Factor Graph Summary

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

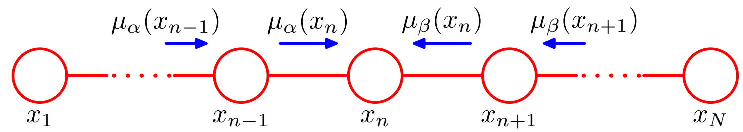
Denote variables by circles

Denote each factor by a square

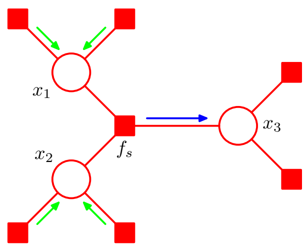
Draw links between squares and variables in the sets x_s.

Factor graphs are bipartite

Factor graph for a distribution is not necessarily unique.



Factor graphs conveniently represent the extended message passing needed for inference on trees/polytrees.

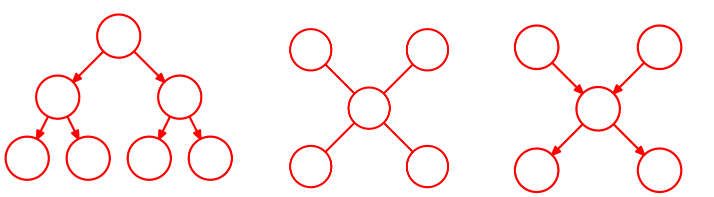


Trees/Polytrees

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.



Factor Graphs and Trees

The factor graphs for directed trees, undirected trees, and directed polytrees are all 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.

Observations about factor graphs for trees

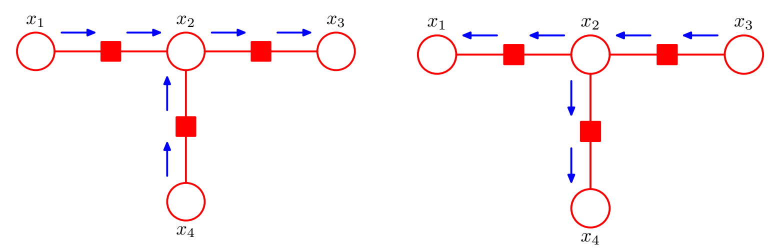
Any node can be root

Any node with N links splits the graph into N 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.

Observations about factor graphs



(x_3 is the root)

Sum-product algorithm

Generalizes what we did with chains.

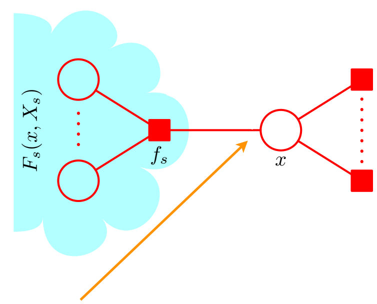
Generalizes and simplifies an algorithm introduced as "belief propagation".

As with chains, consider the problem of computing the marginal of a selected node, x_n.

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

Sum-product algorithm

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



In analogy with chains, factor-to-node messages provide marginal distributions for the node, for a subgraph. (In the chain case, we had the left side and the right side).

In the chain case we did not have factor nodes. This worked because then second kind of message is just "pass through" or "copy" in the case of only two links. So, we described it as simply passing messages from node to node.

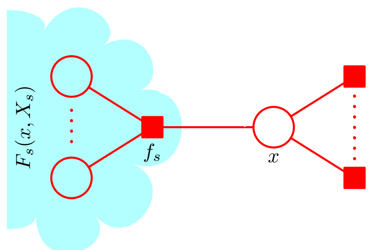
Marginal distribution for a node x

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.

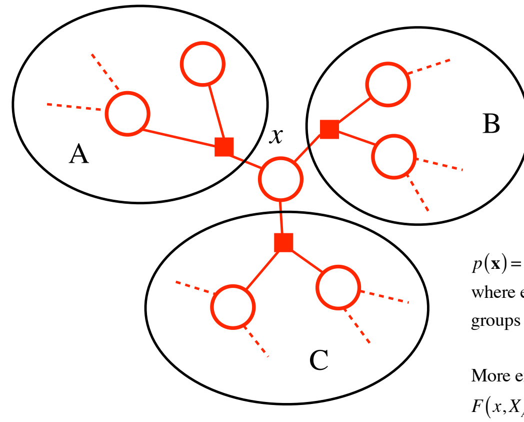
$$p(\mathbf{x}) = \prod_{s \in n(x)} F(x, X_s)$$

(joint distribution)



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 each of these three factors are themselves groups of factors over x and the subgraphs.

More explicitly,

$F(x, X_A) = \prod_s f(X_s)$ with $X_s \subseteq \{x\} \cup A$

$F(x, X_B) = \prod_s f(X_s)$ with $X_s \subseteq \{x\} \cup B$

$F(x, X_C) = \prod_s f(X_s)$ with $X_s \subseteq \{x\} \cup C$

Marginal distribution for a node x

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

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

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

(define)

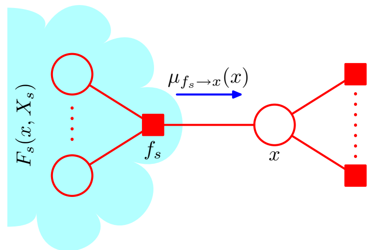
$$\mu_{f_x \rightarrow x}(x) \equiv \sum_{X_s} F(x, X_s)$$

Factor $\to$ node messages

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

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

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

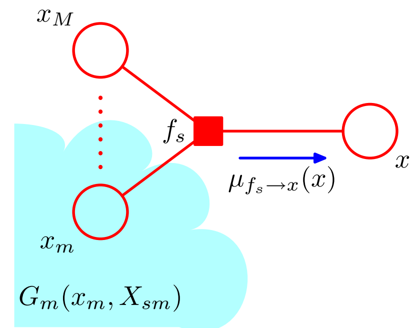


$$\mu_{f_s\to x}(x) \equiv \sum_{X_s} F(x,X_s)$$

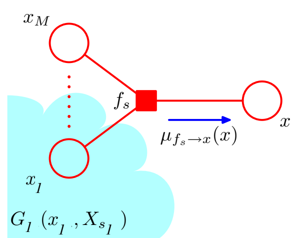
Computing the factor $\to$ node messages

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

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



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.)

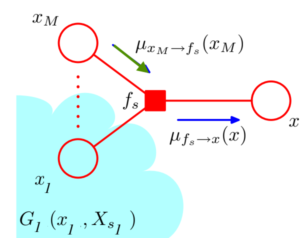
Where $F_s(x,X_s) = f_s(x,x_1,x_2,\ldots,x_M)G_1(x_1,X_{s1})G_2(x_2,X_{s2})\ldots G_M(x_M,X_{sM})$

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

(We get the above by moving products outside sums where possible)

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

Computing the factor $\to$ node messages



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

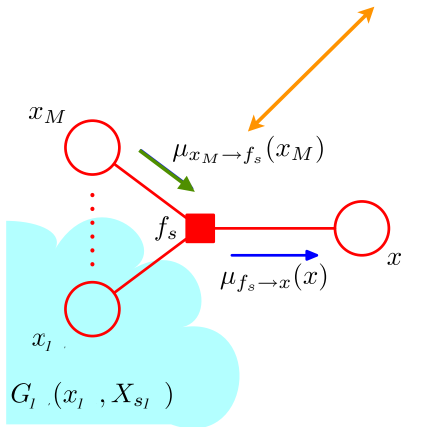
$$= \sum_{x_1}\sum_{x_2}\cdots\sum_{x_M} f_s(x,x_1,x_2,\ldots,x_M) \prod_{m\in ne(f_s)\backslash x} \mu_{x_m\to f_s}(x_m)$$

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)

Summary of computation for factor → node message

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

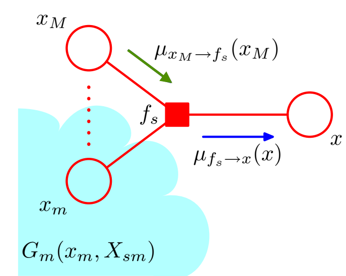


The node → factor message

(We have defined)

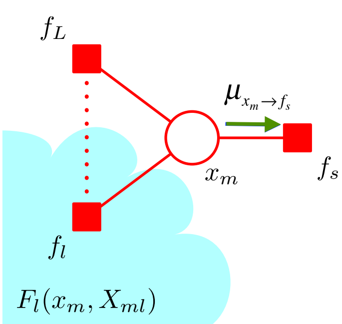
$$\mu_{x_m\to f_s}(x_m) \equiv \sum_{X_{sm}} G_m(x_m, X_{s_m})$$

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



Computing the node → factor message

$$\mu_{x_m\to f_s}(x_m) = ?$$

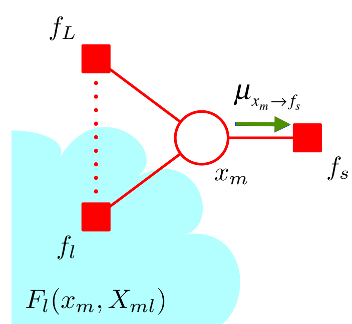


Computing the node → factor message

$$\mu_{x_m\to f_s}(x_m) = ?$$

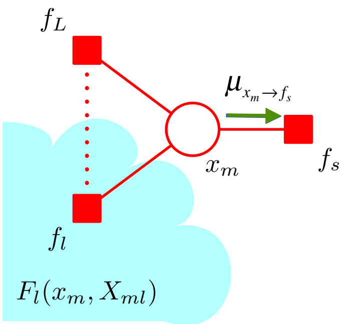
This is just like the first case, but we exclude the node we are sending to.

$$\left(\begin{array}{l}\text{Recall we started with} \\ p(x) = \prod_{s\in n(x)} \mu_{f_s\to x}(x)\end{array}\right)$$



Computing the node $\rightarrow$ factor message

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



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

Review in pictures