Graphical Models

Reference for some of today's material

Chapter 8 of Bishop, available on-line

http://research.microsoft.com/~cmbishop/PRML

and Kollar and Friedman, Chapter 3.

Note that Bishop uses $\perp\!\!\!\perp$ instead of $\perp$.

Factor Graph Review (1)

$$p(\mathbf{x}) = \prod_s f(x_s)$$ where x_s are sets of of variables within $\mathbf{x}$.

Denote variables by circles

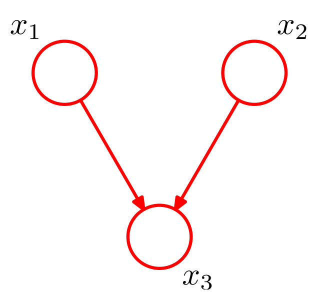
Denote each factor by a square

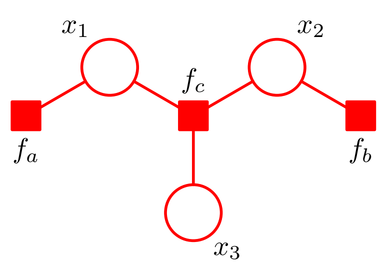
Draw links between squares and variables in the sets x_s.

Factor graphs are bipartite

Factor graph for a distribution is not necessarily unique.

Factor Graph Review (2)





$$p(\mathbf{x}) = \underbrace{p(x_1)}_{f_a}\underbrace{p(x_2)}_{f_b}\underbrace{p(x_3|x_1,x_2)}_{f_c}$$

Factor Graph Review (3)

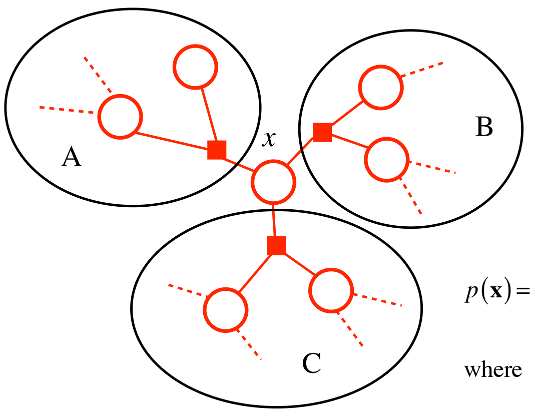
Factor graphs for directed trees, undirected trees, and directed polytrees are trees.

(Recall definition for undirected trees---there is only one path between any two nodes).

This means that (variable) node, x_n, with K branches divides a tree into K subtrees whose factors do not share variables except x_n.

Example, N=3

x connects subgraphs with node sets A, B, C.



$$p(\mathbf{x}) = F(x, X_A)\,F(x, X_B)\,F(x, X_C)$$

where $F(x, X_A) = \prod_s f(X_s)$

and where $X_s \subseteq \{x\} \cup A$

(similarly for B and C)

Sum-product algorithm

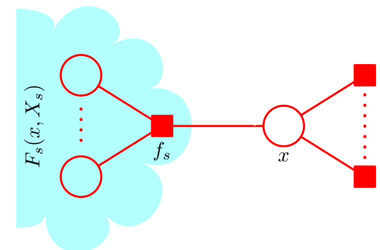
Basic computation --- compute marginal for a node, x_n.

The node x with N neighbors divides the graph into N subgraphs.

Define $F(x, X_s)$ as the product of all factors involving x and nodes in the subgraph, X_s.

$$p(\mathbf{x}) = \prod_{s \in n(x)} F(x, X_s)$$

(joint distribution)



Factor --> node messages

$$p(x) = \sum_{\mathbf{x}/x} \prod_{s \in n(x)} F(x, X_s)$$ (now marginalize)

$$= \prod_{s \in n(x)} \left\{ \sum_{X_s} F(x, X_s) \right\}$$ (interchange sums and products)

(recall our fancy formula)

$$\left(\sum a_i\right)\left(\sum b_j\right) = \sum \sum a_i b_j$$

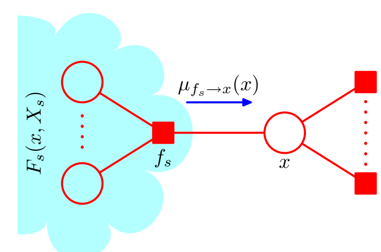
Note that each sum is simpler than what we started with because the variable sets are disjoint except for x.

Factor --> node messages

$$p(x) = \sum_{\mathbf{x}/x} \prod_{s \in n(x)} F(x, X_s)$$

$$= \prod_{s \in n(x)} \left\{ \sum_{X_s} F(x, X_s) \right\}$$

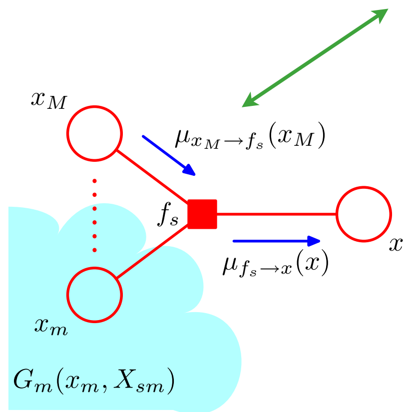
$$= \prod_{s \in n(x)} \mu_{f_x \to x}(x)$$



$$\mu_{f_x \to x}(x) \equiv \sum_{X_s} F(x, X_s)$$

Computing the factor-->node message

$$\underbrace{\mu_{f_x \to x}(x)}_{\text{factor}\to\text{node}} = \sum_{x_1} \cdots \sum_{x_M} f(x, x_1, \ldots, x_M) \prod_{m \in n(f_s) \setminus x} \underbrace{\mu_{x_m \to f_s}(x_m)}_{\text{node}\to\text{factor}}$$

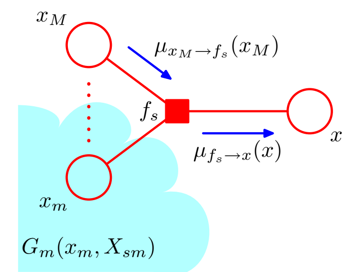


The node-->factor message

(define)

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

(for a node x_m we send its distribution with the other variables in the subgraph marginalized out).

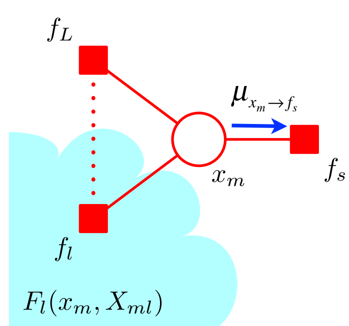


Computing the node-->factor message

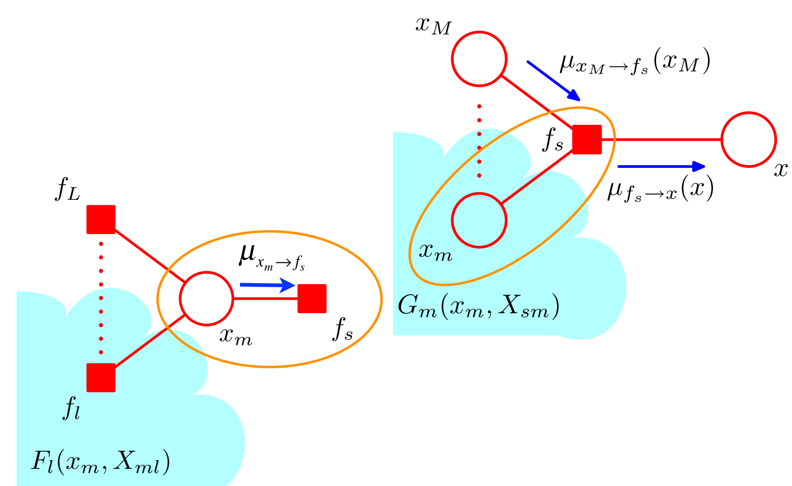
This is just essentially recursion (like base case except that we exclude the node we are sending to)

$$\mu_{x_m \to f_s}(x_m) = \prod_{l \in n(x_m) \setminus f_s} \mu_{f_l \to x_m}(x_m)$$

Note that nodes that only have two links just pass the message through (i.e., in the chain we skipped this part).



Review in pictures

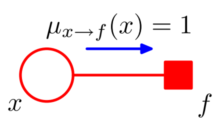


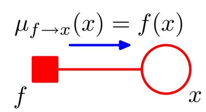
Finally, the sum-product algorithm

We could implement what we have just described as recursion, but the local view of nodes getting and passing messages suggests:

Pass messages from leaves to root. If you just want more than one marginal or plan to do other computation, store the results.

Initialization: If leaf node is a variable node, then start with a unity message. If leave node is factor, then start with the factor.





Finally, the sum-product algorithm

We could implement what we have just described as recursion, but the local view of nodes getting and passing messages suggests:

Pass messages from leaves to root. If you just want more than one marginal or plan to do other computation, store the results.

Initialization: If leaf node is a variable node, then start with a unity message. If leave node is factor, then start with the factor.

Note that all needed messages for computation will arrive at each node eventually.

The root node can compute the needed marginal.

The sum-product algorithm (2)

To prepare for other computations (e.g, all marginals), pass messages from the root to the leaves.

Now every node has incoming messages on all its links, and can thus be considered the root.

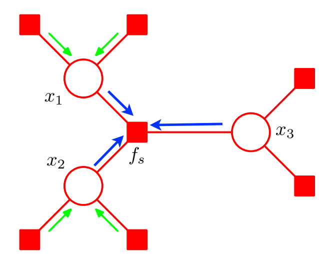
Hence we can compute all marginals for twice the cost of computing one of them.

The sum-product algorithm (3)

Another easy computation is the marginal for the group of variables in a factor.

Intuitively (and easily shown---homework) this is given by:

$$p(\mathbf{x}_s) = f(\mathbf{x}_s) \prod_{i \in n(f_s)} \mu_{x_i \to f_s}(x_i)$$



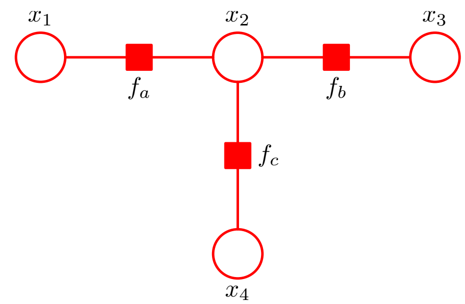
The sum-product algorithm (4)

If the factor graph came from a directed graph, then the expression for p(x) is already normalized.

Otherwise (as was the case of the chain), we can determine the normalization constant from one of the marginals (relatively inexpensive because only one variable is involved at that point).

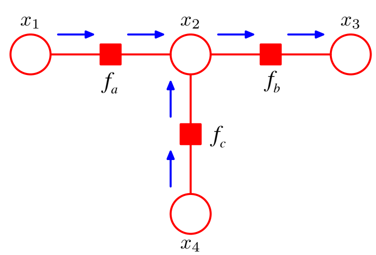
Sum-product algorithm example

Let $\tilde{p}(\mathbf{x}) = f_a(x_1, x_2) f_b(x_2, x_3) f_c(x_2, x_4)$

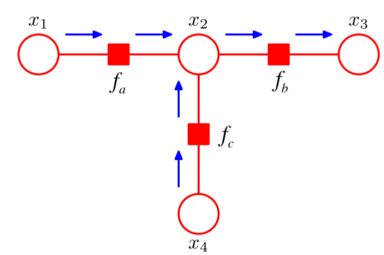


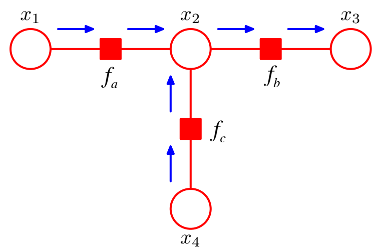
Declare x_3 as root node.

First we pass messages from leaves to root.

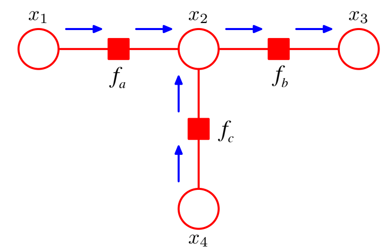


What are the messages we need to send?

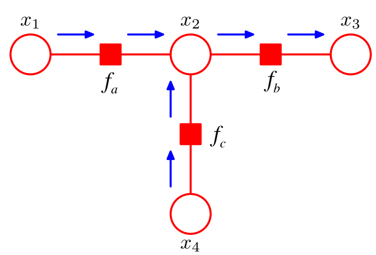




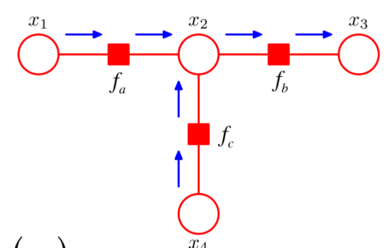
$$\mu_{x_1 \to f_a}(x_1) = 1$$
$$\mu_{f_a \to x_2}(x_2) = \sum_{x_1} f_a(x_1, x_2)$$



$$\mu_{x_1 \to f_a}(x_1) = 1$$
$$\mu_{f_a \to x_2}(x_2) = \sum_{x_1} f_a(x_1, x_2)$$
$$\mu_{x_4 \to f_c}(x_4) = 1$$
$$\mu_{f_c \to x_2}(x_2) = \sum_{x_4} f_c(x_2, x_4)$$

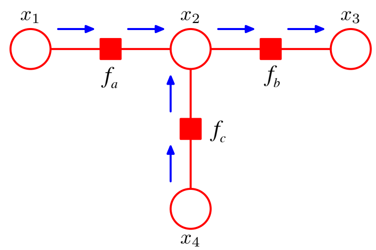


What is next?



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

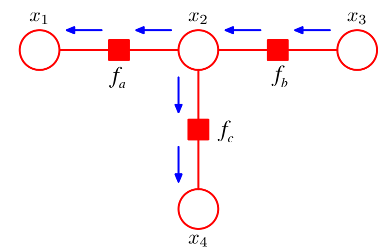
Summary of messages from leaves to root

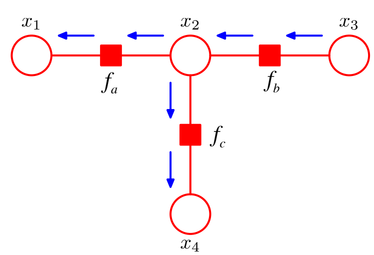


$$\mu_{x_1 \to f_a}(x_1) = 1$$
$$\mu_{f_a \to x_2}(x_2) = \sum_{x_1} f_a(x_1, x_2)$$
$$\mu_{x_4 \to f_c}(x_4) = 1$$
$$\mu_{f_c \to x_2}(x_2) = \sum_{x_4} f_c(x_2, x_4)$$
$$\mu_{x_2 \to f_b}(x_2) = \mu_{f_a \to x_2}(x_2)\,\mu_{f_c \to x_2}(x_2)$$
$$\mu_{f_b \to x_3}(x_3) = \sum_{x_2} f_b(x_2, x_3)\,\mu_{x_2 \to f_b}(x_2)$$

Next we pass messages from root to leaves.

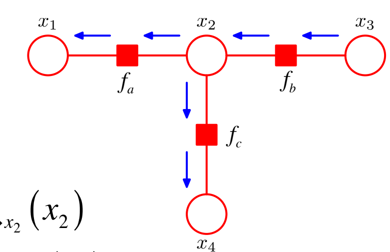
Candidate for the first and second ones?



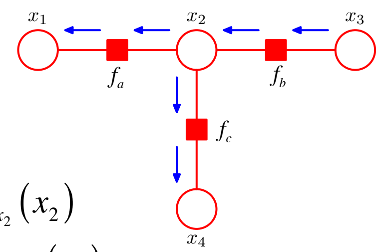


$$\mu_{x_3 \to f_b}(x_3) = 1$$
$$\mu_{f_b \to x_2}(x_2) = \sum_{x_3} f_b(x_2, x_3)$$

Candidate for third and fourth?

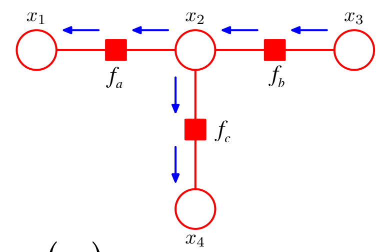


$$\mu_{x_2 \to f_a}(x_2) = \mu_{f_b \to x_2}(x_2)\,\mu_{f_c \to x_2}(x_2)$$
$$\mu_{f_a \to x_1}(x_1) = \sum_{x_2} f_a(x_1, x_2)\,\mu_{x_2 \to f_a}(x_2)$$

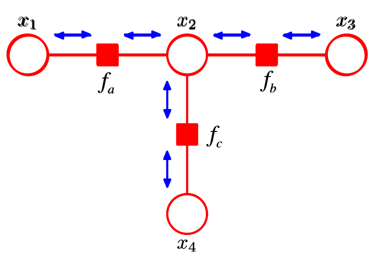


$$\mu_{x_2 \to f_c}(x_2) = \mu_{f_a \to x_2}(x_2)\,\mu_{f_b \to x_2}(x_2)$$
$$\mu_{f_c \to x_4}(x_4) = \sum_{x_2} f_c(x_2, x_4)\,\mu_{x_2 \to f_c}(x_2)$$

(similar to previous one)



$$\mu_{x_3 \to f_b}(x_3) = 1$$
$$\mu_{f_b \to x_2}(x_2) = \sum_{x_3} f_b(x_2, x_3)$$
$$\mu_{x_2 \to f_a}(x_2) = \mu_{f_b \to x_2}(x_2)\,\mu_{f_c \to x_2}(x_2)$$
$$\mu_{f_a \to x_1}(x_1) = \sum_{x_2} f_a(x_1, x_2)\,\mu_{x_2 \to f_a}(x_2)$$
$$\mu_{x_2 \to f_c}(x_2) = \mu_{f_a \to x_2}(x_2)\,\mu_{f_b \to x_2}(x_2)$$
$$\mu_{f_c \to x_4}(x_4) = \sum_{x_2} f_c(x_2, x_4)\,\mu_{x_2 \to f_c}(x_2)$$



An illustrative check

$$\begin{aligned}
\tilde{p}(x_2) &= \mu_{f_a \to x_2}(x_2)\,\mu_{f_b \to x_2}(x_2)\,\mu_{f_c \to x_2}(x_2) \\
&= \left(\sum_{x_1} f_a(x_1, x_2)\,\mu_{x_1 \to f_a}(x_1)\right)\left(\sum_{x_3} f_b(x_2, x_3)\,\mu_{x_3 \to f_b}(x_1)\right)\left(\sum_{x_4} f_c(x_2, x_4)\,\mu_{x_4 \to f_c}(x_1)\right) \\
&= \left(\sum_{x_1} f_a(x_1, x_2)\right)\left(\sum_{x_3} f_b(x_2, x_3)\right)\left(\sum_{x_4} f_c(x_2, x_4)\right) \\
&= \sum_{x_1}\sum_{x_3}\sum_{x_4} f_a(x_1, x_2)\,f_b(x_2, x_3)\,f_c(x_2, x_4) \\
&= \sum_{x_1}\sum_{x_3}\sum_{x_4} \tilde{p}(\mathbf{x})
\end{aligned}$$