordal graphs
- assign each factor to a clique

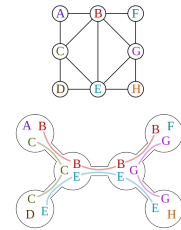


image: wikipedia

Building a clique-tree

how to create it for a given graphical model:

1. triangulation: make a chordal graph

- e.g. *induced graph* in VE
- finding the chordal graph with min max-clique is NP-hard (heuristics we discussed)

2. find maximal cliques (**clusters** in the clique-tree)

- in general graphs NP-hard, but easy for chordal graphs
- assign each factor to a clique

3. use max. spanning-tree to build a tree (edge-cost $|C_i \cap C_j|$)

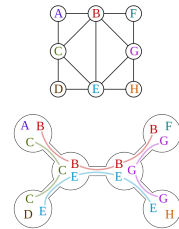
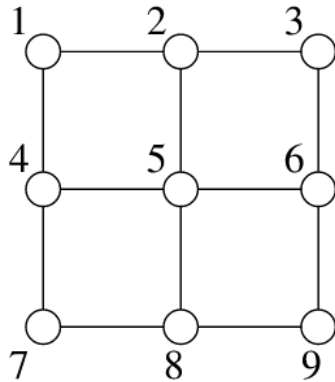


image: wikipedia

Building a clique-tree: **example**

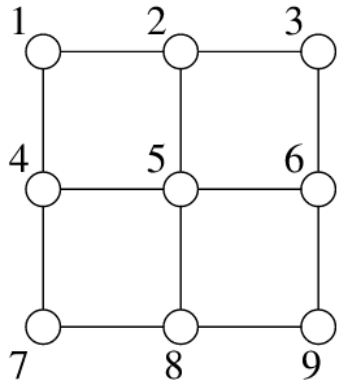
input



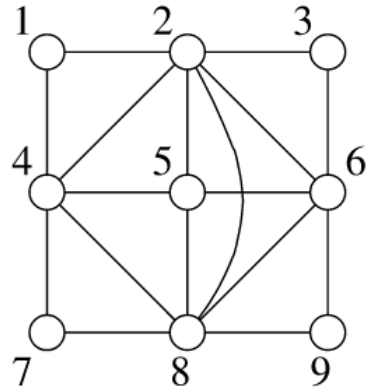
from: wainwright & jordan

Building a clique-tree: **example**

input

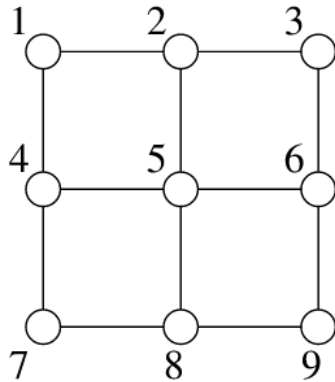


triangulated

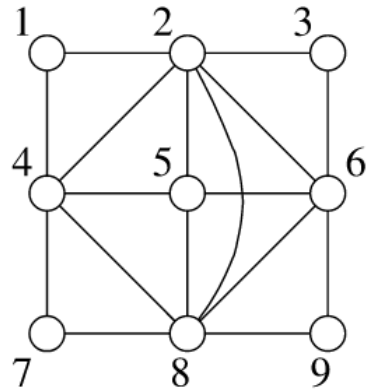


Building a clique-tree: **example**

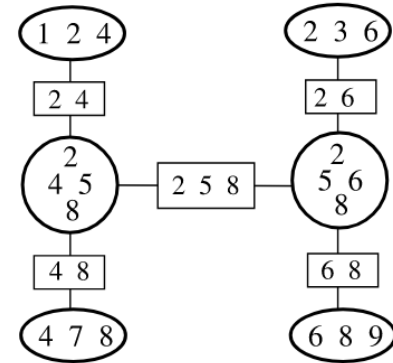
input



triangulated



clique-tree

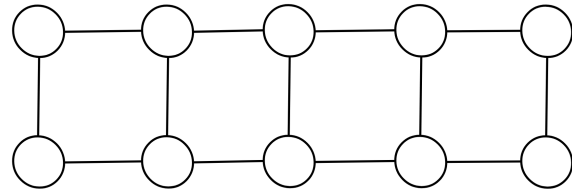


clique-tree quiz

what clique-tree to use here?

what are the sepsets?

cost of exact inference?



Summary

- VE as message passing in a clique-tree
- clique-tree: running intersection & family preserving
- belief propagation updates:
 - message update
 - belief update
- types of queries
- how to build a clique-tree for exact inference

Chordal graph and clique-tree

Chordal graph = Markov \cap Bayesian networks

convert MRF to Bayes-net (the actual procedure):

- triangulate
- build a clique-tree
- **within cliques:** fully connected directed edges
- **between cliques:** from root to leaves