Review

Factor Graphs

Suppose p(**x**) factorizes as:

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

Make a node for each x_i as usual.

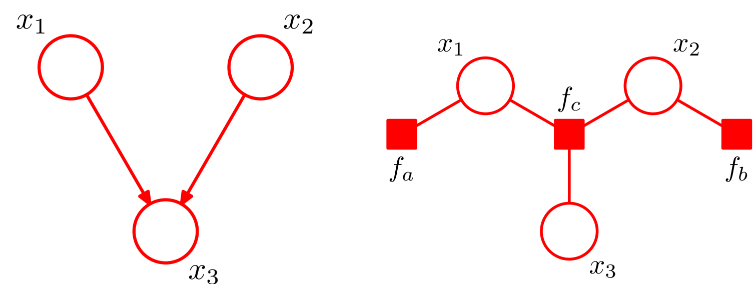
Now, make a different kind of node for *f()* (e.g., squares).

Draw edges between the factor nodes and the variables in the variable set, *s*.

Note that the factorization formula means that we can convert **both** directed and undirected graphs to factor graphs.

Review

Factor Graph Example (2)



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

Review

Factor Graph Summary

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

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.

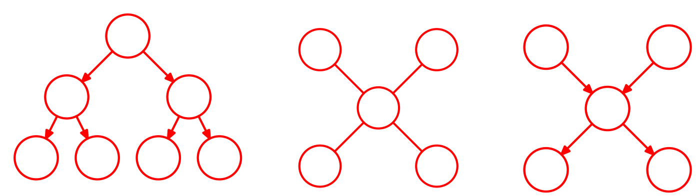
Review

Trees/Polytrees

A directed graph is tree if the root node has no parents, others have exactly one parent.

An undirected graph is a tree if there is only one path between any pair of nodes.

A directed graph is a polytree if there is only one path per pair of nodes.



Review

Factor Graphs and Trees

Factor graphs for directed trees, undirected trees, and directed polytrees can all be trees.

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

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

Review

Observations about factor graphs for trees

Any node can be root

Any node with K links splits the graph into K subgraphs which do not share nodes.

If we pass messages from:
1) the leaves to a chosen root;
2) the chosen root to the leaves,

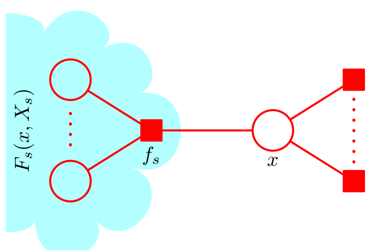
then **all** messages that **can** be passed **have** been passed.

Further, the number of messages in 1 and 2 are the same.

Review

Sum-product algorithm

We defined two kinds of messages:
1) From factors to nodes.
2) From nodes to factors.



Review

Marginal distribution for a node x

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

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

(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.

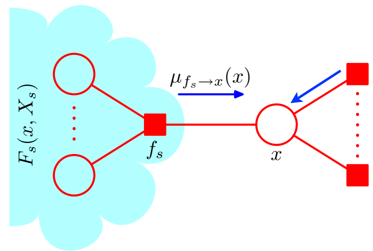
Review

Factor $\to$ node messages

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

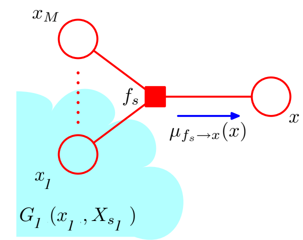
$$= \prod_{s \in n(x)} \left\{ \sum_{X_s} F(x, X_s) \right\}$$

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



$$\mu_{f_x \to x}(x) \equiv \sum_{X_s} F(x, X_s)$$ (factor-to-node message)

Computing the factor $\to$ node messages



$\mu_{f_s \to x}(x) \equiv \sum_{X_s} F_s(x, X_s)$ (sum removes all variables except x.)

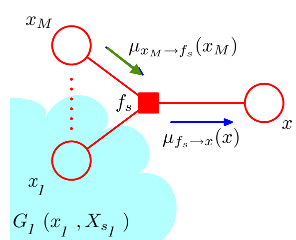
Where $F_s(x, X_s) = f_s(x, x_1, x_2, \ldots, x_M) G_1(x_1, X_{s1}) G_2(x_2, X_{s2}) \ldots G_M(x_M, X_{sM})$

$$\sum_{X_s} F_s(x, X_s) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_M} f_s(x, x_1, x_2, \ldots, x_M) \sum_{X_{s1}} \sum_{X_{s2}} \cdots \sum_{X_{sM}} G_1(x_1, X_{s1}) G_2(x_2, X_{s2}) \ldots.\ G_M(x_M, X_{sM})$$

So, $\sum_{X_s} F_s(x, X_s) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_M} f_s(x, x_1, x_2, \ldots, x_M) \prod_{m \in ne(f_s) \backslash x} \sum_{X_{sm}} G_m(x_m, X_{sm})$

(interchanging sums and products)

Computing the factor $\to$ node messages



$$\sum_{X_s} F_s(x, X_s) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_M} f_s(x, x_1, x_2, \ldots, x_M) \prod_{m \in ne(f_s) \backslash x} \sum_{X_{sm}} G_m(x_m, X_{sm})$$

$$= \sum_{x_1} \sum_{x_2} \cdots \sum_{x_M} f_s(x, x_1, x_2, \ldots, x_M) \prod_{m \in ne(f_s) \backslash x} \mu_{x_m \to f_s}(x_m)$$

where we define:

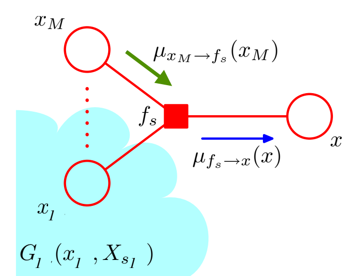
$$\mu_{x_m \to f_s}(x_m) \equiv \sum_{X_{sm}} G_m(x_m, X_{sm})$$ (node $\to$ factor messages)

The node $\to$ factor message

(We have defined)

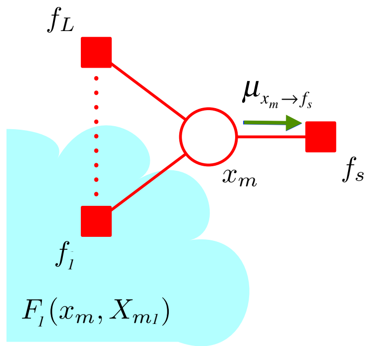
$$\mu_{x_m \to f_s}(x_m) \equiv \sum_{X_{s_m}} G_m(x_m, X_{s_m})$$

(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 like where we started, but we exclude the node we are sending to. Doing the drill one more time ...



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

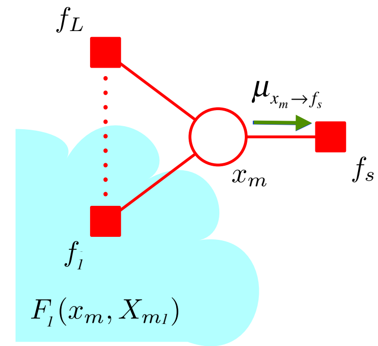
$$= \sum_{X_{s_m}} \prod_{l \in n(x_m) \setminus f_s} F_l(x_m, X_{ml}) \quad \text{(factors inside } G_m\text{)}$$

$$= \prod_{l \in n(x_m) \setminus f_s} \sum_{X_{ml}} F_l(x_m, X_{ml}) \quad \text{(exchanging sums and products)}$$

$$= \prod_{l \in n(x_m) \setminus f_s} \mu_{f_l \to x_m}(x_m) \quad \text{where } \mu_{f_l \to x_m}(x_m) = \sum_{X_{ml}} F_l(x_m, X_{ml})$$

Computing the node → factor message

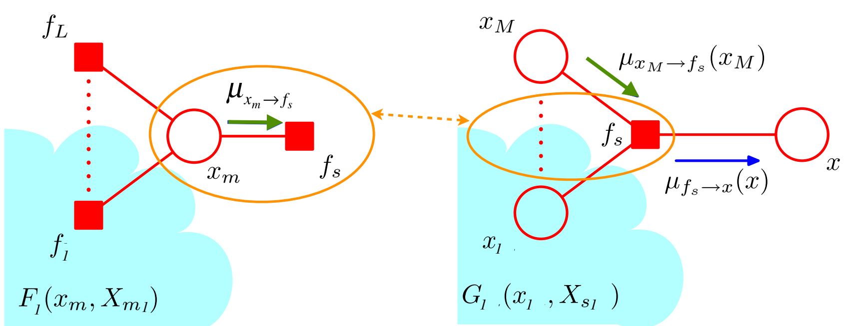
This is just like where we started, but we exclude the node we are sending to. Doing the drill one more time ...



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

All nodes attached to the factor except the one the message is going to. Nodes that only have two links just pass the message through (i.e., in the chain we skipped this part).

Sum product on a slide



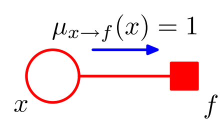
$$\mu_{x_m \to f_s}(x_m) = \prod_{l \in n(x_m) \setminus f_s} \mu_{f_l \to x_m}(x_m)$$

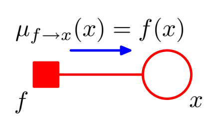
$$\mu_{f_s \to x}(x) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_M} f_s(x, x_1, x_2, \ldots, x_M) \prod_{m \in ne(f_s) \setminus x} \mu_{x_m \to f_s}(x_m)$$

The sum-product algorithm (1)

First, 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 leaf node is factor, then start with the factor.





The sum-product algorithm (2)

First, 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 leaf 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 (3)

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

Because we just passed all messages to the root, the root has all the messages from its neighbors, and thus all it needs to compute the outward messages

Once we pass the messages back to the leaves, every node has all possible incoming messages on all its links, and can thus be considered the root.

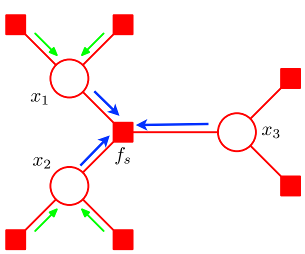
Hence we can compute all marginals for twice the cost of computing one of them (from before, all messages that can be passed, have now been passed).

The sum-product algorithm (4)

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

Intuitively (and easily shown) this is given by:

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



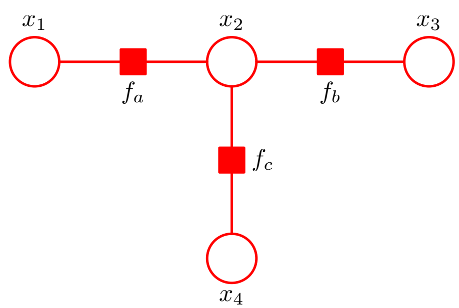
The sum-product algorithm (5)

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 by summing up one of the marginals (relatively inexpensive because only one variable is involved).

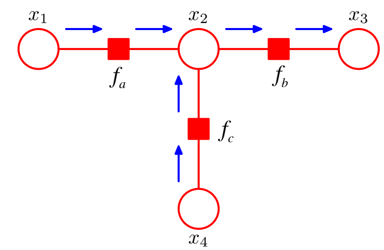
Sum-product algorithm example

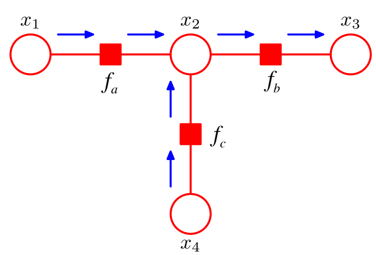
Let $\tilde{p}(\mathbf{x}) = f_a(x_1,x_2)\, f_b(x_2,x_3)\, f_c(x_2,x_4)$



Declare x_3 as root node.

First we pass messages from leaves to root.





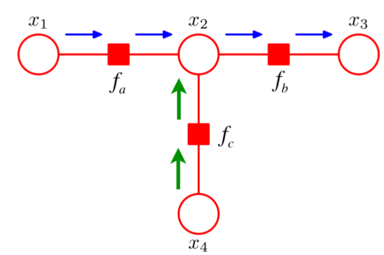
$$\mu_{x_1 \to f_a}(x_1) = 1$$

$$\mu_{f_a \to x_2}(x_2) = ?$$

Recall

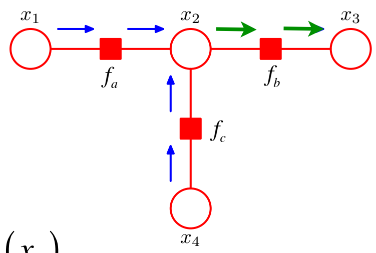
$$\underbrace{\mu_{f_x \to x}(x)}_{\text{factor}\to\text{node}} = \sum_{x_1} \cdots \sum_{x_M} f(x, x_1, \ldots, x_M) \prod_{m \in n(f_s)\backslash x} \underbrace{\mu_{x_m \to f_s}(x_m)}_{\text{node}\to\text{factor}}$$

(red colored x_i are not the same as the ones in example!)



$$\mu_{x_4 \to f_c}(x_4) = 1$$

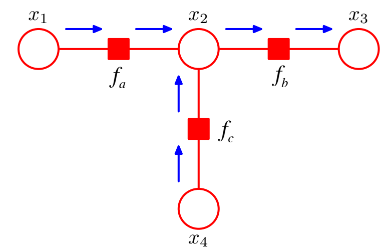
$$\mu_{f_c \to x_2}(x_2) = \sum_{x_4} f_c(x_2, x_4)$$



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

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

Summary of messages from leaves to root



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

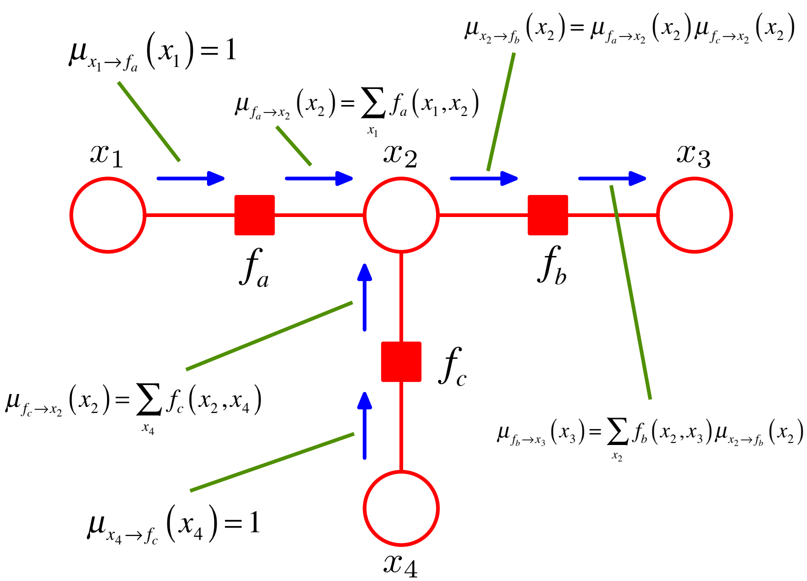
$$\mu_{f_a \to x_2}(x_2) = \sum_{x_1} f_a(x_1, x_2)$$

$$\mu_{x_4 \to f_c}(x_4) = 1$$

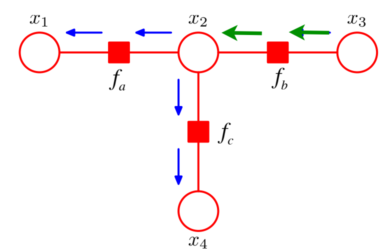
$$\mu_{f_c \to x_2}(x_2) = \sum_{x_4} f_c(x_2, x_4)$$

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

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



Next we pass messages from root to leaves.

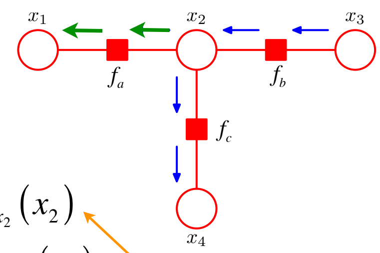


$$\mu_{x_3 \to f_b}(x_3) = 1$$

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

Candidate for third and fourth?

Lets go towards x_1 first.

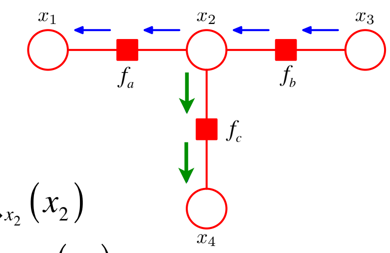


$$\mu_{x_2 \to f_a}(x_2) = \mu_{f_b \to x_2}(x_2)\,\mu_{f_c \to x_2}(x_2)$$

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

Note use of saved message from going the other way.

Also, note that arrows define which link to exclude, and who gets the message. Factors are included irrespective of the direction of the blue arrows.

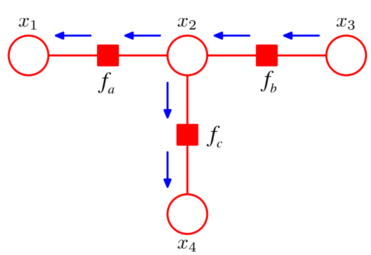


$$\mu_{x_2 \to f_c}(x_2) = \mu_{f_a \to x_2}(x_2)\,\mu_{f_b \to x_2}(x_2)$$

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

(similar to previous one)

Summary of messages from root to leaves.



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

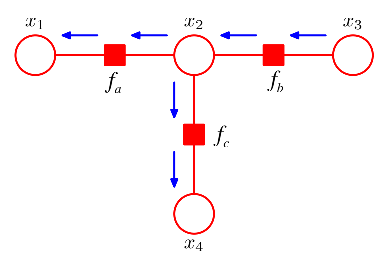
$$\mu_{f_b \to x_2}(x_2) = \sum_{x_3} f_b(x_2, x_3)$$

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

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

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

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



An illustrative check

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

Handling observed variables

Usually we have observed variables (e.g., evidence). We simply clamp those variables to their observed values.

More formally, denote hidden variables by $\mathbf{h}$, and observed ones by $\mathbf{v}$. Denote the observed value as $\hat{\mathbf{v}}$. For each observed variable, v_i, with value $\hat{v}_i$, we can introduce factors into the graph

$$I(v_i, \hat{v}_i) = \begin{cases} 1 & \text{if } v_i = \hat{v}_i \\ 0 & \text{otherwise} \end{cases}$$

Then, $p(\mathbf{h}, \mathbf{v} = \hat{\mathbf{v}}) = p(\mathbf{h}, \mathbf{v}) \prod_i I(v_i, \hat{v}_i)$

Adds factor nodes, i.e.,

(needs to be normalized to get $p(\mathbf{h}|\hat{\mathbf{v}})$, but this is easy since we are doing sum-product.)