Sum-Product Algorithm

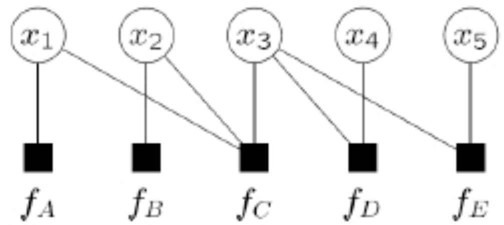
CSci 5512: Artificial Intelligence II

Factor Graphs

- Many problems deal with global function of many variables
- Global function “factors” into product of local functions
- Efficient algorithms take advantage of such factorization
- Factorization can be visualized as a factor graph

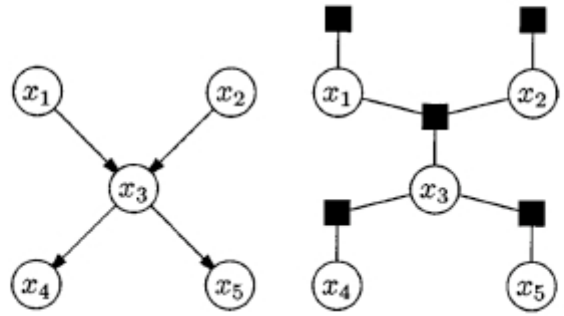
Example

$$g(x_1, x_2, x_3, x_4, x_5) = f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_D(x_3, x_4) f_E(x_3, x_5)$$



- Bipartite graph over variables and local functions
- Edge ≡ “is an argument of” relation
- Encodes an efficient algorithm

Bayes Nets to Factor Graphs



$f_A(x_1) = p(x_1)$ $f_B(x_2) = p(x_2)$ $f_C(x_1,x_2,x_3) = p(x_3|x_1,x_2)$

$f_D(x_3,x_4) = p(x_4|x_3)$ $f_E(x_3,x_5) = p(x_5|x_3)$

Marginalize Product of Functions

- Many problems involve "marginalize product of functions" (MPF)
- Inference in Bayesian networks
  - Compute $p(x_1|x_4,x_5)$
  - Need to compute $p(x_1,x_4,x_5)$ and $p(x_4,x_5)$
  - Marginalization of joint distribution is a MPF problem
- Several other problems use MPF
  - Prediction/Filtering in dynamic Bayes nets
  - Viterbi decoding in hidden Markov models
  - Error correcting codes

Marginalize Product of Functions (Contd.)

- The "not-sum" notation

$$\sum_{\sim x_2} h(x_1, x_2, x_3) = \sum_{x_1, x_3} h(x_1, x_2, x_3)$$

- Recall

$$g(x_1, x_2, x_3, x_4, x_5) = f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_C(x_1, x_2, x_3) f_D(x_3, x_4) f_E(x_3, x_5)$$

- Computing marginal function using not-sum notations

$$g_i(x_i) = \sum_{\sim x_i} g(x_1, x_2, x_3, x_4, x_5)$$

MPF using Distributive Law

We focus on two examples: $g_1(x_1)$ and $g_3(x_3)$
From distributive law

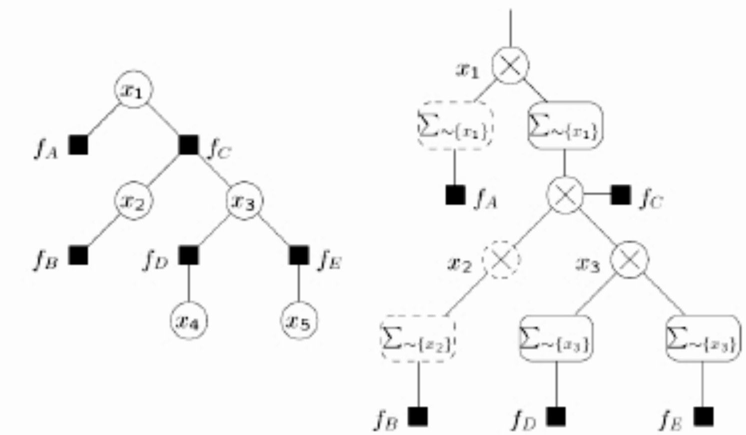
$$g_1(x_1) = f_A(x_1) \sum_{\sim x_1} f_B(x_2) f_C(x_1, x_2, x_3) f_C(x_1, x_2, x_3) \sum_{\sim x_3} f_D(x_3, x_4) \sum_{\sim x_3} f_E(x_3, x_5)$$

Also

$$g_3(x_3) = \sum_{\sim x_3} f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_C(x_1, x_2, x_3) \sum_{\sim x_3} f_D(x_3, x_4) \sum_{\sim x_3} f_E(x_3, x_5)$$

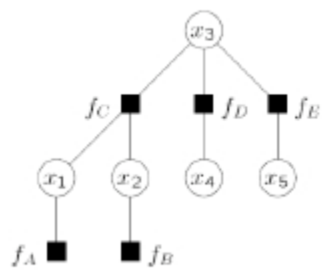
Message Passing Example: Computing $g_1(x_1)$

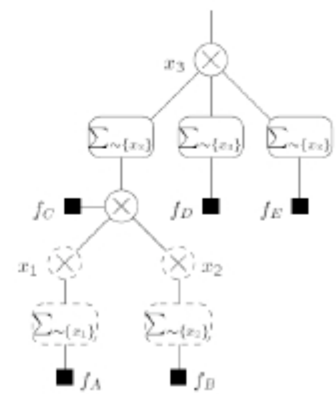
$$f_A(x_1) \times \sum_{\sim\{x_1\}} \left(f_B(x_2) \times f_C(x_1, x_2, x_3) \times \left(\sum_{\sim\{x_3\}} f_D(x_3, x_4) \right) \times \left(\sum_{\sim\{x_3\}} f_E(x_3, x_5) \right) \right)$$



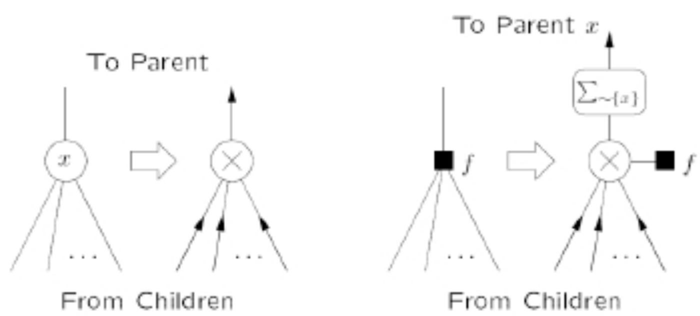
Message Passing Example: Computing g2(x2)

$$g_3(x_3) = \left(\sum_{\sim\{x_3\}} f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) \right) \times \left(\sum_{\sim\{x_3\}} f_D(x_3, x_4) \right) \times \left(\sum_{\sim\{x_3\}} f_E(x_3, x_5) \right)$$





Local Transformation for Message Passing



Sum-Product Algorithm

The overall strategy is simple message passing

- To compute $g_i(x_i)$, form a rooted tree at x_i
- Apply the following two rules:

Product Rule:

At a **variable** node, take the product of descendants

Sum-product Rule:

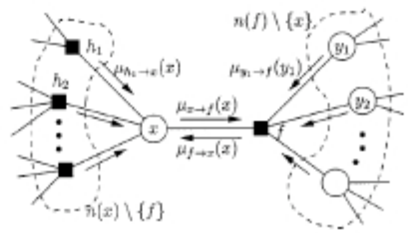
At a **factor** node, take the product of *f* with descendants; then perform not-sum over the parent of *f*

Known as the sum-product algorithm

Computing All Marginals

- Interested in computing all marginal functions $g_i(x_i)$
- One option is to repeat the sum-product for every single node
- Complexity of $\mathbf{O}(n^2)$
- Repeat computations can be avoided
- Sum-product algorithm for general trees

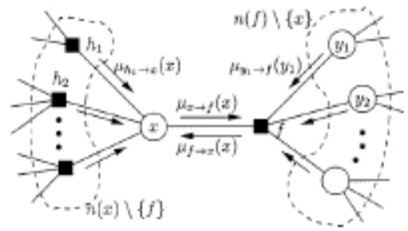
Sum Product Updates



Variable to local function:

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

Sum Product Updates



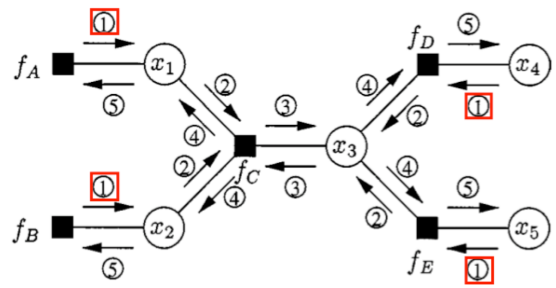
Variable to local function:

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

Local function to variable:

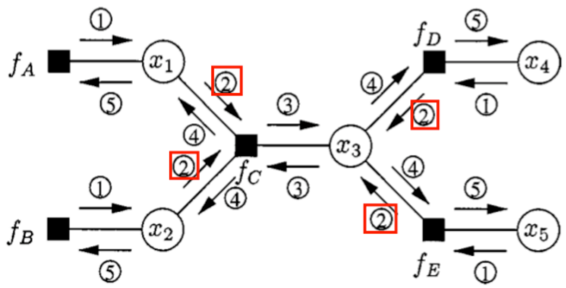
$$\mu_{f\to x}(x) = \sum_{\sim x}\left(f(x)\prod_{y\in n(f)\setminus\{x\}}\mu_{y\to f}(y)\right)$$

Example: Step 1



$$\mu_{f_A \to x_1}(x_1) = f_A(x_1)$$
$$\mu_{f_B \to x_2}(x_2) = f_B(x_2)$$
$$\mu_{x_4 \to f_D}(x_4) = 1$$
$$\mu_{x_5 \to f_E}(x_5) = 1$$

Example: Step 2



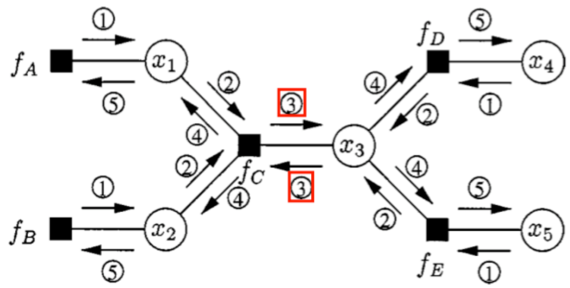
$$\mu_{x_1 \to f_C}(x_1) = \mu_{f_A \to x_1}(x_1)$$

$$\mu_{x_2 \to f_C}(x_2) = \mu_{f_B \to x_2}(x_2)$$

$$\mu_{f_D \to x_3}(x_3) = \sum_{\sim x_3} f_D(x_3, x_4)\mu_{x_4 \to f_D}(x_4)$$

$$\mu_{f_E \to x_3}(x_3) = \sum_{\sim x_3} f_D(x_3, x_5)\mu_{x_5 \to f_E}(x_5)$$

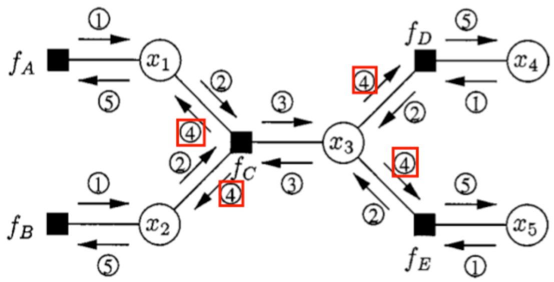
Example: Step 3



$$\mu_{f_C \to x_3}(x_3) = \sum_{\sim x_3} f_C(x_1, x_2, x_3)\mu_{x_1 \to f_C}(x_1)\mu_{x_2 \to f_C}(x_2)$$

$$\mu_{x_3 \to f_C}(x_3) = \mu_{f_D \to x_3}(x_3)\mu_{f_E \to x_3}(x_3)$$

Example: Step 4



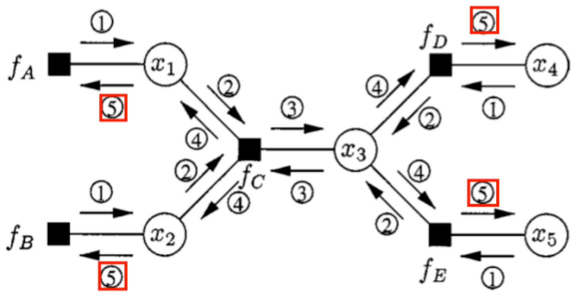
$$\mu_{f_C \to x_1}(x_1) = \sum_{\sim x_1} f_C(x_1, x_2, x_3)\mu_{x_2 \to f_C}(x_2)\mu_{x_3 \to f_C}(x_3)$$

$$\mu_{f_C \to x_2}(x_2) = \sum_{\sim x_2} f_C(x_1, x_2, x_3)\mu_{x_1 \to f_C}(x_1)\mu_{x_3 \to f_C}(x_3)$$

$$\mu_{x_3 \to f_D}(x_3) = \mu_{f_C \to x_3}(x_3)\mu_{f_E \to x_3}(x_3)$$

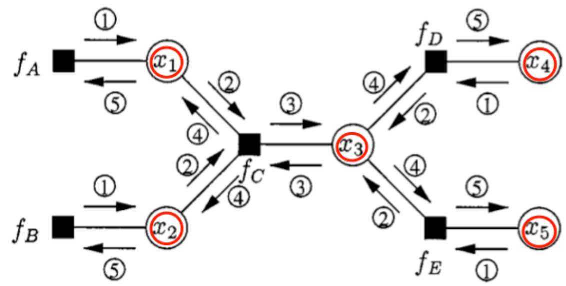
$$\mu_{x_3 \to f_E}(x_3) = \mu_{f_C \to x_3}(x_3)\mu_{f_D \to x_3}(x_3)$$

Example: Step 5



$$
\begin{aligned}
\mu_{x_1 \to f_A}(x_1) &= \mu_{f_C \to x_1}(x_1) \\
\mu_{x_2 \to f_B}(x_2) &= \mu_{f_C \to x_2}(x_2) \\
\mu_{f_D \to x_4}(x_4) &= \sum_{\sim x_4} f_D(x_3, x_4)\mu_{x_3 \to f_D}(x_4) \\
\mu_{f_E \to x_5}(x_5) &= \sum_{\sim x_5} f_D(x_3, x_5)\mu_{x_3 \to f_E}(x_5)
\end{aligned}
$$

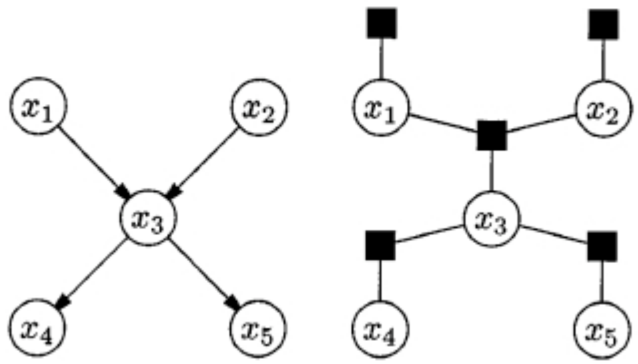
Example: Termination



Marginal function is the product of all incoming messages

$$
\begin{aligned}
g_1(x_1) &= \mu_{f_A \to x_1}(x_1)\mu_{f_C \to x_1}(x_1) \\
g_2(x_2) &= \mu_{f_B \to x_2}(x_2)\mu_{f_C \to x_2}(x_2) \\
g_3(x_3) &= \mu_{f_C \to x_3}(x_3)\mu_{f_D \to x_3}(x_3)\mu_{f_E \to x_3}(x_3) \\
g_2(x_2) &= \mu_{f_D \to x_4}(x_4) \\
g_5(x_5) &= \mu_{f_E \to x_5}(x_5)
\end{aligned}
$$

Belief Propagation in Bayes Nets



$f_A(x_1) = p(x_1)$ $f_B(x_2) = p(x_2)$ $f_C(x_1,x_2,x_3) = p(x_3|x_1,x_2)$

$f_D(x_3,x_4) = p(x_4|x_3)$ $f_E(x_3,x_5) = p(x_5|x_3)$

The Sum-Product Algorithm