Recapping Inference Algs

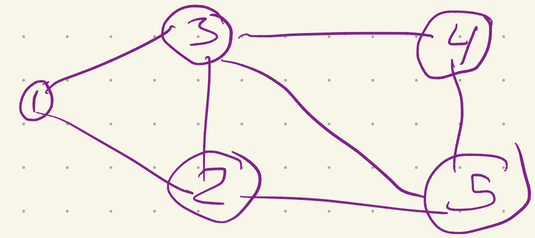
Marginalization	Maximization	Exact/Apx	Which GMs?	C.C.
Elimination Alg		Exact	Any G	$O(\lvert\chi\rvert^{TW})$ or worse
Sum-product = Belief Prop	Max-product	Apx	Pairwise MRFs	$O(\lvert\chi\rvert^{2})$
Sum-product on FGs = Belief Prop	Max-product on FGs	Apx	any FG	$O(\lvert\chi\rvert^{\text{max-deg}})$
Elimination Alg on junction trees		Exact	Junction trees	$O(\lvert\chi\rvert^{TW})$ or worse

Junction Tree Alg : Elim on A junction $\boxed{\text{tree}}$ → exact

- Fixing the tree, inference is easy using most methods
- Similar to elimination alg [exact, ordering]
- the data structure is meant to support efficient elimination

- MRF ⟶ Clique tree
  [non-unique]

$\boxed{M}$



One example clique tree for M

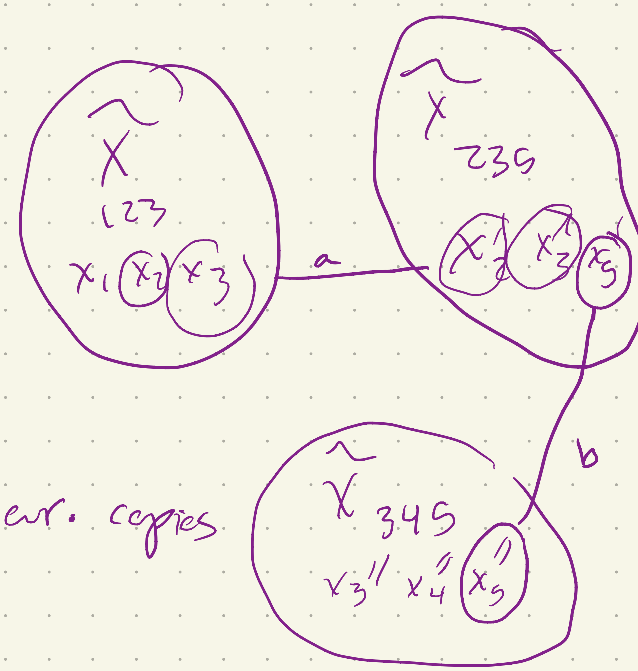
$$P(X) \propto f_{123} \, f_{235} \, f_{345}$$

– Create a joint node for each clique
  $\tilde{x}_C \in \mathcal{X}^{|C|}$

  - each node has a local copy of its vars

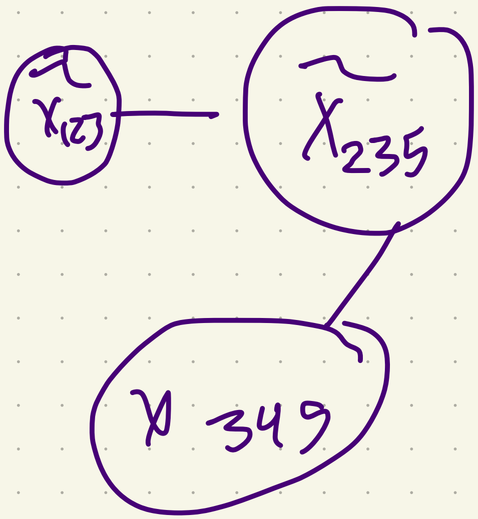
$$\tilde{x}_{123} = (x_1, x_2, x_3)$$
$$\tilde{x}_{235} = (x_2', x_3', x_5')$$
$$\tilde{x}_{345} = (x_3'', x_4'', x_5'')$$

  - Assign edges to form a tree
    & ensure consistency across var. copies



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

- $\mathbb{1}$[local copies are = across edges]



:= $\mathbb{1}[\tilde{x}_{123}$'s copy of $x_2 = \tilde{x}_{235}$'s copy of 2]

- $\mathbb{1}$[" " $x_3 =$ " of 3]
- $\mathbb{1}[x_{235}$'s $x_3 = x_{345}$'s $x_3]$
- $\mathbb{1}$[" $x_5 =$ " x_5]

Global consistency: where my edge set allows me to enforce all copies of x_i are =

$\forall$ variables x_i.

Q: When will a tree support global consistency?

Def | Junction tree property |

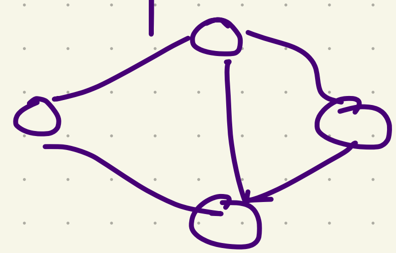
A MRF $G=(V,E)$ with $\mathcal{C}$, set of maximal cliques

A tree T over $\mathcal{C}$ satisfies the JTP w.r.t. vertex $i \in V$ if all cliques containing i form a connected subtree in T.

A tree is a junction tree if it has JTP $\forall i \in V$.

Q When does a MRF G have a JT?

A: When G is chordal. ($\forall$ loops $\in G$ of size ≥ 4, the loop has a chord)



We can compute a JT for a chordal graph G.

Sketch for finding JTs for a MRF that is chordal

- Consider the complete graph on "$\mathcal{C}$"
- Assign wts to the edges (# of shared vars)
- Find a max-wt spanning tree of $\mathcal{G}$

Claim: A clique tree T is a junction tree $\iff$ it's a max-wt spanning tree of this clique graph.

Pf sketch

$\forall$ trees T

$$\omega(T) = \sum_{(C_k, C_\ell) \in T} |C_\ell \cap C_k|$$

$$= \sum_{(C_k, C_\ell) \in T} \sum_{i \in V} \mathbb{1}[i \in C_\ell \cap i \in C_k]$$

$$= \sum_{i \in V} \sum_{(C_\ell, C_k) \in T} \text{''}$$

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

$\Rightarrow$ # of cliques containing vertex i

= when these form a connected subtree T_i

If G isn't chordal

Choose an elim ordering

Make G chordal

(Replace w reconstituted graph G'

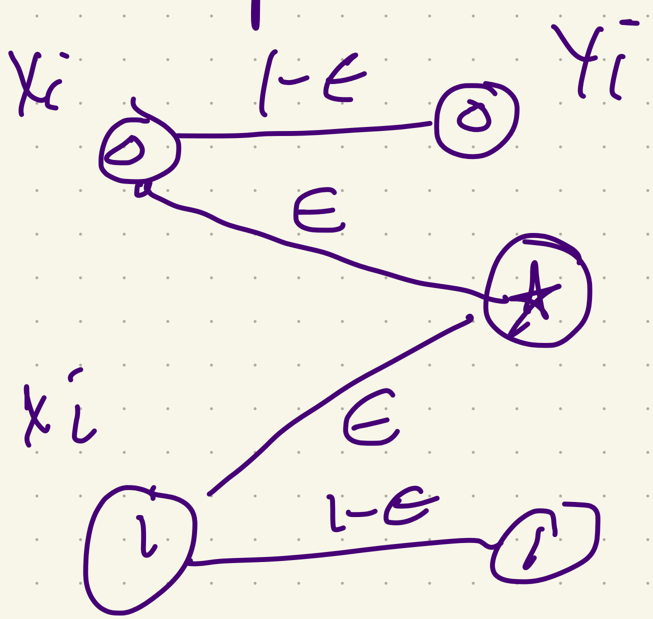
then repeat for now chordal G':

find JT on Complete weight cliquegraph

When does BP peform well?

- Exact on trees
- On a graph w 1 cycle, converges but might be wrong.
- In the limit of a large (usually sparse) graph density estimation provides an asymptotic apx to BP performance.

Example of density estimation in LDPC



Erasure channel w.e.p. ϵ

When does BP "work" for erasure channels?

↳ When does its output = (whp) the original x?

↳ what fraction of $\hat{x} = x$ (bitwise)

Start w a random factor graph for a LDPC

$$RG(n, m, l = [l_1, \ldots, l_n], r = [r_1, \ldots r_m])$$

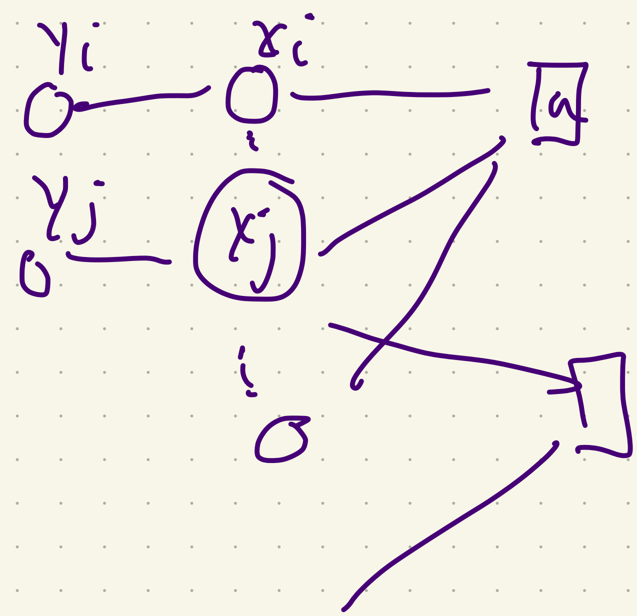
$P[y_i = * \mid x_i] = \epsilon$

$P[y_i = x_i \mid x_i] = 1-\epsilon$

BP on this FG

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

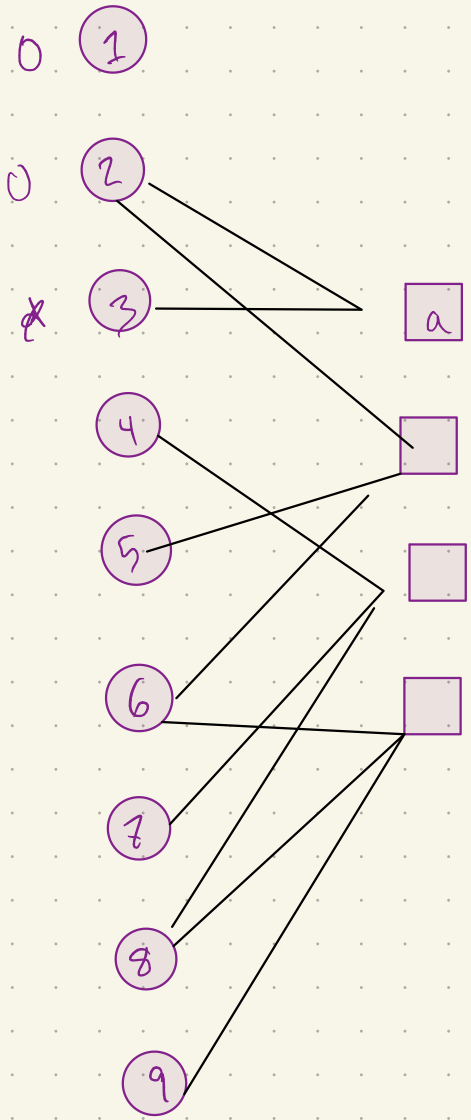
$$\tilde{m}_{a\to i}(x_i) = \sum_{x_{\partial a\setminus\{i\}}} \prod_{j\in\partial a\setminus\{i\}} m_{j\to a}(x_j)\, \mathbb{I}(\oplus x_{\partial a} = 0)$$

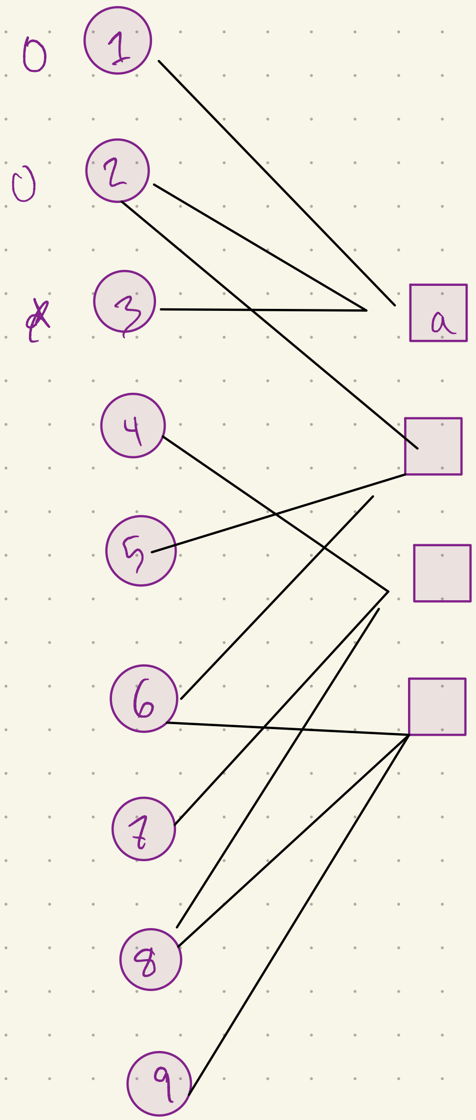


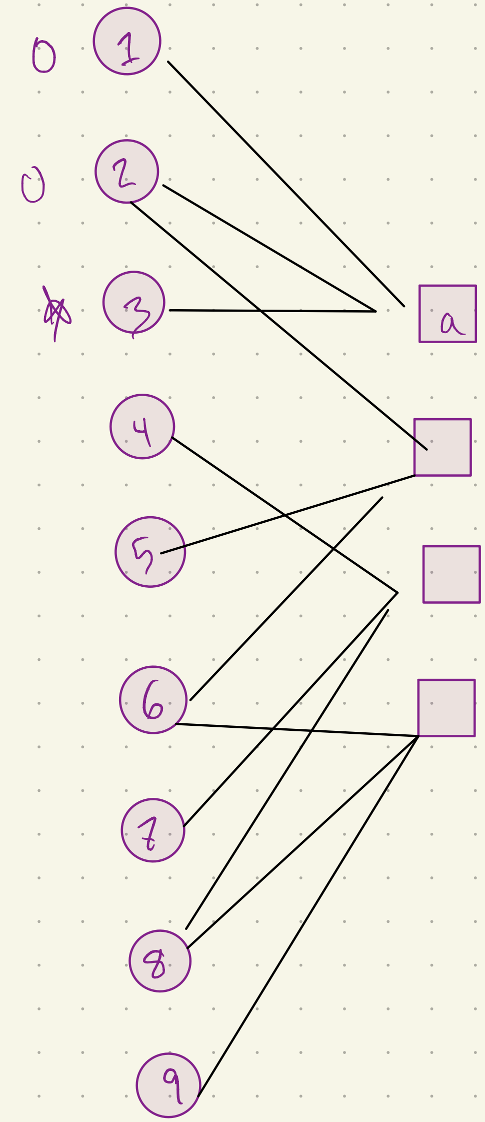
Claim:

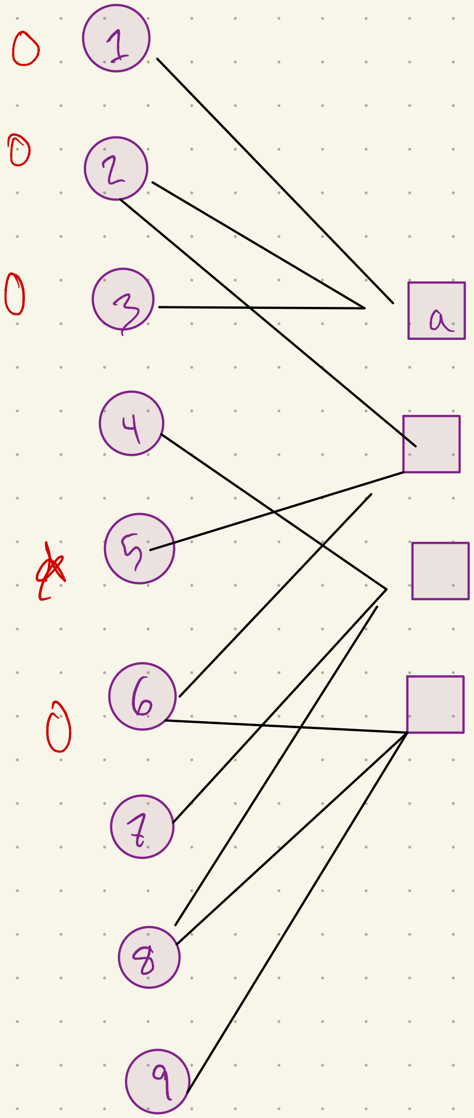
BP keeps $m_{i\to a}$, $\tilde{m}_{a\to i} \in \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1/2\\1/2\end{bmatrix}$

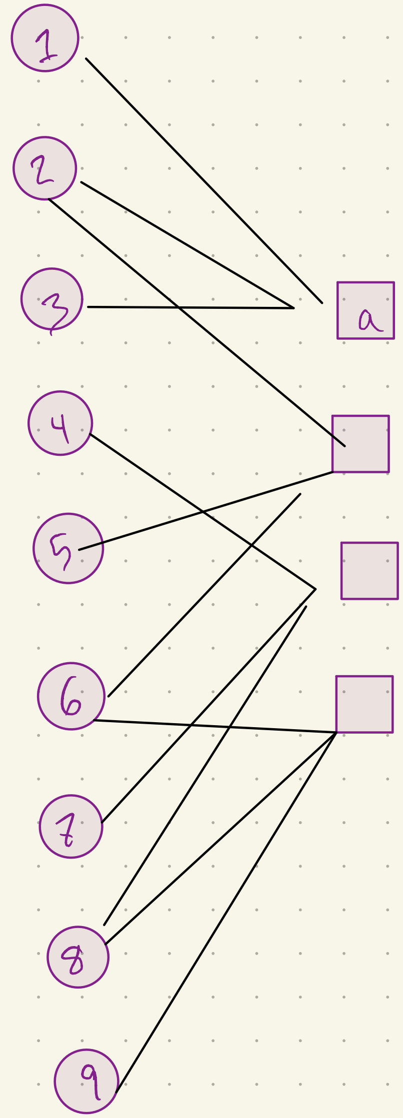
Suppose $\bar{x} = 0$








0
1
0
2
0
3
4
5
0
6
7
8
9
a


1
2
3
4
5
6
7
8
9
a