· Variable Elimination and Clique tree

Consider a factor ψ_i to be a computational data structure, which takes "message" τ_j generated by other factors ψ_j, and generate message τ_i that is used by yet another factor

· Cluster Graphs

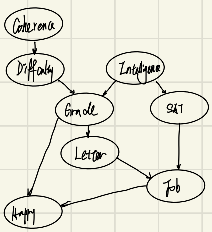
A cluster graph $\mathcal{U}$ for a set of factors Φ over $\mathcal{X}$ is an undirected graph.

each node i is associated with a subset $C_i \subseteq \mathcal{X}$

A cluster graph must be family-preserving: each factor $\phi \in \Phi$ must be associated with a cluster C_i, denoted $\alpha(\phi)$, such that $Scope[\phi] \subseteq C_i$

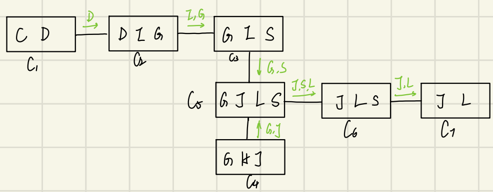
Each edge between a pair of clusters C_i and C_j is associated with a sepset $S_{ij} \subseteq C_i \cap C_j$

An execution of variable elimination defines a cluster graph



$$P(C,D,I,G,S,L,J,H) = \phi_C(C)\ \phi_D(D,C)\ \phi_I(I)\ \phi_G(G,I,D)\ \phi_S(S,I)\ \phi_L(L,G)\ \phi_J(J,L,S)\ \phi_H(H,G,J)$$

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



The message $\tau_1(D)$, generated from $\psi_1(C,D) = \phi_C(C)\ \phi_D(C,D)$, participates in the computation of ψ_2

The directions indicate the flow of messages between clusters in the execution of the variable elimination algorithm

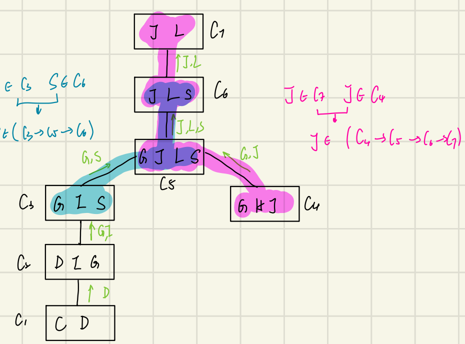
· Clique Trees (VE algorithm induces a clique tree)

The variable Elimination algorithm uses each intermediate factor only once

$\therefore$ the cluster graph induced by an execution of VE algorithm must be a tree

$\underbrace{S \in C_3 \quad S \in C_6}$

$S \in (C_3 \to C_5 \to C_6)$



Let $\mathcal{T}$ be a cluster tree over a set of factors Φ.

denote vertices by $\mathcal{V}_\mathcal{T}$ and edges by $\mathcal{E}_\mathcal{T}$

$\mathcal{T}$ has the "running intersection property" if

whenever there is a variable X s.t. $X \in C_i$ and $X \in C_j$,

X is also in every cluster in the (unique) path in $\mathcal{T}$ between C_i and C_j

when $\mathcal{T}$ is a cluster tree induced by VE algorithm.

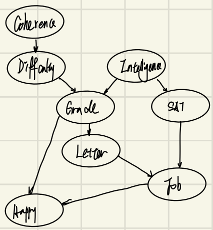
$$\begin{cases} 1.\ \mathcal{T} \text{ must have "running intersection property"} \\ 2.\ \forall \text{ neighbor clusters } C_i, C_j.\ C_i \text{ passes message } \tau_i \text{ to } C_j, \text{ then } Scope[\tau_i] = C_i \cap C_j \end{cases}$$

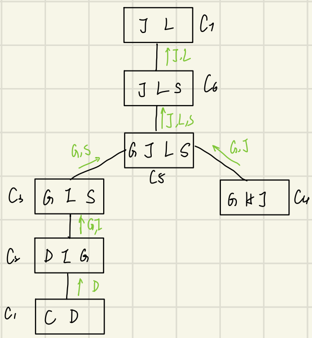
A cluster tree over factors Φ that satisfies the running intersection property is called a clique tree / junction tree / join tree

In the case of a clique tree, the clusters are also called cliques

In definition 4.17, a tree $\mathcal{T}$ is a clique tree for Markov Network $\mathcal{H}$ if:

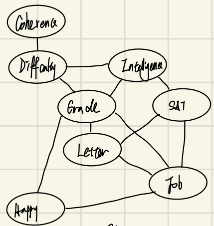
1. each node in $\mathcal{T}$ corresponds to a clique in $\mathcal{H}$, each maximal clique in $\mathcal{H}$ is a node in $\mathcal{H}$
2. each sepset $S_{i,j}$ separates $W_{<(i,j)}$ and $W_{<(j,i)}$ in $\mathcal{H}$ ($W_{<(i,j)}$ is all vars in any clique on the C_i side of $C_i - C_j$)





$\phi_C(C)\cdot\phi_D(D,C)\cdot\phi_G(D,I,G)\cdot\phi_S(S,I)$

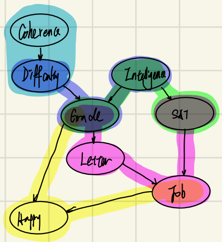
$\phi_L(L,G)\cdot\phi_J(J,L,S)\cdot\phi_H(H,J,G)$



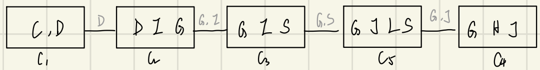
$\mathcal{H}_\Phi$

$\mathcal{T}$ satisfies the "running intersection property" iff $\forall S_{i,j}$, $W_{<(i,j)}$ and $W_{<(j,i)}$ are separated in $\mathcal{H}_\Phi$ given $S_{i,j}$

· Variable Elimination in a Clique Tree : Example



G



a possible (simple) clique tree for G

1. Generate a set of initial potentials associated with different cliques $\psi_i(C_i) = \prod_{\phi \in C_i}$

$\psi_1(C,D) = \phi_C(C) \cdot \phi_D(C,D)$ $\quad$ $\psi_4(G,H,J) = \phi_H(H|G,J)$

$\psi_2(D,I,G) = \phi_G(D,I,G)$ $\quad$ $\psi_5(G,J,S,L) = \phi_L(L,G) \cdot \phi_J(J,L,S)$

$\psi_3(G,I,S) = \phi_I(I) \cdot \phi_S(S|I)$

2.(1) Assume that we want to compute P(J). We want to do the variable elimination process so that J is not eliminated. Thus we select some clique that contains J as the root (eg. C_5)

In C_1: $\delta_{1\to2}(D) = \sum_C \psi_1(C,D)$

In C_2: $\delta_{2\to3}(G,I) = \sum_D \delta_{1\to2}(D) \cdot \psi_2(D,I,G)$

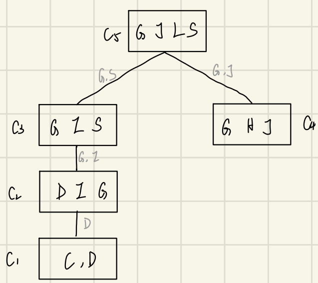
In C_3: $\delta_{3\to5}(G,S) = \sum_I \delta_{2\to3}(G,I)\ \psi_3(G,I,S)$

In C_4: $\delta_{4\to5}(G,J) = \sum_H \psi_4(G,J,H)$

In C_5: $\beta_5(G,J,S,L) = \psi_5(G,J,S,L) \cdot \delta_{4\to5}(G,J) \cdot \delta_{3\to5}(G,S)$

β_5 is the joint probability of (G,J,S,L)

$P(J) = \sum_{G,S,L} \beta_5(G,J,S,L)$



The operations in the elimination process could also have been done in another order. The only constraint is that a cliques get all of its incoming from the downstream neighbors before it sends its outgoing message toward its upstream neighbor

2.(2). choose C_4 as root

In C_1: $\delta_{1\to2}(D) = \sum_C \psi_1(C,D)$

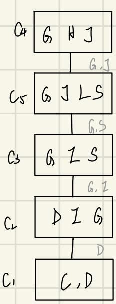
In C_2: $\delta_{2\to3}(G,I) = \sum_D \delta_{1\to2}(D) \cdot \psi_2(D,I,G)$

In C_3: $\delta_{3\to5}(G,S) = \sum_I \delta_{2\to3}(G,I)\ \psi_3(G,I,S)$

In C_5: $\delta_{5\to4}(G,J) = \sum_{S,L} \delta_{3\to5}(G,S)\ \psi_5(G,J,S,L)$

In C_4: $\beta_4(G,J,H) = \delta_{5\to4}(G,J)\ \psi_4(G,J,H)$

$P(J) = \sum_{H,G} \beta_4(G,J,H)$



def initialize_cliques(Φ: A set of factors, α: Initial assignment of factors to cliques):

for each $C_i \subseteq \alpha$:

$\psi_i(C_i) = \prod \{\phi \mid \phi \in \alpha(C_i)\}$

return $\{\psi_i(C_i)\}$

def sum_product_message(i: sending clique, j: receiving clique):

$\psi(C_i) = \psi_i \cdot \prod \{\delta_{k \to i} \mid k \in \{\text{Neighbors}(i) - j\}\}$ // initial potentials x upstream message

$\tau(S_{i,j}) = \sum_{C_i - S_{i,j}} \psi(C_i)$

return $\tau(S_{i,j})$

def clique_tree_sum_product_upward(Φ: set of factors, T: clique tree over Φ, α: initial assignment of factors to cliques, C_r: selected root clique):

$\{\phi_i(C_i)\}$ = initialize_cliques(Φ, α)

while C_r is not ready:

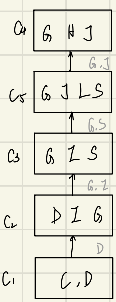
let C_i be a ready clique

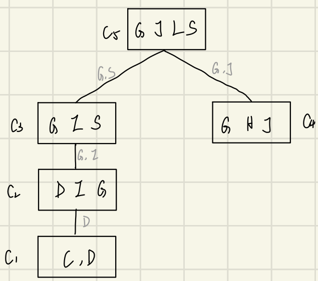
$\delta_{i \to pa(i)}(S_{i,pa(i)})$ = sum_product_message($i, pa(i)$)

$\beta_r = \psi_r \cdot \prod_{k \in \text{Neighbors}(C_r)} \delta_{k \to r}$

return β_r

Clique Tree Calibration





The message sent from C_i to C_j does not depend on specific choice of root clique, as long as the root clique is on C_i's C_j's side

in all execution of clique tree algorithm, whenever a message is sent between two cliques in the same direction, it's necessarily the same

for any given clique tree, each edge has 2 messages associated with it, one for each direction

def clique_tree_sum_product_calibrate(Φ: set of factors, T: clique tree over Φ)

$\{\psi_i\}$ = initialize_cliques(Φ, T)

while $\exists (i,j)$ st i is ready to transmit to j

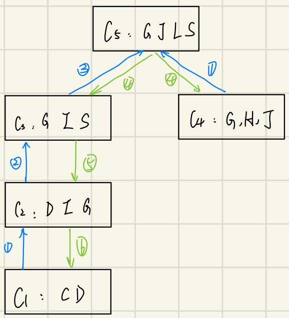
$\delta_{i \to j}(S_{ij})$ = sum_product_message(i,j)

for each clique i:

$\beta_i = \psi_i \cdot \prod_{k \in Nb(i)} \delta_{k \to j}$

return $\{\beta_i\}$

Let T be a clique tree. C_i is ready to transmit to a neighbor C_j when C_i has messages from all its neighbor except C_j



upward pass

1° C_1 is ready to transmit to C_2, C_4 ready to transmit to C_5

$\delta_{1\to2}(D) = \sum_C \psi_1(C, D)$

$\delta_{4\to5}(G,J) = \sum_H \psi_4(G,H,J) \qquad \delta = \{\delta_{1\to2}, \delta_{4\to5}\}$

2° C_2 is ready to transmit to C_3

$\delta_{2\to3}(G,I) = \sum_D \delta_{1\to2}(D) \cdot \psi_2(G,I,D) \qquad \delta = \{\delta_{1\to2}, \delta_{2\to3}, \delta_{4\to5}\}$

3° C_3 is ready to transmit to C_5

$\delta_{3\to5}(G,S) = \sum_I \delta_{2\to3}(G,I) \cdot \psi_3(G,I,S) \qquad \delta = \{\delta_{1\to2}, \delta_{2\to3}, \delta_{3\to5}, \delta_{4\to5}\}$

$\beta_5 = \psi_5(G,J,L,S) \cdot \delta_{3\to5}(G,S) \cdot \delta_{4\to5}(G,J)$

downward pass

4° C_5 is ready to transmit to C_3, C_4

$\delta_{5\to3}(G,S) = \sum_{J,L} \psi_5(G,J,L,S)\ \delta_{5\to4}(G,J) \qquad \beta_3 = \psi_3(G,I,S) \cdot \delta_{2\to3}(G,I) \cdot \delta_{5\to3}(G,S)$

$\delta_{5\to4}(G,J) = \sum_{L,S} \psi_5(G,J,L,S)\ \delta_{3\to5}(G,S) \qquad \beta_4 = \psi_4(G,H,J) \cdot \delta_{5\to4}(G,J)$

5° C_3 is ready to transmit to C_2

$\delta_{3\to2}(G,I) = \sum_S \psi_3(G,I,S) \cdot \delta_{5\to3}(G,S) \qquad \beta_2 = \psi_2(D,I,G) \cdot \delta_{1\to2}(D) \cdot \delta_{3\to2}(G,I)$

6° C_2 is ready to transmit to C_1

$\delta_{2\to1}(D) = \sum_{I,G} \psi_2(D,I,G) \cdot \delta_{3\to2}(G,I) \qquad \beta_1 = \psi_1(C,D) \cdot \delta_{2\to1}(D)$

$$\beta_i(C_i) = \sum_{\mathcal{X}-C_i} \tilde{P}_{\Phi}(\mathcal{X})$$

when this process ends, each clique contains the (unnormalized / normalized) probability

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

$$\sum_{C_i - S_{ij}} \beta_i(C_i) = \sum_{C_j - S_{ij}} \beta_j(C_j) \qquad \text{(the marginals agree)}$$

A clique tree T is calibrated if all pairs of adjacent cliques are calibrated.

For a calibrated clique tree,

$\beta_i(C_i)$ is called clique beliefs

$\mu_{ij}(S_{ij}) = \sum_{C_i - S_{ij}} \beta_i(C_i) = \sum_{C_j - S_{ij}} \beta_j(C_j)$ is called sepset beliefs

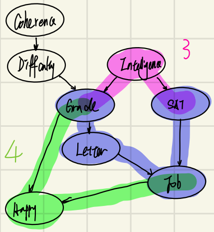
The main advantage of the clique tree algorithm is that it computes the posterior probability of all variables in a graphical model using only twice the computation of upward pass in the tree

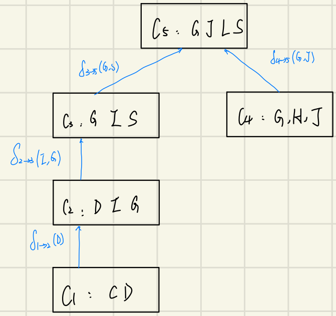
· A calibrated clique Tree as A distribution

$\beta_i = \psi_i \cdot \prod_{k \in Nb(i)} \delta_{k \to i}$ is the joint distribution of variables in clique i

$$\mu_{ij}(S_{ij}) = \sum_{C_i - S_{ij}} \beta_i(C_i)$$
$$= \sum_{C_i - S_{ij}} \psi_i \cdot \prod_{k \in Nb(i)} \delta_{k \to i}$$
$$= \sum_{C_i - S_{ij}} \psi_i \, \delta_{j \to i} \prod_{k \in \{Nb(i) - j\}} \delta_{k \to i}$$
$$= \delta_{j \to i} \sum_{C_i - S_{ij}} \psi_i \prod_{k \in \{Nb(i) - j\}} \delta_{k \to i}$$
$$= \delta_{j \to i} \cdot \delta_{i \to j}$$

(no variable in $\delta_{j \to i}$ is included in the summation)





$$\mu_{45}(G,S) = \sum_{J,L} \beta_5(G,J,L,S)$$
$$= \sum_{J,L} \psi_5(G,J,L,S) \cdot \delta_{4 \to 5}(G,J) \cdot \delta_{3 \to 5}(G,S)$$
$$= \delta_{3 \to 5}(G,S) \cdot \sum_{J,L} \psi_5(G,J,L,S) \cdot \delta_{4 \to 5}(G,J)$$
$$= \delta_{3 \to 5}(G,S) \cdot \delta_{5 \to 3}(G,S)$$

At convergence of clique tree calibration

$$\tilde{P}_\Phi(x) = \frac{\prod_{i \in V_T} \beta_i(C_i)}{\prod_{(i,j) \in E_T} \mu_{ij}(S_{ij})}$$

$$\left.\begin{aligned} \prod_{i \in V_T} \beta_i(C_i) &= \prod_{i \in V_T} \psi_i(C_i) \cdot \prod_{k \in Nb(i)} \delta_{k \to i} \\ \prod_{(i,j) \in E_T} \mu_{ij}(S_{ij}) &= \prod_{(i,j) \in E_T} \delta_{i \to j} \cdot \delta_{j \to i} \end{aligned}\right\} \frac{\prod_{i \in V_T} \psi_i(C_i) \cdot \prod_{k \in Nb(i)} \delta_{k \to i}}{\prod_{(i,j) \in E_T} \delta_{i \to j} \, \delta_{j \to i}} = \prod_i \psi_i(C_i) = \tilde{P}_\Phi$$

The clique beliefes and sepset beliefs provide a reparameterization of the unnormalized measure

This property is called the "clique tree invariant"

eg. Markov Network : A — B — C — D $(A \perp C \mid B)$

Clique tree : C_1 : A,B — C_2 : B,C — C_3 : C D

when the clique tree is calibrated, we have $\begin{cases} \beta(A,B) = \tilde{P}_\Phi(A,B) \\ \beta(B,C) = \tilde{P}_\Phi(B,C) \end{cases}$

$$\tilde{P}_\Phi(A,B,C) = \tilde{P}_\Phi(A,B)\, \tilde{P}_\Phi(C \mid A,B) = \tilde{P}_\Phi(A,B) \cdot \tilde{P}_\Phi(C \mid B) = \tilde{P}_\Phi(A,B) \cdot \frac{\tilde{P}_\Phi(B,C)}{\tilde{P}_\Phi(B)}$$
$$= \beta_1(A,B) \, \frac{\beta_2(B,C)}{\sum_C \beta_2(B,C)}$$

$\because$ $\beta_1(A,B)$ and $\beta_2(B,C)$ must agree on the marginal

$\therefore$ $\sum_A \beta_1(A,B) = \sum_C \beta_2(B,C)$

$\therefore$ $\tilde{P}_\Phi(A,B,C) = \beta_1(A,B) \, \frac{\beta_2(B,C)}{\sum_C \beta_2(B,C)} = \frac{\beta_1(A,B)}{\sum_A \beta_1(A,B)} \cdot \beta_2(B,C)$

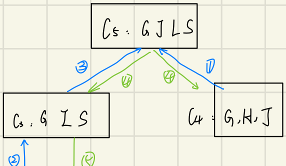
symmetry in Markov net A — B — C

define the measure induced by a calibrated tree T to be:

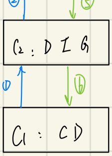
$$Q_T = \frac{\prod_{i \in V_T} \beta_i(C_i)}{\prod_{(i,j) \in E_T} \mu_{ij}(S_{ij})} \qquad \text{where } \mu_{ij} = \sum_{C_i - S_{ij}} \beta_i(C_i) = \sum_{C_j - S_{ij}} \beta_j(C_j)$$

Let T be a clique tree over Φ, and let $\beta_i(C_i)$ be a set of calibrated potentials for T.
$\tilde{P}_\Phi(X) \propto Q_T$ iff $\forall i \in V_T,\ \beta_i(C_i) \propto \tilde{P}_\Phi(C_i)$

· Beliefe update: Message passing with Division

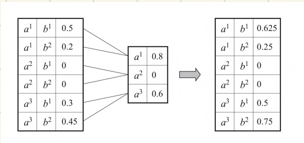


$$\left.\begin{aligned} \beta_2 &= \psi_2 \cdot \delta_{1\to2} \cdot \delta_{3\to2} \\ \delta_{2\to1} &= \sum_{I,G} \delta_{3\to2}\, \psi_2 \end{aligned}\right\} \text{similar}$$



Let X and Y be disjoint set of variables, let $\phi_1(X,Y)$ and $\phi_2(Y)$ be 2 factors
define the division $\frac{\phi_1}{\phi_2}$ be a factor

$$\psi(X,Y) = \frac{\phi_1(X,Y)}{\phi_2(Y)} \qquad \text{where } \frac{0}{0} = 0$$



a^1	b^1	0.5
a^1	b^2	0.2
a^2	b^1	0
a^2	b^2	0
a^3	b^1	0.3
a^3	b^2	0.45

a^1	0.8
a^2	0
a^3	0.6

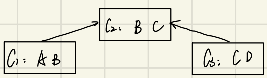
a^1	b^1	0.625
a^1	b^2	0.25
a^2	b^1	0
a^2	b^2	0
a^3	b^1	0.5
a^3	b^2	0.75

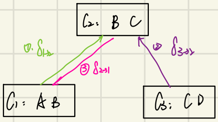
then $\psi(X,Y) \cdot \phi_2(Y) = \phi_1(X,Y)$

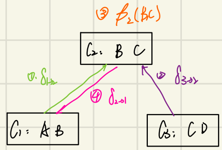
$$\delta_{i\to j} = \sum_{C_i - S_{ij}} \psi_i \cdot \prod_{k \in \{Nb(i) - j\}} \delta_{k\to i} = \sum_{C_i - S_{ij}} \psi_i \cdot \frac{\prod_{k \in \{Nb(i)\}} \delta_{k\to i}}{\delta_{j\to i}} = \frac{\sum_{C_i - S_{ij}} \beta_i}{\delta_{j\to i}}$$

eg. MN: A — B — C — D

CT:







$\delta_{1\to2}(B) = \sum_A \psi_1(A,B)$

$\delta_{3\to2}(C) = \sum_D \psi_3(C,D)$

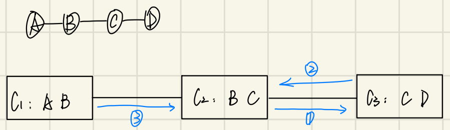
$\delta_{2\to1}(B) = \sum_C \psi_2(B,C) \cdot \delta_{3\to2}(C)$

$\beta_2(B,C) = \psi_2(B,C) \cdot \delta_{1\to2}(B) \cdot \delta_{3\to2}(C)$

$\delta_{2\to1}(B) = \sum_C \psi_2(B,C) \cdot \delta_{3\to2}(C) = \sum_C \psi_2(B,C) \frac{\delta_{3\to2}(C)\, \delta_{1\to2}(B)}{\delta_{1\to2}(B)} = \frac{\sum_C \beta_2(B,C)}{\delta_{1\to2}(B)}$

Can define the sum-product-divide message passing scheme, where each clique C_i maintains its fully updated current beliefs $\beta_i = \psi_i \cdot \prod_{k \in NB(i)} \delta_{k \to i}$. Each sepset also maintains its beliefe $\mu_{ij}(S_{ij}) = \delta_{i \to j}\, \delta_{j \to i}$. Then the entire message passing process can be executed in an equivalent way in terms of the clique and sepset beliefes

eg.



① C_3 passes an uninformed message to C_3,

Each clique C_i initializes β_i as ψ_i, and then updates it by multiplying with message updates received from its neighbors.

$\sigma_{2 \to 3}(C) = \sum_B \psi_2(B,C)$ $\quad \beta_3(C,D) = \psi_3(C,D) \sum_B \psi_2(B,C)$

$\mu_{2,3} = \sigma_{2 \to 3}$ Each sepset S_{ij} maintains μ_{ij} as the previous message passed along the edge $(i-j)$, regardless the direction

② C_3 sends a message to C_2.

$\sigma_{3 \to 2}(C) = \sum_D \beta_3(C,D)$, which is divided by $\mu_{2,3}(C)$

whenever a new message is passed along the edge, it is divided by the old message, diminating the previous message from the update to the clique

the actual update for C_2 is

$$\frac{\sigma_{3 \to 2}(C)}{\mu_{2,3}(C)} = \frac{\sum_D \beta_3(C,D)}{\mu_{2,3}(C)} = \frac{\sum_D \psi_3(C,D) \cdot \mu_{2,3}(C)}{\mu_{2,3}(C)} = \sum_D \psi_3(C,D)$$

$$\mu_{2,3}(C) = \sigma_{3 \to 2}(C) = \sum_D \beta_3(C,D) = \sum_D \psi_3(C,D) \cdot \sum_B \psi_2(B,C)$$

③ C_2 receives a message from C_1

$\sigma_{1 \to 2}(B) = \sum_A \psi_1(A,B)$, $\mu_{12} = 1$ (the initial sepset beliefe)

C_2 has received informed messages from both side and is therefore informed

$$\beta_2(B,C) = \psi_2(B,C) \cdot \left(\sum_A \psi_1(A,B)\right) \cdot \sum_D \psi_3(C,D)$$

- Calibration using beliefe propagation in clique tree

def initialize_clique_tree( ):

  for each clique C_i :

    $\beta_i = \prod \{\phi \mid \phi \in C_i\}$

  for each edge $(i-j) \in \mathcal{E}_T$

    $\mu_{ij} = 1$

def beliefe_update_message (i: sending clique, j: receiving clique):

  $\sigma_{i \to j} = \sum_{C_i - S_{ij}} \beta_i$

  $\beta_j = \beta_j \cdot \frac{\sigma_{i \to j}}{\mu_{ij}}$

  $\mu_{ij} = \sigma_{i \to j}$

```
def clique_tree_belief_update_calibrate(Φ: set of factors, T: clique tree over Φ):
    initialize_clique_tree()
    while exists an uninformed clique in T
        select (i-j) ∈ E_T
        belief_update_message(i,j)
    return {β_i}
```

1° passing the same message twice

$C_1: A\,B$ — $C_2: B\,C$ $\xrightarrow[②]{①}$ $C_3: C\,D$

①: $\delta_{2\to3}(C) = \sum_B \psi_2(B,C)$

$\beta_3(C,D) = \beta_3(C,D)\cdot \delta_{2\to3}(C)/\mu_{2,3}(C) = \psi_3(C,D)\cdot \sum_B \psi_2(B,C)$

$\mu_{2,3} = \delta_{2\to3}(C) = \sum_B \psi_2(B,C)$

② $\tilde{\delta}_{2\to3}(C) = \sum_B \psi_2(B,C) = \delta_{2\to3}(C)$

$\tilde{\beta}_3 = \beta_3(C,D)\cdot \tilde{\delta}_{2\to3}(C)/\mu_{2,3}(C) = \beta_3(C,D)$

$\tilde{\mu}_{2,3}(C) = \tilde{\delta}_{2\to3}(C) = \mu_{2,3}(C)$

we passing the same message twice,
clique beliefs and sepset beliefs are unchanged

2° pass message according partial information

$C_1: A\,B$ $\xrightarrow{②}$ $C_2: B\,C$ $\xrightarrow[③]{①}$ $C_3: C\,D$

① $\delta_{2\to3}(C) = \sum_B \psi_2(B,C)$

$\beta_3(C,D) = \psi_3(C,D)\cdot \delta_{2\to3}(C)/1 = \psi_3(C,D)\cdot \sum_B \psi_2(B,C)$

$\mu_{2,3}(C) = \delta_{2\to3}(C) = \sum_B \psi_2(B,C)$

② $\delta_{1\to2}(B) = \sum_A \psi_1(A,B)$

$\beta_2(B,C) = \psi_2(B,C)\cdot \delta_{1\to2}(B)/1 = \psi_2(B,C)\cdot \sum_A \psi_1(A,B)$

$\mu_{1,2}(B) = \delta_{1\to2} = \sum_A \psi_1(A,B)$

③ $\tilde{\delta}_{2\to3}(C) = \sum_B \beta_2(B,C) = \sum_B \psi_2(B,C)\cdot \sum_A \psi_1(A,B)$

cancel

$\tilde{\beta}_3(C,D) = \beta_3(C,D)\cdot \tilde{\delta}_{2\to3}(C)/\mu_{2,3}(C) = \frac{(\psi_3(C,D)\cdot \sum_B \psi_2(B,C))(\sum_B \psi_2(B,C)\cdot \sum_A \psi_1(A,B))}{\sum_B \psi_2(B,C)} = \psi_3(C,D)\cdot \sum_B \psi_2(B,C) \sum_A \psi_1(A,B)$

if C_i passes a message to C_j based on partial information
and then resends a more updated message,
the effect is identical to sending the updated message once

3° at convergence, all message updates have no effect

$\beta_i = \beta_i \cdot \frac{\sum_{C_j - S_{ij}} \beta_j}{\mu_{ji}}$ $\qquad \frac{\sum_{C_j - S_{ij}} \beta_j}{\mu_{ji}} = 1$

$- \cdots$ $\qquad \frac{\sum_{S_i - S_{ij}} \beta_i}{\mu_{ij}} = 1$

$\because \mu_{ij} = \mu_{ji}$ in belief propagation

$\sum_{C_j - S_{ij}} \beta_j = \sum_{C_i - S_{ij}} \beta_i$

at convergence of belief propagation, we necessarily have a calibrated tree

· Answering Queries: Incremental Updates

have a set of observations → conditioning (naive approach: conditioning the initial factors

A more efficient approach is to view the clique tree as representing the distribution $\tilde{P}_\Phi$

Assume the current distribution over $\mathcal{X}$ is defined by a set of factors Φ

$$\tilde{P}_\Phi(\mathcal{X}) = \prod_{\phi \in \Phi} \phi$$

zero out the entries in unnormalized distribution that are inconsistent with the evidence $Z = z$

$$\tilde{P}'_\Phi(\mathcal{X}) = \tilde{P}_\Phi(\mathcal{X}, Z = z) = \mathbb{1}\{Z = z\} \cdot \prod_{\phi \in \Phi} \phi$$

Assume that we have a clique tree (calibrated or not) that represents this distribution using the clique tree invariance

we can represent the distribution $\tilde{P}_\Phi(\mathcal{X})$ as

$$\tilde{P}_\Phi(\mathcal{X}) = Q_T = \frac{\prod_{i \in V_T} \beta_i(C_i)}{\prod_{(i-j) \in \mathcal{E}_T} \mu_{ij}(S_{ij})} \qquad \underbrace{\tilde{P}'_\Phi(\mathcal{X}) = \mathbb{1}\{Z = z\} \cdot \frac{\prod_{i \in V_T} \beta_i(C_i)}{\prod_{(i,j) \in \mathcal{E}_T} \mu_{ij}(S_{ij})}}$$

multiply in a new factor $\mathbb{1}\{Z = z\}$ in some clique C_i that contains variable Z

if the clique tree is calibrated before the new factor is introduced, then C_i has all incoming messages

the entire tree can be recalibrated with a single pass

eg. Recalibrate a calibrated clique tree

β_1 β_2 β_3

$\boxed{C_1: A\ B} \overset{\mu_{12}}{—} \boxed{C_2: B\ C} \overset{\mu_{23}}{—} \boxed{C_3: C\ D}$

A	B

B	C	
b^1	c^1	
b^1	c^2	
b^2	c^1	0
b^2	c^2	0

$\uparrow \mathbb{1}\{B = b\}$

C	D

$\beta_1(A,B) = \tilde{P}_\Phi(A,B) \qquad \beta_2(B,C) = \tilde{P}_\Phi(B,C) \qquad \beta_3(C,D) = \tilde{P}_\Phi(C,D)$

$\mu_{12} = \sum_A \tilde{P}_\Phi(A,B) = \tilde{P}_\Phi(B) \qquad \mu_{23} = \tilde{P}_\Phi(C)$

$$\beta_2' = \mu_{12}(B) \cdot \mu_{23}(C) \cdot \prod_{\phi \in \Phi} \phi \cdot \mathbb{1}\{B = b\}$$

$$\delta'_{2\to1}(B) = \sum_{C_2 - S_{12}} \beta_2'(B,C) = \sum_C \tilde{P}_\Phi(B=b, C) =$$

B	
b^1	
b^2	0

$$\beta_1' = \beta_1 \cdot \frac{\delta'_{2\to1}}{\mu_{12}} = \tilde{P}_\Phi(A,B) \cdot \frac{\tilde{P}_\Phi(B=b)}{\tilde{P}_\Phi(B)} =$$

$\beta_1 =$

A	B	
a^1	b^1	:
a^1	b^2	:
a^2	b^1	:
a^2	b^2	:

$\beta_1' =$

A	B	
a^1	b^1	.
a^1	b^2	0
a^2	b^1	.
a^2	b^2	0

$$\delta'_{2\to3}(C) = \sum_B \tilde{P}_\Phi(B=b, C) = \tilde{P}_\Phi(B=b, C)$$

$$\beta_3' = \beta_3 \cdot \frac{\delta'_{2\to3}(C)}{\mu_{23}(C)} = \tilde{P}_\Phi(C,D) \cdot \frac{\tilde{P}_\Phi(B=b, C)}{\tilde{P}_\Phi(C)}$$

$$\mu_{23}' = \delta'_{2\to3} = \tilde{P}_\Phi(B=b, C)$$

B	C	
b^1	c^1	.
b^1	c^2	.
b^2	c^1	0
b^2	c^2	0

δ'_{23}

· Answering Queries : Queries outside a clique

perform variable elimination over a calibrated clique tree

if the entire clique tree is calibrated, so is any connected subtree $\mathcal{T}'$

eg. Ⓐ—Ⓑ—Ⓒ—Ⓓ

$\boxed{C_1: A\ B}$ — $\boxed{C_2: B\ C}$ — $\boxed{C_3: C\ D}$

the clique tree is calibrated so that $\tilde{P}_{\Phi}(A,B,C,D) = \dfrac{\prod_{C_i \in V_{\mathcal{T}}} \beta_i(C_i)}{\prod_{(i,j)\in \mathcal{E}_{\mathcal{T}}} \mu_{ij}(S_{ij})}$

let $\mathcal{T}' =$ $\boxed{C_2: B\ C}$ — $\boxed{C_3: C\ D}$

$\tilde{P}_{\Phi}(B,C,D) = Q_{\mathcal{T}'}$

$$\tilde{P}_{\Phi}(B,D) = \sum_C \tilde{P}_{\Phi}(B,C,D)$$

$$= \sum_C \frac{\beta_2(B,C)\cdot\beta_3(C,D)}{\mu_{23}(C)}$$

$$= \sum_C \tilde{P}_{\Phi}(B|C)\cdot\tilde{P}_{\Phi}(C,D)$$

def out_of_clique_query($\mathcal{T}$, $\{\beta_i\}$, $\{\mu_{ij}\}$, Y): Y is query variables that's not in any single clique

let $\mathcal{T}'$ be a subtree of $\mathcal{T}$ such that $Y \subseteq Scope[\mathcal{T}']$

select a clique $r \in V_{\mathcal{T}'}$ to be the root

$\Phi = \beta_r$

for each $i \in V_{\mathcal{T}'} - \{r\}$:

$\phi \leftarrow \dfrac{\beta_i}{\mu_{i,p_r(i)}}$

$\Phi = \Phi \cup \{\phi\}$

$Z = Scope[\mathcal{T}'] - Y$

return sum_product_variable_elimination(Φ, Z)

we do not have to perform inference over the entire clique tree,
but only over a portion of the tree that contains the variables Y that constitute our query