Recap.

Inference Algorithms

	Marginalization	Maximization			
√	Elimination Algo	Elimination Algo	Exact,	any G,	$O(\lvert\mathcal{X}\rvert^{\text{Treewidth}})$ at best but NP-hard to find ordering
√	sum-product = Belief Propagation	Max-product	Approx.,	Pairwise MRF,	$O(\lvert\mathcal{X}\rvert^2)$
√	Sum-product algorithm on factor graphs	Max-product algo on factor graphs	Approx,	any G,	$O(\lvert\mathcal{X}\rvert^{\text{max-degree}})$
Today	Elimination algorithm on Junction trees	Elimination Algorithm on Junction trees	Exact,	Junction Trees,	$O(\lvert\mathcal{X}\rvert^{\text{Treewidth}})$ NP-hard to find the best ordering.

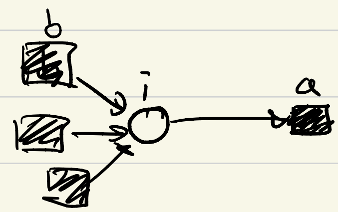
An exercise to derive Pairwise MRF BP, get 2 different looking updates.

BP for Factor Graphs.

Repeat $t = 1 \dots T$

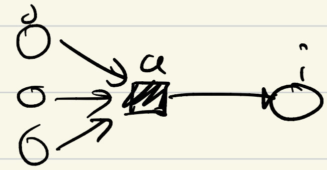
Update $m_{i\to a}$'s :

$$m_{i\to a}(x_i) = \prod_{b\in\partial i\setminus\{a\}} \tilde{m}_{b\to i}(x_i)$$



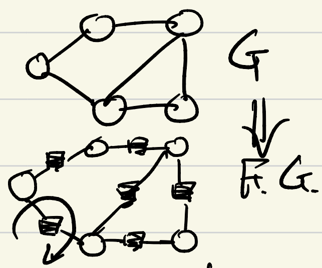
Update $\tilde{m}_{a\to i}$'s :

$$\tilde{m}_{a\to i}(x_i) = \sum_{\partial a\setminus\{i\}} f_a(x_{\partial a}) \prod_{j\in\partial a\setminus\{i\}} m_{j\to a}(x_j)$$



Let's apply it to Pairwise MRF:

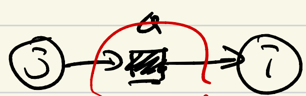
$$p(x) = \frac{1}{Z}\prod_{(i,j)\in E} f_{ij}(x_i, x_j)$$



degree on factornode is 2

Update: $\boxed{m_{i\to a}(x_i)} = \prod_{b\in\partial i\setminus\{a\}} \boxed{\tilde{m}_{b\to i}(x_i)}$

$$\boxed{\tilde{m}_{a\to i}(x_i)} = \sum_{x_j} f_{ij}(x_i, x_j) \boxed{m_{j\to a}(x_j)}$$

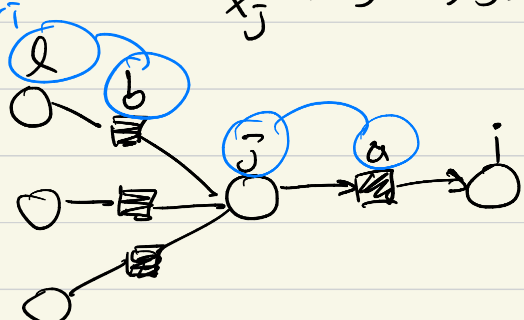


degree=2

let's try to simplify by ① plug-in $m_{i\to a}$'s

$$\tilde{m}_{\underset{j\to i}{\cancel{a\to i}}}(x_i) = \sum_{x_j} f_{ij}(x_i, x_j) \cdot \prod_{\underset{l\in\partial j\setminus\{i\}}{\cancel{b\in\partial j\setminus\{a\}}}} \tilde{m}_{\underset{l\to j}{\cancel{b\to j}}}(x_j)$$

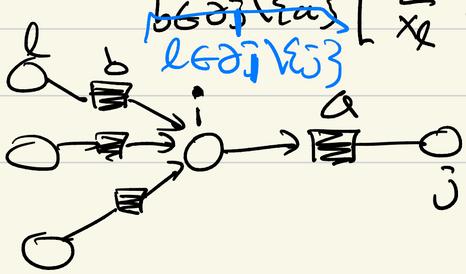
= exactly BP for Pairwise MRF



$\tilde{m}_{i\to j}(x_j)$
↑
target

let's try to simplify by ② plug-in $\tilde{m}_{a\to i}$'s

$$m_{\underset{i\to j}{\cancel{i\to a}}}(x_i) = \prod_{\underset{l\in\partial j\setminus\{i\}}{\cancel{b\in\partial i\setminus\{a\}}}} \left\{ \sum_{x_l} f_{li}(x_l, x_i) \cdot m_{\underset{l\to i}{\cancel{l\to b}}}(x_l) \right\}$$



$m_{i\to j}(x_i)$
↑
parent/Source

* Junction Tree Algorithm : Elimination algorithm exact on Junction Tree

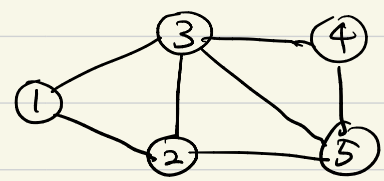
any G can be turned into a Junction tree (at a loss)

* If G is chordal, we can easily find a Junction Tree (Special case of MRF) and solve inference exactly.

* If G is not chordal, we add edges to make G chordal

↳ increases complexity.

MRF: $P(x) = \frac{1}{Z} f_{123}(x_1,x_2,x_3)\, f_{235}(x_2,x_3,x_5)\, f_{345}(x_3,x_4,x_5)$



Def. clique tree of MRF $G=(V,E)$.

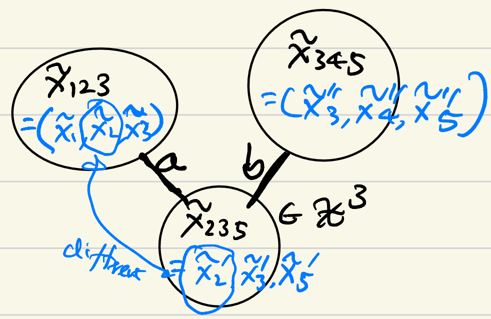
one example : Clique Tree

- Create a super-node for each clique $c \in \mathcal{C}$
- assign variable $\tilde{x}_c \in \mathcal{X}^{|c|}$

  each node has a local copy of global x_i's.

- assign edges to form a tree.

  each edge ensures that the local copies are consistent.



Clique Tree Model $T=(\tilde{V},\tilde{E})$ w.r.t $G=(V,E)$

$$\tilde{P}(\tilde{x}_{123},\tilde{x}_{235},\tilde{x}_{345}) = \frac{1}{Z} f_{123}(\tilde{x}_{123}) \cdot f_{235}(\tilde{x}_{235})\, f_{345}(\tilde{x}_{345})$$

$= f_{123}(\tilde{x}_1, \boxed{\tilde{x}_2}, \tilde{x}_3)$ $\quad (\boxed{\tilde{x}_2'}, \tilde{x}_3', \tilde{x}_5')$

not the same

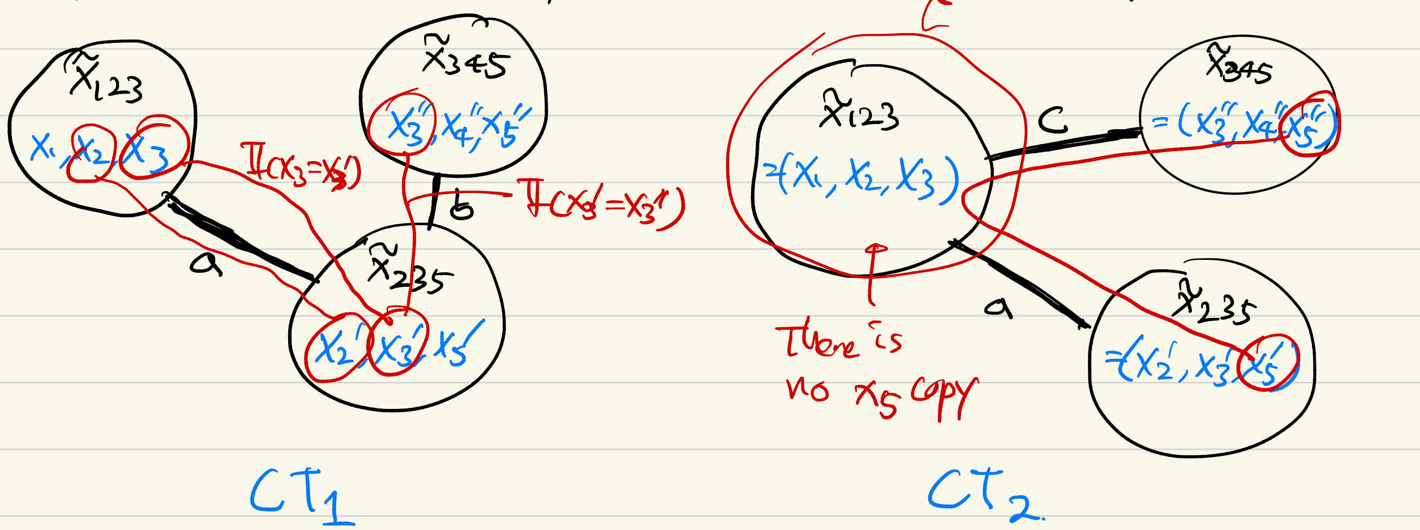
$$\times\ f_a(\tilde{x}_{123}, \tilde{x}_{235}) \qquad \times\ f_b(\tilde{x}_{235}, \tilde{x}_{345})$$

$\| \qquad\qquad \|$

$\mathbb{I}(\tilde{x}_2 = \tilde{x}_2')\cdot \mathbb{I}(\tilde{x}_3 = \tilde{x}_3') \qquad \mathbb{I}(\tilde{x}_3' = \tilde{x}_3'')\cdot \mathbb{I}(\tilde{x}_5' = \tilde{x}_5'')$

$$= \frac{1}{Z}\ \boxed{\tilde{f}_a(\tilde{x}_{123}, \tilde{x}_{235})} \cdot \boxed{\tilde{f}_b(\tilde{x}_{235}, \tilde{x}_{345})}$$

example when Global consistency is not achievable (depending on the tree)



x_5 cannot be made consistent

Claim:

If CT is globally consistent, for each variable i, remove all super-nodes not containing local copy of x_i, and the resulting graph is still connected

then,

$$\underset{\text{marginal}}{\tilde{P}_{CT}(\tilde{X}_{123})} = \underset{\text{marginal}}{P_{MRF}(X_1, X_2, X_3)}$$

Q. When ~~is a~~ $\exists$ clique tree of MRF $G=(V,E)$ globally consistent? [existence]

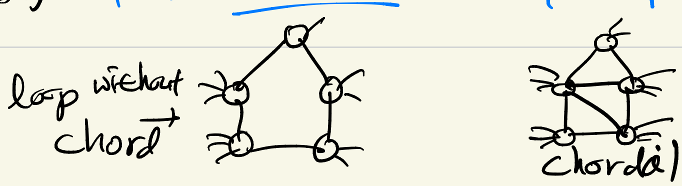
Def. Junction Tree Property.

a clique tree $T=(\tilde{V},\tilde{E})$ from $G=(V,E)$ satisfy J.T.P if all $\underset{\text{super nodes}}{\text{cliques}}$ contain i form a connected subtree in T.

Def. T is a junction tree for G if all $\underset{x_i}{i \in V}$ has J.T.P

Q. [Existence] when does G have a Junction Tree?

If G is chordal ($\hat{=}$ any loop ≥ 4 has a chord)



Q. [Construction] Given a chordal G, how do we find it? ①

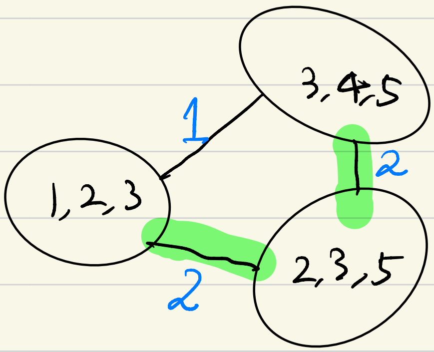
Q. If the graph is not chordal, what do we do? ②

① Sketch of Algorithm : to find J.T for $G=(V,E)$, $\mathcal{C}$ set of max cliques

- Create a complete graph on supernodes
- assign weights to each edge as # variables shared btw two cliques.
- find the maximum weight spanning tree

$\triangleq$ a tree whose sum of edge weights is maximum

cf. Kruskal's.



Claim: [Correctness]

A clique tree T is a Junction tree if and only if it is a max. spanning tree.

$$\underset{\text{for any tree}}{w[T]} = \sum_{(C_\ell,C_r)\in T} \underbrace{|C_\ell \cap C_r|}_{\text{weight on edge } (C_\ell, C_r)}$$

$$= \sum_{(C_\ell,C_r)\in T} \sum_{i\in V} \mathbb{I}(X_i \in C_\ell \,\&\, X_i \in C_r)$$

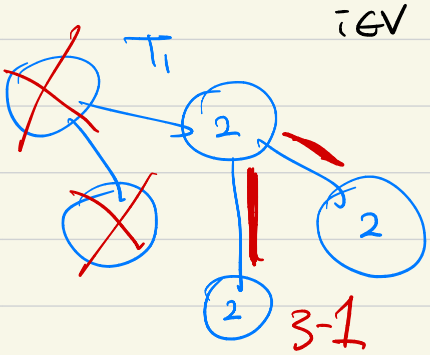
$$= \sum_{i\in V} \boxed{\sum_{(C_\ell,C_r)\in T} \mathbb{I}(X_i\in C_\ell \,\&\, X_i\in C_r)}$$

$\leq (\underbrace{\text{\#cliques containing } X_i}_{m_i}) - 1$

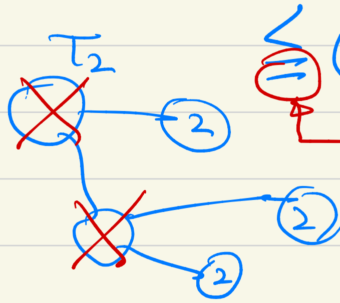
$=$ iff subgraph containing cliques with X_i form a tree

$\Updownarrow$

If T is a Junction Tree



3-1



0.

$$\leq \boxed{\sum_{i\in V} (m_i - 1)}$$

② What if G is not chordal?

Sketch Algorithm.

make G chordal.
- choose an elimination ordering π
- find reconstituted graph. (which is chordal)
- Construct clique nodes complete
- max-weight spanning tree.

at best π, $O(|\mathcal{X}|^{\text{Treewidth}})$