IPF proof — Each step maximizes log-likelihood over ψ_c alone

— Coordinate ascent ψ_D, ψ_c

$$\ell(\theta|D) = \sum_c \sum_{x_c} m(x_c) \log \psi_c(x_c) - N \log Z$$ ← function of $\psi_1, \dots, \psi_c$

write single table entry

$\psi_c(x_c)$

$$\frac{\partial \ell}{\partial \psi_c(x_c)} = \frac{m(x_c)}{\psi_c(x_c)} - \frac{N}{Z} \frac{\partial}{\partial \psi_c(x_c)} \left[\sum_{\tilde{x}} \prod_D \psi_D(\tilde{x}_D) \right]$$

only hits terms in sum where table entry active

$$= \frac{m(x_c)}{\psi_c(x_c)} - \frac{N}{Z} \sum_{\tilde{x}} \delta(\tilde{x}_c, x_c) \frac{\partial}{\partial \psi_c(x_c)} \left[\prod_D \psi_D(\tilde{x}_D) \right]$$

$$= \frac{m(x_c)}{\psi_c(x_c)} - \frac{N}{Z} \sum_{\tilde{x}} \delta(\tilde{x}_c, x_c) \prod_{D \neq c} \psi_D(\tilde{x}_D)$$ ← locked at old t

$$\frac{\partial \ell}{\partial \psi_c(x_c)} \left(\psi_c^{(t+1)}(x_c)\right) = \frac{m(x_c)}{\psi_c^{(t+1)}(x_c)} - \frac{N}{Z^{(t+1)}} \sum_{\tilde{x}} \delta(\tilde{x}_c, x_c) \prod_{D \neq c} \psi_D^{(t)}(\tilde{x}_D)$$ ← Z const

$$= \frac{m(x_c)}{\psi_c^{(t+1)}(x_c)} - \frac{N}{Z^{(t)}} \sum_{\tilde{x}} \delta(\tilde{x}_c, x_c) \prod_{D \neq c} \psi_D^{(t)}(\tilde{x}_D)$$

$$= \frac{m(x)}{\psi_c^{(t+1)}(x_c)} - \frac{N}{\psi_c^{(t)}(x_c)} \sum_{\tilde{x}} \delta(\tilde{x}_c, x_c) \frac{1}{Z^{(t)}} \prod_D \psi_D^{(t)}(\tilde{x}_D)$$

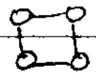
$$= \frac{m(x)}{\psi_c^{(t+1)}(x_c)} - \frac{N}{\psi_c^{(t)}(x_c)} \sum_{\tilde{x}} \delta(\tilde{x}_c, x_c)\, p(\tilde{x})$$

M-step

$$= \frac{m(x)}{\psi_c^{(t+1)}(x_c)} - \frac{N}{\psi_c^{(t)}(x_c)}\, p(x)$$

example

$\psi_c(x_c) =$



$$= \frac{m(x)}{\psi_c^{(t)}(x_c)} \frac{p(x)}{\tilde{p}(x)} - \frac{N\, p(x)}{\psi_c^{(t)}(x_c)} = \frac{N\, p(x)}{\psi_c^{(t)}(x_c)} - \frac{N\, p(x)}{\psi_c^{(t)}(x_c)} = 0$$

summary: have efficient algo's for $A \perp B | C$, $p(y|x)$ & Max. Likelihood

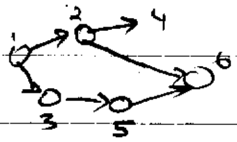
Junction Tree Algorithm: — General inference tool for graphical mod.

— Workhorse, inference cpdf is main tool

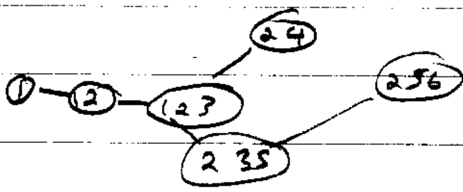
— Node elimination algo must be repeated for each query
redundant work gets discarded & repeated.
instead look closely at cliques we are eliminating
connect them & their separators.

— Tree: unique path between vertices, root node.

from node elimination

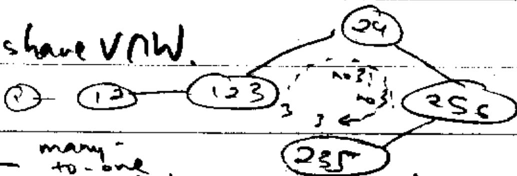


cliques $\{X_2, X_5, X_6\}$
$\{2,3,5\}$
$\{2,4\}$
$\{1,2,3\}$
$\{1,2\}$
$\{1\}$



- junction tree property: each node # appears connected (in a path) to its other instantiations through the tree & all subtrees.

  on unique path from V to W, all nodes have $V \cap W$.



- more generally: undirected graphs not on maximal cliques: $p(x) = \frac{1}{Z} \prod_{C \in C} \psi_C(x_C)$

many-to-one then. But once we choose, it's ok

- Want to generally compute $p(X_F \mid X_E)$ or $p(X_F \mid \bar{X}_E)$
  - over a set of nodes (clique for now)
- Efficiently infer marginals & conds.
- for a single clique

  $p(X_C, \bar{X}_E)$ then get $p(\bar{X}_E) = \sum_{X_C} p(X_C, \bar{X}_E)$

  then $p(X_C \mid \bar{X}_E) = \dfrac{p(X_C, \bar{X}_E)}{\sum_{X_C} p(X_C, \bar{X}_E)}$

example

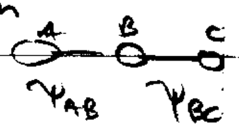
$p(x) = p(X_A)\, p(X_B \mid X_A)\, p(X_C \mid X_B)$

easily can get $p(A,B)$ by $p(X_A)\, p(X_B \mid X_A)$

but how to get $p(B,C)$?

First $p(x) = \prod_i p(x_i \mid x_{\pi_i}) = \frac{1}{Z} \prod_C \psi_C(X_C)$

$\longrightarrow$ moralization



Review:
- Assignment 1, 2 coming
- Assignment 3 online, Projects online
- Max-Likelihood for directed, undirected
- decomposable, I.P.F.

Junction Trees Continued: want local marginals everywhere — easy

$\prod_i p(x_i|x_{\pi_i}) = \frac{1}{Z}\prod_C \psi_C = A \to B \to C$

Can write $\psi_{AB} = p(x_A)\,p(x_B|x_A) = p(x_A, x_B)$

$\psi_{BC} = p(x_C|x_B) \neq p(x_B, x_C)$

how to get?

- Would like all cliques
- More flexible notation needed:

$$p(x) = \frac{1}{Z}\,\frac{\prod_C \psi_C(x_C)}{\prod_S \phi_S(x_S)}$$

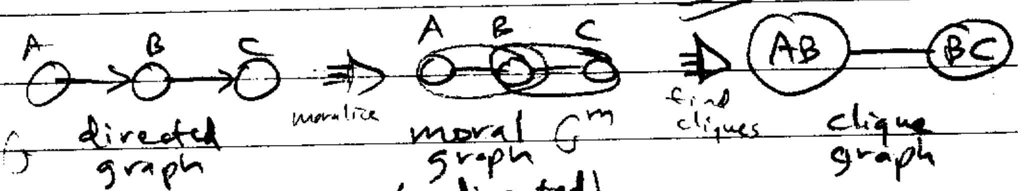
← cliques

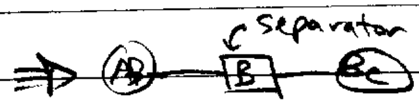
← separators (between all clique-paths)

Z=1 in practice; put this in a $\psi_S(x_S)$ over null set

- Separators are function of intersection of neighbouring cliques
- Superset, if all $\psi_S(x_S) := 1$ get previous.
- But doesn't span anything new since any separator is subset of clique

TRIANGULATE



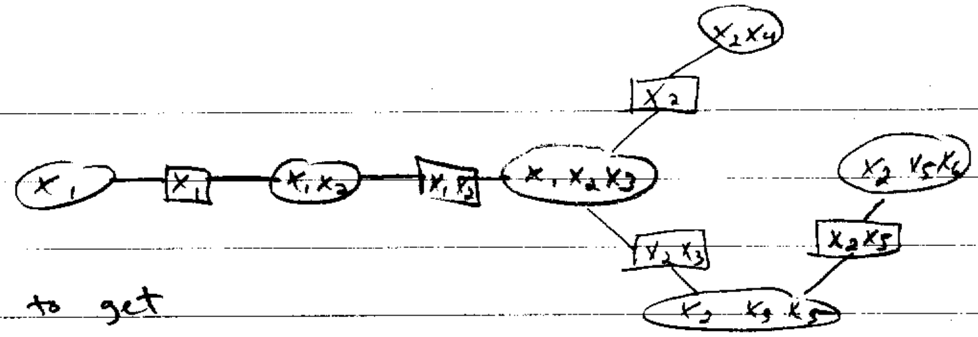


Junction Tree

now have nice marginals upstairs can just read off.

$$p(x) = \frac{\overbrace{p(A,B)}^{\psi_{AB}}\;\overbrace{p(B,C)}^{\psi_{BC}}}{\underbrace{p(B)}_{\phi_B}}$$

⚠ note division by zero can occur, but numerator will also be 0 so treat $\frac{0}{0} = 0$ since $p(B) = \sum_{x_C} p(B,C)$

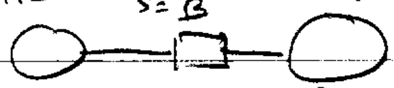


How to get
the product over marginals form from $p(x_i | x_{\pi_i})$?

Junction Tree Algorithm — Message Passing between cliques $\Rightarrow$ consistency
To get marginals with $p(x)$ staying consistent

$V = AB$ $\quad S = B$ $\quad W = BC$

mini-example



marginal $\quad \phi_S^* = \sum_{V \setminus S} \psi_V \quad \& \quad \phi_S^* = \sum_{W \setminus S} \psi_W$

i.e. $p(B) = \sum_A p(A,B) \qquad$ i.e. $p(B) = \sum_C p(B,C)$

if ψ_V is valid $p(A,B)$ can stop

message from V to W

$$\phi_S^* = \sum_{V \setminus S} \psi_V$$
$$\psi_W^* = \frac{\phi_S^*}{\phi_S} \psi_W$$
$$\psi_V^* = \psi_V$$

$\therefore$ $p(x)$ unchanged

$$\frac{\psi_V^* \psi_W^*}{\phi_S^*} = \frac{\psi_V \psi_W \phi_S^*}{\phi_S \phi_S^*} = \frac{\psi_V \psi_W}{\phi_S}$$

if ψ_W is valid $p(B,C)$ can stop

message from W to V

$$\phi_S^{**} = \sum_{W \setminus S} \psi_W^*$$
$$\psi_V^{**} = \frac{\phi_S^{**}}{\phi_S^*} \psi_V^*$$
$$\psi_W^{**} = \psi_W^*$$

$\therefore$ $p(x)$ unchanged AND

$$\sum_{V \setminus S} \psi_V^{**} = \sum_{V \setminus S} \frac{\phi_S^{**}}{\phi_S^*} \psi_V^* = \frac{\phi_S^{**}}{\phi_S^*} \sum_{V \setminus S} \psi_V^* = \phi_S^{**} = \sum_{W \setminus S} \psi_W^{**}$$

Can "pop" these out now

$\longrightarrow$ full clique margs. V & W agree with marginal on S

mini-example 2

$x \to y \to z \;=\; (xy) - \boxed{y} - (yz)$

init $\begin{cases} \psi_{xy} = p(y|x)p(x) = p(x,y) \\ \psi_{yz} = p(z|y) \\ \phi_y = 1 \end{cases}$

$\phi_y^* = \sum_x p(x,y) = p(y)$

$\psi_{yz}^* = \frac{p(y)}{1} p(z|y) = p(y,z)$ end.

with evidence $\quad X := 1$. Init as above.

$\phi_y^* = \sum_x p(x,y)\, \delta(x = \bar{x}) = p(x=1, y) \qquad \psi_{yz}^* = \frac{p(X=1,y)}{1} p(z|y) = p(x=1,y,z)$

and $\psi_{xy}^* = p(x=1,y)$ as before end.

to get conds: $p(y,z | x=1) = \frac{\psi_{yz}^*}{\sum_{yz} \psi_{yz}^*}$

- Message Passing Makes clique marginals with proper intersections
- On big junction tree, can keep iterating messages but inefficient

- Send message only after hearing from all neighbours

<u>J.T.A.</u> - No need to iterate mindlessly.

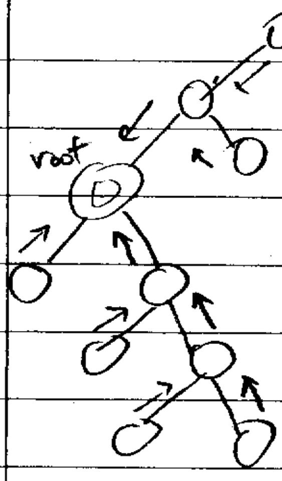
- initialize: - pick a root
  - $\phi_S = 1$ for all cliques
  - $\psi_c(x_c) = p(x_i | x_{\pi_i})$ $\forall i$

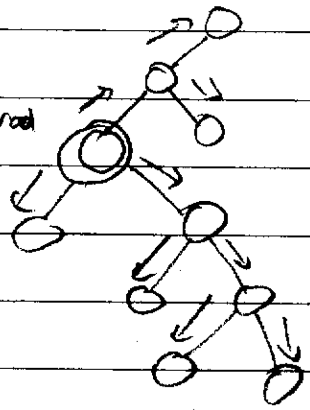
$$p(x) = \prod p(x_i | x_{\pi_i}) = \prod_c \psi_c(x_c)$$
$$= \frac{\prod_c \psi_c(x_c)}{\prod_S \psi_S(x_S)} \leftarrow \text{dens} = 1$$

- update cliques & separators on a tree structure recursively

```
Collect Evidence (node)
    for each child of node
        { update (node, collect Evidence (child)) };
    return (node);

Distribute Evidence (node)
    for each child of node
        { update (child, node);
          distribute evidence (child)    };
```



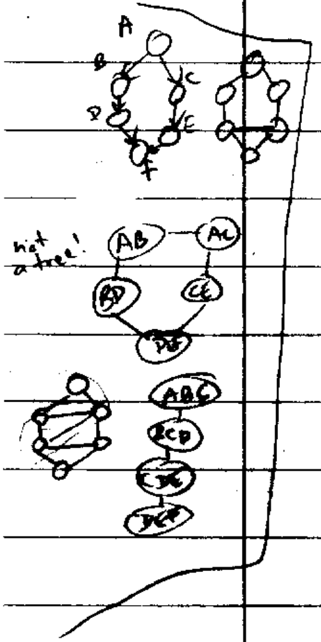


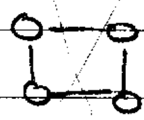
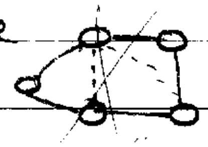
TRIANGULATION To get a valid Junction Tree → triangulate

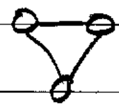
Undir. Node Elimination Algo → Triangulates Graph.

Triangulation (like Moralization) adds links → more complex graph, fewer independencies

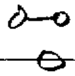
Such that: No 4 or more node cycle within graph where some nodes do not ⌊

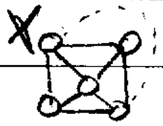


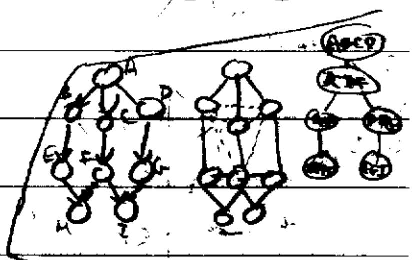
i.e. a 4-cycle  5-cycle 

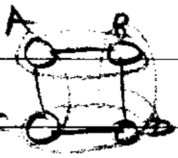
3-cycle  ok ✓

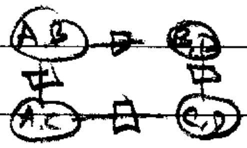
2-cycle ✓

1-cycle 

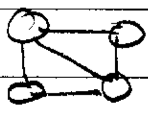
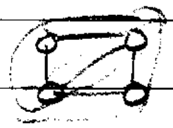
X 

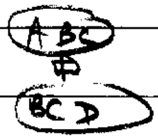


example: 

= 

← not a junction tree

triangulate:  or 



- optimal triangulation (adds fewest links) is NP
- but any triangulation (can be bad (heuristic) is P
- want to keep max clique size small
- also need resulting clique graph to be junction tree
  a) tree (no loops)
  b) satisfy junction tree property: cliques between path of ψ_A & ψ_B contain $A \cap B$ nodes

- When junction tree algo completes (induction proof)

$$\Psi_C = p(x_C, \bar{x}_E)$$
$$\Phi_S = p(x_S, \bar{x}_E)$$

← all cliques are marginals.

- Hugin (Junction Tree Algorithm) → Jensen's book

online → offline (once for a given graph):
  - Moralize (poly complex)
  - Introduce evidence $x_E \to \bar{x}_E$ (poly)
  - Triangulate (poly or NP for optimal)
  - Construct Junction Tree + (separators) (poly, Kruskal)

online (for different evidence queries etc.):
  - Propagate Probabilities (poly in # of cliques, expo in clique size)

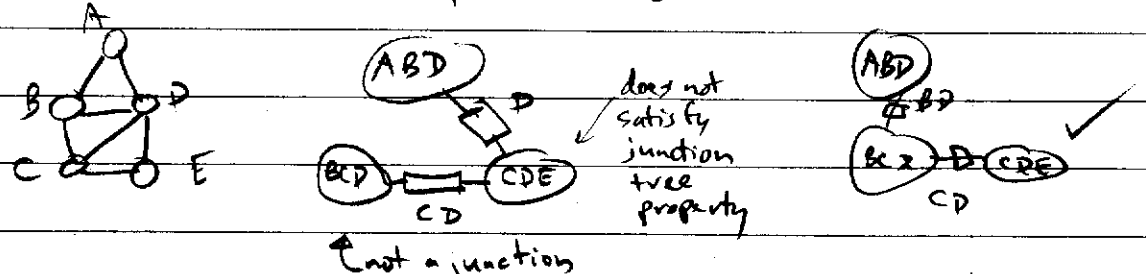
$$\Phi_S^* = \sum_{V \setminus S} \Psi_V$$
$$\Psi_W^* = \frac{\Phi_S^*}{\Phi_S} \Psi_W$$

- Without Proof: Triangulated Graph ⟺ decomposable

Triangulated Graph ↔ junction tree ↔ decomposable

junction tree ↕ maximum weight spanning tree

largest cardinality of separator nodes → maximum weight spanning tree

- 

not a junction tree

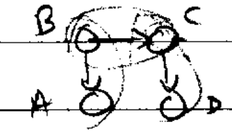
|separators| = |D| + |CD| = 3

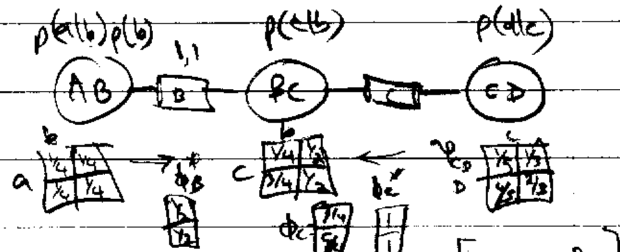
|separators| = |BD| + |CD| = 4

- Algo to find junction tree: max card. of separators → Kruskal

- Kruskal: Max Span Tree:
  1) Begin with no edges between cliques
  2) Add edge with max separator card. & make sure we don't make a loop
  3) Repeat 2 until graph connected (all cliques are connected via a path)

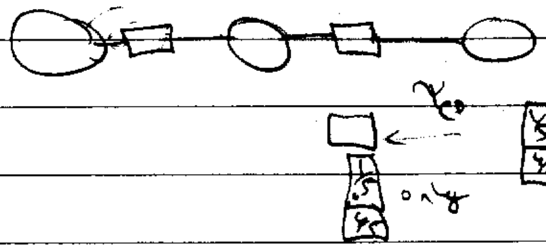
- Example:





$$\psi_{BC} = \begin{bmatrix} \frac{1}{4}/\frac{3}{4} & \frac{1}{2}/\frac{3}{4} \\ \frac{3}{4}/\frac{5}{4} & \frac{1}{2}/\frac{3}{4} \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \\ \frac{3}{5} & \frac{2}{3} \end{bmatrix}$$

with evidence
i.e. $X_{C_i} = 0$



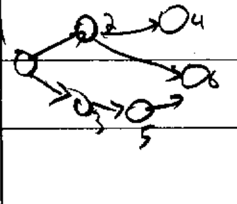
- Can compute fast marginals
- Can also compute fast argmax.

$$p(X_H, \bar{X}_E) = p(x_1, \ldots, x_n, \bar{x}_{n+1}, \ldots, \bar{x}_N)$$

← lock all $\bar{X}_E$, find X_H that maximizes $p()$

i.e. what is most likely state of patient if he has fever & headache?

$$= \arg\max_x p(x) = \max_1 p(x_1) \max_2 p(x_2|x_1) \max_3 p(x_3|x_1) \max_4 p(x_4|x_2) \max_5 p(x_5|x_3) \max_6 p(x_6|x_2, x_5)$$



fly
fever
sinus inf.
~~temperature~~
sinus swell
headache