(12) 491 - L8 Inf GM2 (Lecture #2)

The sum-product algorithm

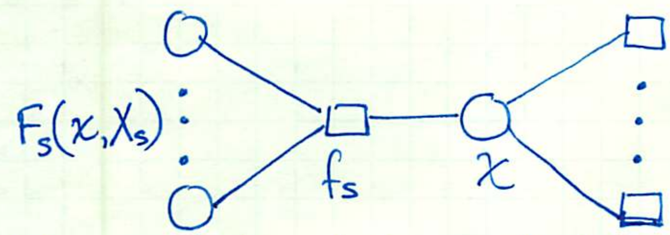
- special case of sum-product alg. is belief propagation.
- deal with UG & DG in same framework, either can be converted to a factor graph.

Goal:
1) efficient, exact inference alg for computing marginals
2) share computations when computing several marginals

Again, start with: $p(\underline{x}) = \prod_s f_s(\underline{x}_s)$

$$p(x_i) = \sum_{\underline{x} \setminus x_i} p(\underline{x}) = \sum_{\underline{x} \setminus x_i} \prod_s f_s(x_s)$$

$\underline{x} \setminus x_i \rightarrow$ all x_j except $j = i$



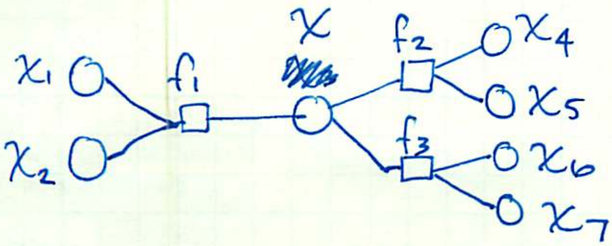
Can partition the factors of the joint distr. into groups:

$$p(\underline{x}) = \prod_{s \in ne(x)} F_s(x, X_s)$$

$ne(x)$ = set of factor nodes that are neighbors of x

X_s set of all vars in the subtree connected to the var node x via factor node f_s.

$F_s(x, X_s)$ product of all factors in the group associated with factor f_s.



$\cancel{p(\underline{x}) = \prod_s f_s(\underline{x}_s)}$

$$p(\underline{x}) = \prod_s f_s(\underline{x}_s)$$

$$= f_1(x_1, x_2, x)\, f_2(x, x_4, x_5)\, f_3(x, x_6, x_7)$$

$$p(\underline{x}) = \prod_{s \in ne(x)} F_s(x, X_s) =$$

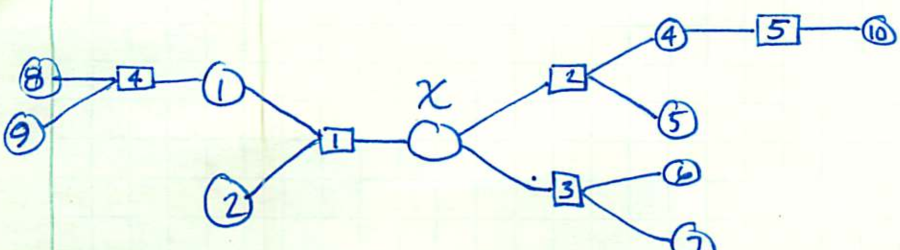
$s \in ne(x) = f_1, f_2, f_3$ = set of factor nodes that are neighbors of x.

X_s: $X_1 = \{x_1, x_2\}$, $X_2 = \{x_4, x_5\}$, $X_3 = \{x_6, x_7\}$ = set of all vars in the subtree connected to the variable node x via factor node f_s

$F_s(x, X_s)$ = product of all the factors in the group associated with factor f_s.

$$F(x, X_1) = f_1(x_1, x_2, x)$$
$$F(x, X_2) = f_2(x, x_4, x_5)$$
$$F(x, X_3) = f_3(x, x_6, x_7)$$

14 491 - L8 Inf GM2



$$p(\underline{x}) = \prod_s f_s(\underline{x}_s)$$

$$= f_4(x_1, x_8, x_9)\, f_1(x, x_1, x_2)\, f_2(x, x_4, x_5)\, f_3(x, x_6, x_7)\, f_{\cancel{3}\,5}(x_{\cancel{3}\,4}, x_{10})$$

$$p(\underline{x}) = \prod_{s \in ne(x)} F_s(x, X_s)$$

$s \in ne(x) = \{f_1, f_2, f_3\}$ = neighboring factor nodes

X_s: $X_1 = \{x_1, x_2, x_8, x_9\}$ = set of all vars in the subtree connected to the variable node x via factor node f_s.

$X_2 = \{x_4, x_5, x_{10}\}$

$X_3 = \{x_6, x_7\}$

$F_s(x, X_s)$ = product of all the factors in the group associated with factor s.

$F_1(x, X_1) = f_1(x, x_1, x_2)\, f_4(x_1, x_8, x_9)$

$F_2(x, X_2) = f_2(x, x_4, x_5)\, f_5(x_4, x_{10})$

$F_3(x, X_3) = f_3(x, x_6, x_7)$

15 491-L8 Inf GM2

We have:

$$P(\underline{x}) = \prod_{s \in ne(x)} F_s(x, X_s)$$

and $$p(x) = \sum_{\underline{x} \setminus x} P(\underline{x})$$

$$= \sum_{\underline{x} \setminus x} \prod_{s \in ne(x)} F_s(x, X_s)$$

Or from our example:

$$\sum_{\underline{x} \setminus x} F_1(x, X_1)\, F_2(x, X_2)\, F(x, X_3)$$

↓

$$\sum_{\cancel{X_1 \setminus x}\; X_1} \sum_{\cancel{X_2 \setminus x}\; X_2} \sum_{\cancel{X_3 \setminus x}\; X_3} F_1(x, X_1)\, F_2(x, X_2)\, F_3(x, X_3)$$

$=$

$$= \sum_{X_1} \sum_{X_2} F_1(x, X_1)\, F_2(x, X_2) \sum_{X_3} F_3(x_3, X_3)$$

$$= \sum_{X_1} F_1(x, X_1) \sum_{X_2} F_2(x, X_2) \sum_{X_3} F_3(x_3, X_3)$$

generalizing: $$= \prod_{s \in ne(x)} \left[\sum_{X_s} F_s(x, X_s) \right] \equiv \prod_{s \in ne(x)} \mu_{f_s \to x}(x)$$

Can you just "interchange sum and products"?

$$\sum_x \prod_s f_s(x, X_s)$$

$$\sum_i \prod_j a_{ij}$$

$$a_{11} a_{12} \cancel{a_{13}} + a_{21} a_{22} \cancel{a_{23}} + \cancel{a_{31}} a_{32} \cancel{a_{33}}$$

$$\prod_j \sum_i a_{ij} =$$

$$(a_{11} + a_{12}) \times (a_{21} + a_{22}) =$$

$$a_{11} a_{21} + a_{11} a_{21} + a_{1} a_{22} + a_{12} a_{22}$$

Interchanging sum and products: $p(x_i) = \sum_{\underline{x} \setminus x_i} p(\underline{x})$

$$= \sum_{\underline{x} \setminus x_i} \prod_{s \in ne(x_i)} F_s(x_i, X_s) = \prod_{s \in ne(x_i)} \left[\sum_{X_s} F_s(x_i, X_s) \right]$$

$$\sum_i \prod_j a_{ij} \neq \prod_j \sum_i a_{ij}$$

$$a_{11}a_{12} + a_{21}a_{22} \neq (a_{11} + a_{12})(a_{21} + a_{22})$$
$$= (a_{11}a_{21} + a_{12}a_{21})(a_{11}a_{22} + a_{12}a_{22})$$

What's going on?

X_s = set of all vars in sub tree connected to var node x_i via factor node f_s

$\bigcup_s X_s = \underline{x} \setminus x_i$

$$\sum_{\underline{x} \setminus x_i} \prod_{s \in ne(x_i)} F_s(x_i X_s)$$

$$\underbrace{\sum_{x_1} \cdots \sum_{x_{i-1}} \sum_{x_{i+1}} \cdots \sum_{x_N}}_{\text{these can be group into } X_s} \prod_{s \in ne(x_\xi)} F_s(x_\xi, X_s)$$

$\rightarrow F_s(x_\xi, X_s)$ only depends on X_s, so $\sum_{X_s}$ can be pushed in.

$$\sum_{X_1} \cdots \sum_{X_M} F_1(x_\xi, X_1) \cdots F_M(x_\xi, X_M)$$

$$\Rightarrow \sum_{X_1} F_1(x_\xi, X_1) \left[\sum_{X_2} F_2(x_\xi, X_2) \left[\cdots \sum_{X_M} F_M(x_\xi, X_M) \right] \cdots \right] \Big] $$

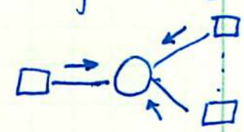
$$= \prod_{s \in ne(x_\xi)} \left[\sum_{X_s} F_s(x_\xi, X_s) \right]$$

(16) 491 - L8 wf GM2

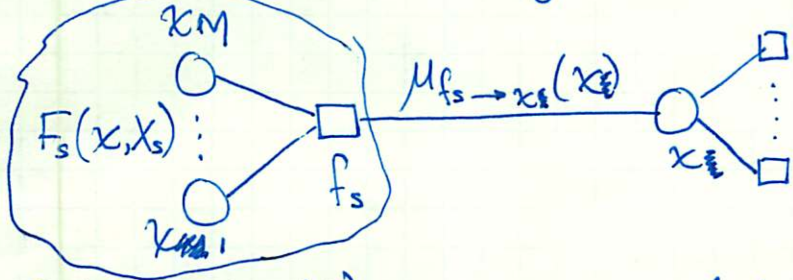
$$p(x) = \prod_{s \in ne(x)} \left[\sum_{X_s} F_s(x, X_s) \right]$$

$$= \prod_{s \in ne(x)} \mu_{f_s \to x}(x) \quad =$$ product of all incomming messages arriving at node x_i.

$$\mu_{f_s \to x}(x) \equiv \sum_{X_s} F_s(x, X_s)$$



can be viewed as the messages from factor nodes f_s to var node x_i.

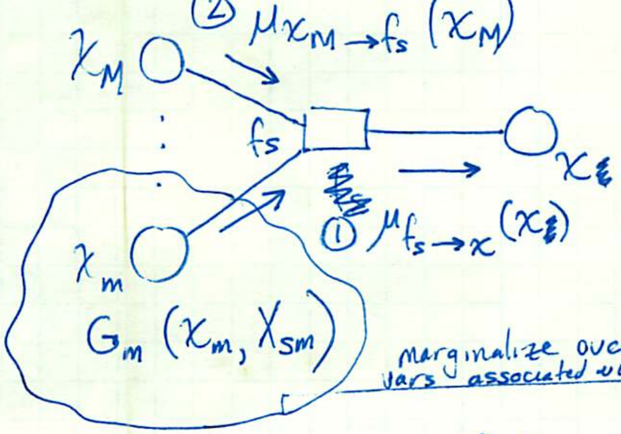


- Each factor $F_s(x, X_s)$ is described by a factor sub-graph

  $\Rightarrow$ it can also be factorized $\quad F_s(x, X_s) =$ product of all factors in group associated with factor f_s.

$$F_s(x, X_s) = f_s(\overbrace{x, x_1, \ldots, x_M}^{\text{vars assoc. w/ } f_s})\, G_1(x_1, X_{s1}) \cdots G_M(x_M, X_{sM})$$

$x + \{x_1 \cdots x_M\} =$ vars associated with factor f_s



$$\mu_{f_s \to x}(x) = \sum_{X_s} F_s(x, X_s)$$

$$\mu_{f_s \to x}(x) = \underbrace{\sum_{x_1} \cdots \sum_{x_M} f_s(x, x_{1:M})}_{\text{marginalize over all vars associated with msgs}} \underbrace{\prod_{m \in ne(f_s) \setminus x} \left[\sum_{X_{sm}} G_m(x_m, X_{sm}) \right]}_{\text{product over incomming msgs}}$$

Two kinds of messages.

$\Rightarrow$ ① $\mu_{f_s \to x}(x)$ & ② $\mu_{x_m \to f_s}(x_m)$ $\quad = \mu_{x_m \to f_s}(x_m)$

factor nodes to var nodes & var nodes to factor nodes $\quad$ Note: fns of x & x_m

(17) 491 - L8 Inf GM2

$$\cancel{\mu_{f_s \to x}(x) \equiv \sum_{X_s} F_s(x, X_s)}$$

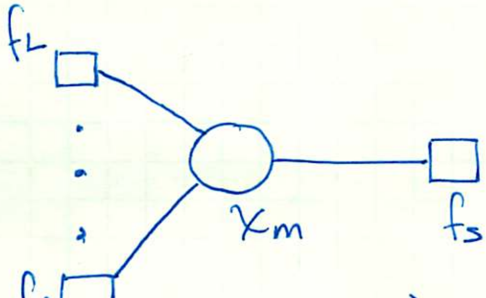
~~Each factor $F_s(x, X_s)$ is described by a factor (sub)graph, so it can be factorized:~~

$$\cancel{F_s(x, X_s) = f_s(x, x_{1:M})\, G_1(x_1, X_{s_1}) \cdots G_M(x_M, X_{s_M})}$$

$$\mu_{f_s \to x}(x) = \sum_{X_s} F_s(x, X_s)$$

$$\mu_{x_m \to f_s}(x_m) = \sum_{X_{s_m}} G_m(x_m, X_{s_m})$$

$$G_m(x_m, X_{s_m}) = \prod_{l \in ne(x_m) \setminus f_s} F_l(x_m, X_{m_l})$$



$F_l(x_m, X_{m_l})$

if only one f_l (two neighbors) x_m passes msg $\mu_{f_l \to x_m}(x_m)$ unchanged, i.e. it's not multiplied with other msgs.

Now we have

$$\mu_{x_m \to f_s}(x_m) = \sum_{X_{s_m}} \prod_{l \in ne(x_m) \setminus f_s} F_l(x_m, X_{m_l})$$

As before we get

$$= \prod_{l \in ne(x_m) \setminus f_s} \underbrace{\left[\sum_{X_{m_l}} F_l(x_m, X_{m_l}) \right]}_{= \mu_{f_l \to x_m}(x_m)}$$

Again we get a recursive soln like for chain.

→ out msg is product of incomming msgs

How do we do this for the whole graph?

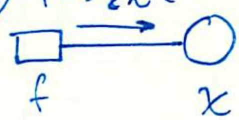
We have recursive definitions ⇒ need base cases.

How do we hande leaf nodes?

$\mu_{x \to f}(x) = 1$ leaf is a var node

because x has only f as a neighbor.

$\mu_{f \to x}(x) = f(x)$ leaf is a factor



$$\mu_{f_s \to x}(x) = \underbrace{\sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots x_M)}_{\text{there are none}} \underbrace{\prod_{m \in ne(f_s) \setminus x} \mu_{x_m \to f_s}(x_m)}_{\text{none}}$$

equiv to node always $= 1$

So we just ha $f_s(x) \cdot 1$

Like before we can efficiently compute all marginals by propagating msgs through whole graph, and the calculate each marginal from the incomming msgs

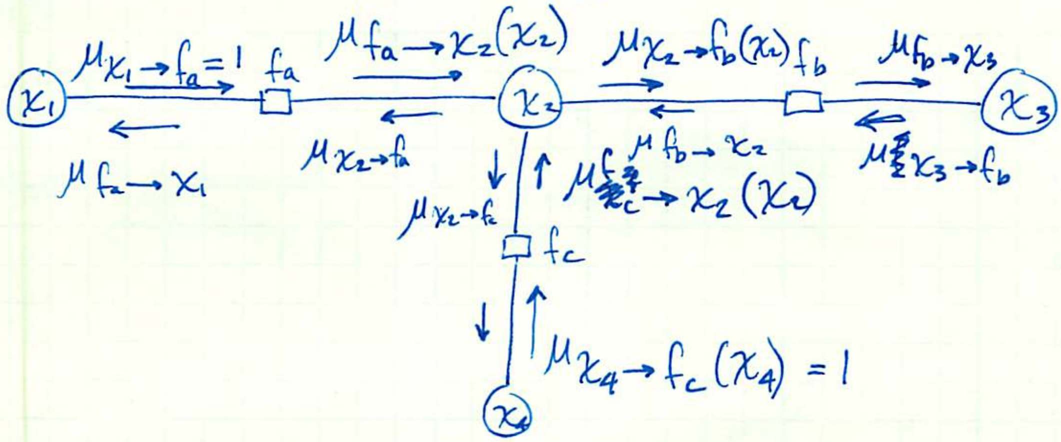
$$p(x_i) = \prod_{s \in ne(x_i)} \mu_{f_s \to x}(x)$$

$\#$ msgs $= 2 * \#$ links $= O(N)$

For a set: $p(\underline{x}_s) = f_s(\underline{x}_s) \prod_{i \in ne(f_s)} \mu_{x_i \to f_s}(x_i)$

What about Z? Easy to compute by computing Z for any one of $p(x_i)$.

Sum-product algorithm summary/illustration:



$$p(\underline{x}) = \frac{1}{Z} f_a(x_1, x_2)\, f_b(x_2, x_3)\, f_c(x_2, x_4)$$

Arbitrarily say x_3 is root.

Compute msgs:

~~all~~ leaf nodes are $= 1$.

$$\mu_{f_a \to x_2}(x_2) = \sum_{x_1} f_a(x_1, x_2) \cdot \underbrace{\prod_{m \in ne(f_a) \setminus x_2} \mu_{x_m \to f_a}(x_m)}_{=1}$$

$$\mu_{f_c \to x_2}(x_2) = \sum_{x_4} f_c(x_2, x_4)$$

$$\mu_{x_2 \to f_b}(x_2) = \prod_{l \in ne(x_2) \setminus f_b} \mu_{f_l \to x_2}(x_2) = \mu_{f_a \to x_2}(x_2)\, \mu_{f_c \to x_2}(x_2)$$

$$\mu_{f_b \to x_3}(x_3) = \sum_{x_2} f_b(x_2, x_3)\, \mu_{x_2 \to f_b}(x_2)$$

Now we can go the other way to get the complete set of msgs

$$\mu_{x_3 \to f_b} = 1$$

$$\mu_{f_b \to x_2}(x_2) = \sum_{x_3} f_b(x_2, x_3) \cdot 1$$

$$\mu_{x_2 \to f_a}(x_2) = \mu_{f_b \to x_2}(x_2)\,\mu_{f_c \to x_2}(x_2)$$

$$\mu_{f_a \to x_1}(x_1) = \sum_{x_2} f_a(x_1, x_2)\,\mu_{x_2 \to f_a}(x_2)$$

$$\mu_{x_2 \to f_c}(x_2) = \mu_{f_a \to x_2}(x_2)\,\mu_{f_b \to x_2}(x_2)$$

$$\mu_{f_c \to x_4}(x_4) = \sum_{x_2} f_c(x_2, x_4)\,\mu_{x_2 \to f_c}(x_2)$$

Now we can compute $p(x_2) = \frac{1}{Z} \prod_{s \in ne(x_2)} \mu_{f_s \to x_2}(x_2)$

$$= \frac{1}{Z}\,\mu_{f_a \to x_2}(x_2)\,\mu_{f_b \to x_2}(x_2)\,\mu_{f_c \to x_2}(x_2)$$

$$= \left[\sum_{x_1} f_a(x_1, x_2)\right]\left[\sum_{x_3} f_b(x_2, x_3)\right]\left[\sum_{x_4} f_c(x_2, x_4)\right]$$

$$= \sum_{x_1, x_3, x_4} f_a(x_1, x_2)\, f_b(x_2, x_3)\, f_c(x_2, x_4) = \sum_{\underline{x} \setminus x_2} p(\underline{x})$$

What about observed vars?

General approach: partition ~~vars~~ vars into hidden & visible $\underline{h}$ $\underline{v}$

~~Let~~ Let $\hat{\underline{v}}$ = observed values of $\underline{v}$

Then we can write

$$p(\underline{x}) \prod_i I(v_i, \hat{v}_i) \qquad I(v, \hat{v}) = 1$$

& Want $p(h_i \mid \underline{v} = \hat{\underline{v}}) = \dfrac{p(h_i, \underline{v} = \hat{\underline{v}})}{p(\underline{v} = \hat{\underline{v}})}$

$p(h_i, \underline{v} = \hat{v})$ is an unnormalize version of $p(h_i \mid \underline{v} = \hat{\underline{v}})$

So we can compute $p(h_i, \underline{v} = v)$ by clamping all the visible vars to their observed values, rather than summing over all possible values. ⇒ The sums collapse to a single term

The we just normalize locally like before