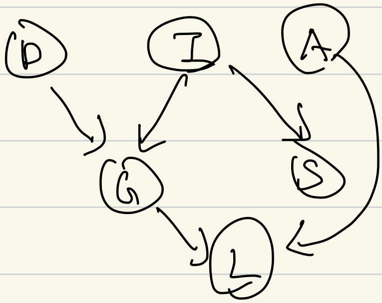
2/13/25 Factor graph + Sum-product algorithm.



Converting to factor graph

What is factor graph?

- Bipartite graph w/ two types of nodes

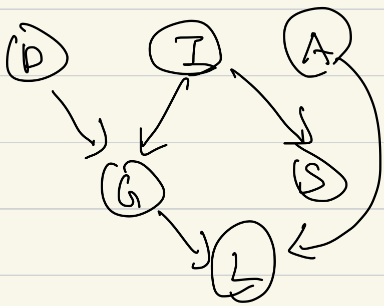
  - Variable nodes V and
  - Factor nodes F

Random variables : round nodes
Factor nodes : Square nodes

Given a DAG, $G=(V,E)$ convert it to a factor graph
$$G'=(V,F,E)$$

Every node in the original graph remains as a variable in G'

For each variable $v \in V$ create a factor $f_v = P(v \mid pa(v))$ ← conditional prob distr.
The factor connects to v and $pa(v)$.



For variable w/o parent factor f_v connects only to v.

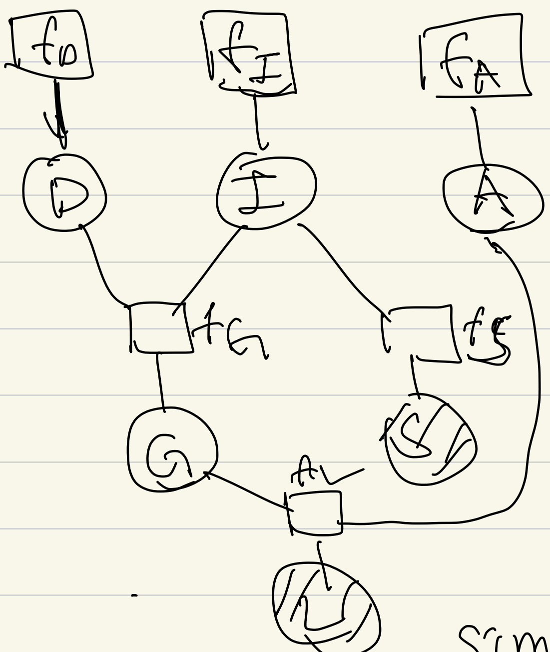
If the resulting factor graph has no cycle, we can perform exact inference

V→F message

$$m_{x \to f}(x) = \prod_{h \in n(x) \setminus \{f\}} m_{h \to x}(x)$$

$$m_{f \to x}(x) = \sum_{\sim x} f(x) \prod_{y \in n(f) \setminus \{x\}} m_{y \to f}(y)$$

$n(f), n(x)$: neighbors of f, x



pass 1

(1)

$$m_{f_D \to D}(d)$$

$$= \sum_{\sim d} f_D(d) \prod_{y \in n(f) \setminus \{d\}} (y)$$

$$n(f_D) \setminus \{D\} = \emptyset$$

$$= f_D(d) = P(d)$$

similarly

$$m_{f_I \to I}(i) = f_I(i) = P(i), \quad m_{f_A \to A}(a) = f_A(a) = P(a)$$

② $D \to f_G$, $I \to f_G$, $I \to f_S$

$$m_{D \to f_G}(d) = \prod_{h \in n(d) \setminus \{f_G\}} m_{h \to d}(d)$$

$n(d) = \{f_D, f_G\}$ so
$n(d) \setminus \{f_G\} = \{f_D\}$

$$m_{D \to f_G}(d) = m_{f_D \to d}(d) = p(d)$$

$$m_{I \to f_G}(i) = m_{f_G \to i}(i) = p(i)$$

$$m_{I \to f_S}(i) = p(i)$$

③ $f_G \to G$ $\quad n(f_G) \setminus \{G\} = \{D, I\}$

$$m_{f_G \to G}(g) = \sum_{-g} f_G(d, i, g) \cdot m_{D \to f_G}(d)\, m_{I \to f_I}(i)$$

$$= \sum_{d,i} P(g \mid d, i)\, p(d)\, p(i)$$

$$= p(g)$$

$f_S \to S \qquad n(f_S) = \{I, S\}$

$n(f_S) \setminus \{S\} = \{I\}$

$$m_{f_S \to S}(s) = \sum_{\sim s} f(s,i)\, m_{I \to f_S}(i)$$

$$= \sum_i p(s|i)\, p(i) = p(s)$$

4 $\quad G \to f_L \qquad n(G) = \{f_G, f_L\}$

$n(G) \setminus \{f_L\} = \{f_G\}$

$$m_{G \to f_L}(g) = m_{f_G \to G}(g) = p(g)$$

$A \to f_L \qquad n(A) \setminus \{f_L\} = \{f_A\}$

$$m_{A \to f_L}(a) = m_{f_A \to A}(a) = p(a)$$

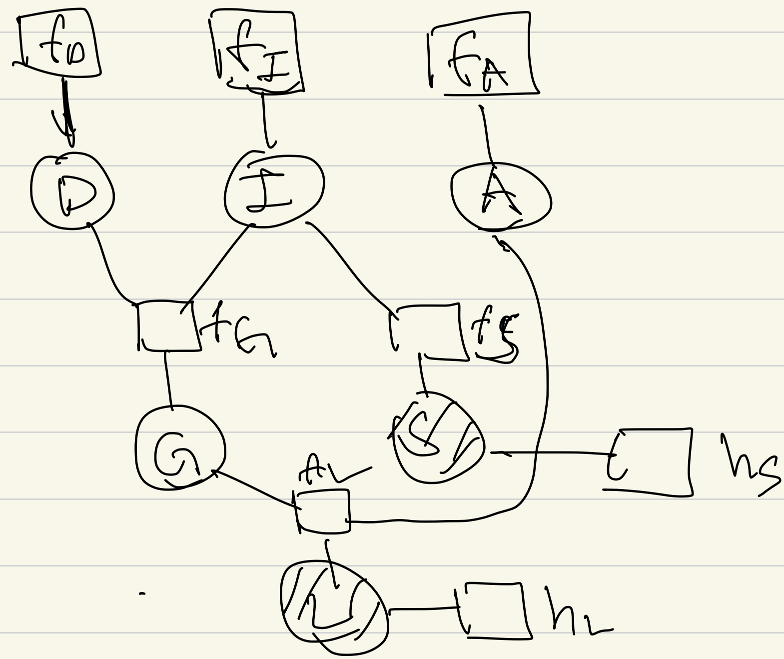
5 $\quad f_L \to L \qquad n(f_L) = \{G, A, L\}$

$$m_{f_L \to L}(l) = \sum_{\sim l} f(g,a,l)\, m_{G \to f_L}(g)\, m_{A \to f_L}(a)$$

$$= \sum_{g,a} p(l|g,a)\, p(a)\, p(g)$$

$$= p(l)$$

To handle observations, we will add a univariate factor for each observed variable



These factors will be indicator functions

$$h_L(l) = \mathbb{1}[l = l^*]$$
$$h_S(s) = \mathbb{1}[S = s^*]$$

let l^*, h^* denote observed values.

⑥ $L \to h_L$ $n(L) \setminus \{h_L\} = \{f_L\}$

$$m_{L \to h_L}(l) = m_{f_L \to L}(l)$$
$$= p(l)$$

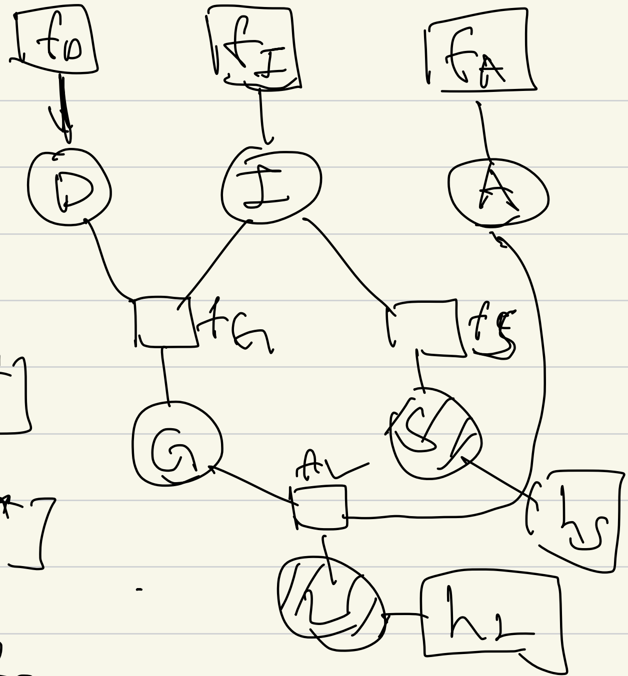
$S \to h_S$

$$m_{S \to h_S}(s) = p(s)$$

Essentially, nodes gather messages by taking the product

factors perform marginalization via summation.

Pass 2



① $m_{h_L \to L}(\ell) = h_L(\ell) = \mathbb{1}[\ell = \ell^*]$

$m_{h_S \to S}(s) = \mathbb{1}[s = s^*]$

② $L \to f_L$, $S \to f_S$

$m_{L \to f_L}(\ell) = h_L(\ell) = \mathbb{1}[\ell = \ell^*]$

$m_{S \to f_S}(s) = h_S(s) = \mathbb{1}[s = s^*]$

③ $f_L \to G$ $\quad n(f_L) \setminus \{G\} = \{L, A\}$

$$m_{f_L \to G}(g) = \sum_{-g} f_L(g, a, \ell)\, m_{L \to f_L}(\ell)\, m_{A \to f_L}(a)$$

$$= \sum_{a,\ell} p(\ell \mid g, a)\, \underline{\mathbb{1}[\ell = \ell^*]}\, p(a)$$

$$= p(\ell^* \mid g)$$

④ $G \to f_G$

$$m_{G \to f_G}(g) = m_{f_L \to G}(g) = p(\ell^* | g)$$

⑤ $f_G \to I$, $f_S \to I$

$$m_{f_G \to I}(i) = \sum_{d,g} f_G(i,d,g)\, m_{D \to f_G}(d)\, m_{G \to f_G}(g)$$

$$= \sum_{d,g} p(g|d,i)\, \underbrace{p(d)\, p(\ell^*|g)}_{}$$

$$p(g, d, \ell^* | i) = p(\ell^* | i)$$

$$m_{f_S \to I}(i) = \sum_{S} p(S|i)\, \mathbb{1}[S = s^*]$$

$$= p(s^* | i)$$

Compute $p(i | \ell^*, s^*)$

$$= p(\ell^*, s^*, i) / p(\ell^*, s^*)$$

Combine messages

$$m_{f_I \to I}(i) \times m_{f_G \to I}(i) \times m_{f_S \to I}(i)$$

$$= p(i) \cdot p(\ell^* | i)\, p(s^* | i) = \boxed{p(\ell^*, s^*, i)}$$

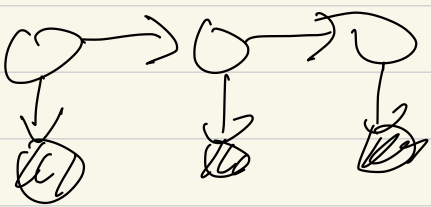
Belief propagation

first proposed by Judea Pearl
in the 80's for trees (poly trees)

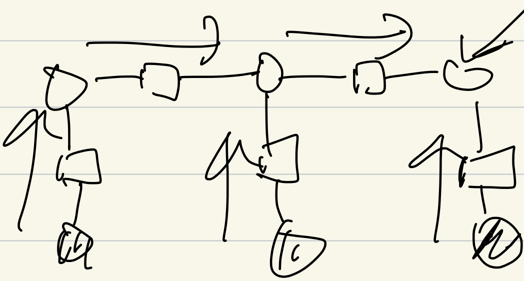
Sum-product algorithm
is exact if the right ordering is
used.

Otherwise, we get loopy belief
propagation.

HMM as message passing

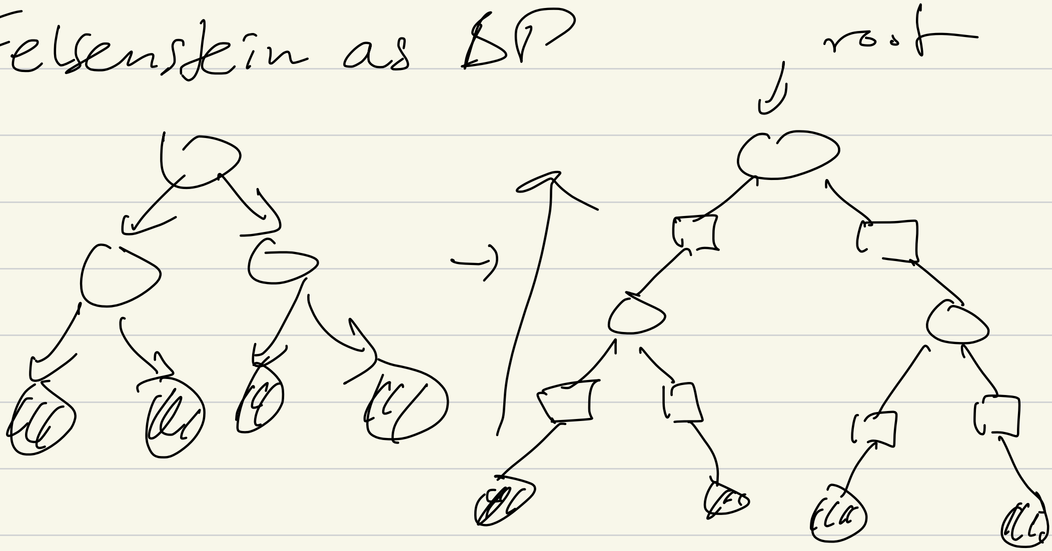


Choose a root



Messages from
obs (leaf)
to root

Felsenstein as DP



Leaf to root.