ECE 175B: Probabilistic Reasoning and Graphical Models
Lecture 17: Message Passing on Factor Graphs

Florian Meyer & Ken Kreutz-Delgado

Electrical and Computer Engineering Department
University of California San Diego

UC San Diego
JACOBS SCHOOL OF ENGINEERING

Factor Graphs - I

Graph-Compatible Distribution Factorizations:

DAG: $P(\mathcal{X}) = \prod_{j=1}^{N} P(x_j \mid Pa(x_j))$, $\mathcal{X} = \{x_1, \cdots, x_N\}$

MN: $P(\mathcal{X}) = 1/Z \prod_{c \in \mathcal{C}} \phi_c(\mathcal{X}_c)$, $\mathcal{X}_c \subset \mathcal{X}$ is a clique.

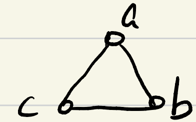
- If moralization is done on a BN, one arrives at a MN which is convenient for computation on serial chains.
- A factorization that is convenient for computation on general BN and MN trees is provided by a Factor Graph (FG).

Definition: Given a factorization $f(\mathcal{X}) = \prod_k \psi_k(\mathcal{X}_k)$, $\mathcal{X}_k \subset \mathcal{X}$, the associated Factor Graph (FG) has a factor node ■ for each factor $\psi_k(\mathcal{X}_k)$ and a variable node ○ for each variable $x_j \in \mathcal{X}_k$, such that for each $x_j \sim$ ○ a link is drawn between $x_j \sim$ ○ and its associated factor node $\psi_k(\mathcal{X}_k) \sim$ ■.

$\mathcal{X}_k$ is the factor cluster for ψ_k, and (*) $\prod_k \psi_k(\mathcal{X}_k)$ = Factor Product (FP)

Factor Graphs — II

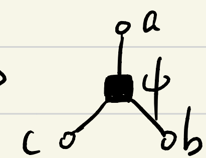
Three factorizations that have the same MN:



$\phi(a,b,c)$

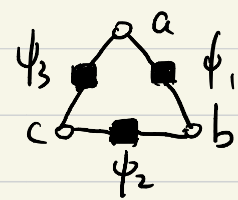
$\{a,b,c\}$ is the maximal clique.

(1). $P(a,b,c) = \phi(a,b,c) = \phi(a,b,c)$ $\Rightarrow$



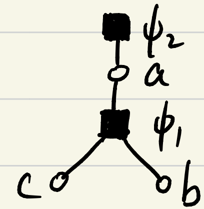
Factor Tree "admissible"

(2). $P(a,b,c) = \psi_1(a,b)\,\psi_2(b,c)\,\psi_3(c,a)$ $\Rightarrow$



Factor Loop "inadmissible"

(3). $P(a,b,c) = \psi_1(a,b,c)\,\psi_2(a)$. $\Rightarrow$

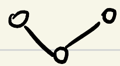
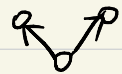
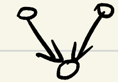


Factor Tree "admissible"

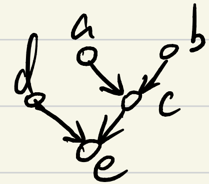
Note that factor clusters are not cliques on the Factor Graphs.

Factor Graphs – III: Factor Trees.

- Factor trees can be easily constructed for:

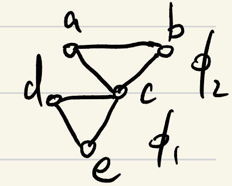
  Trees:  MN  BN or polytrees:  BN (immoral)

- $\boxed{\text{Example: Fig 8.43 Bishop}}$

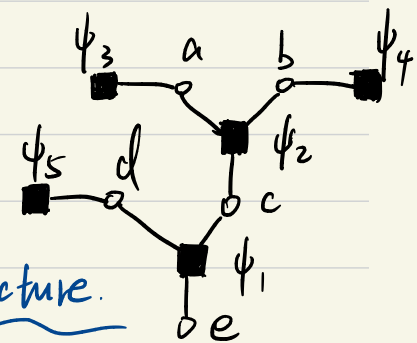
G: 

(polytree) BN.

$$p(a,b,c,d,e) = \underbrace{p(e|c,d)\,p(d)}_{\phi_1(c,d,e)}\,\underbrace{p(c|a,b)\,p(a)\,p(b)}_{\phi_2(a,b,c)} \Rightarrow$$

$M(G)$: 

$$\begin{aligned} p(a,b,c,d,e) &= p(e|c,d)\,p(d)\,p(c|a,b)\,p(a)\,p(b) \\ &= \psi_1(c,d,e)\,\psi_5(d)\,\psi_2(a,b,c)\,\psi_3(a)\,\psi_4(b) \end{aligned} \Rightarrow$$

FG: 

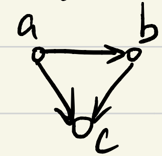
Tree Structure.

Factor Graph-IV: Factor Trees

- Factor trees can also be constructed for:
  - "Locally Loopy" BNs that are almost polytrees

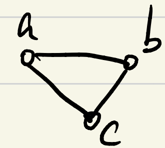
- Example: Fig 8.44 Bishop.

$\mathcal{G}=$



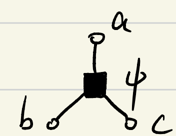
$$p(a,b,c)=\underbrace{p(c|b,a)\,p(b|a)\,p(a)}_{\phi(a,b,c)}$$

$m(\mathcal{G})=$

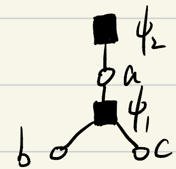


Admissible FGs:

$$p(a,b,c)=\overbrace{p(c|b,a)\,p(b|a)\,p(a)}^{\phi(a,b,c)} \longrightarrow$$



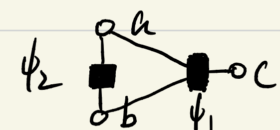
$$p(a,b,c)=\underbrace{p(c|b,a)\,p(b|a)}_{\phi_1(a,b,c)}\,\underbrace{p(a)}_{\phi_2(a)} \longrightarrow$$



} Trees.

Inadmissible FG:

$$p(a,b,c)=\underbrace{p(c|b,a)}_{\phi_1(a,b,c)}\,\underbrace{p(b|a)\,p(a)}_{\phi_2(a,b)} \longrightarrow$$



Not a Tree!

Handling Evidence

Let $\mathcal{Z}$ = all random variables on a graph.
Let $\mathcal{Y}$ = all observable variables / evidence on the graph
Let $\mathcal{X} = \mathcal{Z} \setminus \mathcal{Y}$ = unobservable (hidden) variables on the graph.
Thus $\mathcal{Z} = (\mathcal{X}, \mathcal{Y}) = \mathcal{X} \cup \mathcal{Y}$. $\mathcal{X} \cap \mathcal{Y} = \phi$.

$p(\mathcal{Z}) = \prod_j \phi_j(\mathcal{Z}_j) \cdot \frac{1}{Z}$, where ϕ_j are factors and $\mathcal{Z}_j$ are factor clusters

Let $p(\mathcal{X}, \mathcal{Y}) = \frac{1}{Z} \prod_j \phi_j(\mathcal{X}_j, \mathcal{Y}_j)$ where $\mathcal{Z}_j = \mathcal{X}_j \cup \mathcal{Y}_j$.

$$p(\mathcal{X}|\mathcal{Y}) = \frac{p(\mathcal{X}, \mathcal{Y})}{p(\mathcal{Y})} = \frac{1}{Z\,p(\mathcal{Y})} \prod_j \phi_j(\mathcal{X}_j, \mathcal{Y}_j)$$

Define $p(\mathcal{X}|\mathcal{Y}) = \frac{1}{Z(\mathcal{Y})} \prod_j \psi_j(\mathcal{X}_j)$ the $\boxed{\text{observation-conditional FP}}$.

where $Z(\mathcal{Y}) = Z\,p(\mathcal{Y})$, $\psi_j(\mathcal{X}_j) = \psi_j(\mathcal{X}_j, \mathcal{Y}_j) = \phi_j(\mathcal{X}_j, \mathcal{Y}_j)$

This FP yields $\boxed{\text{observation-based FG}}$.

Admissible Factor Products and Inference

(Henceforth, equation number (B-8.#) refers to Eq. # in chapter 8 of Bishop)

The material presented is drawn for section 8.4.3 - 8.4.7 of Bishop.

We consider distribution $p(\mathcal{X}|y)$ which have admissible Factor Products (FPs):

$$Z(y)\, p(\mathcal{X}|y) = \prod_{\gamma} f_\gamma(\mathcal{X}_\gamma) \qquad (B\text{-}8.59)$$

where "admissible" means the Factor Graph is a tree, which is true if the original graph is a tree, polytree, or "locally loopy".

In this case, the Factor Graph Algorithm (FGA) can be used to efficiently solve the two inference problems of finding:

Types of FGA

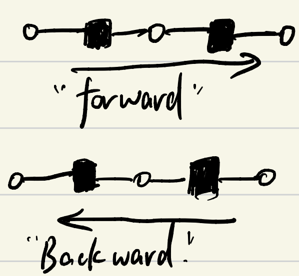
SUM-PRODUCT: $p(x|y) \propto \sum_{\mathcal{X}\setminus x} \prod_{\gamma} f_\gamma(\mathcal{X}_\gamma), \quad x \in \mathcal{X}$

MAX-PRODUCT: $\hat{\mathcal{X}} = \arg\max_{\mathcal{X}} \prod_{\gamma} f_\gamma(\mathcal{X}_\gamma)$

Sum – Product FGA – I

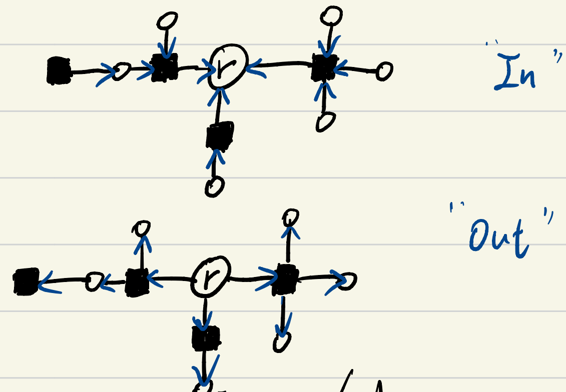
- The derivation of the SP-FGA is quite detialed. (c § 8.4.4 in Bishop for reference) Therefore, only the derived algorithm will be stated.
- Given a (bipartite) Factor Tree, choose any variable node to serve as root node. Call this node x_r.
- The SP-FGA is a generalization of the "Forward-Backward" Algorithm on a chain to an "In-Out" algorithm on a tree.

Chain:



"Forward-Backward" Algorithm.

Factor Tree:



"In-Out" Algorithm / Message-Passing Algorithm.

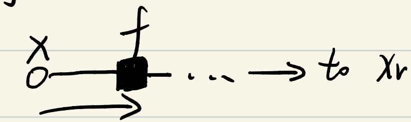
SP-FGA - II.

- Messages passed from variable nodes to factor nodes are denoted as $\mu_{x\to f}(X)$
- Messages passed from factor nodes to variable nodes are denoted as $\mu_{f\to x}(X)$.

The algorithm begins with the inward sweep, which is initialized as follows:

(A). Initialization at leaf nodes.

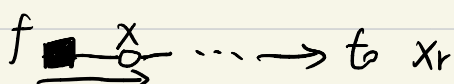
variable leaf node:



$\mu_{x\to f}(X) \equiv 1$

see Figure B-8.49a and Eq. (B-8.70)

factor leaf node:



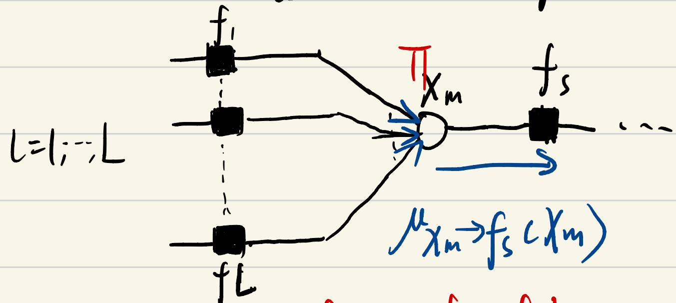
$\mu_{f\to x}(X) \equiv f(x)$

see Figure B-8.49b and Eq. (B-8.71)

Procedure B(i) and B(ii) below are used for both the inward and outward sweeps. When the inward messages are finally sent to, and processed by the root node X_r, then X_r broadcasts outward to begin the outward sweep, which ends when the leaf nodes have processed the outward info.

SP-FGA - III

(B)(i). Variable X_m to factor f_s message passing. (C.f. Figure B-8.48 and E.q. (B-8.69))

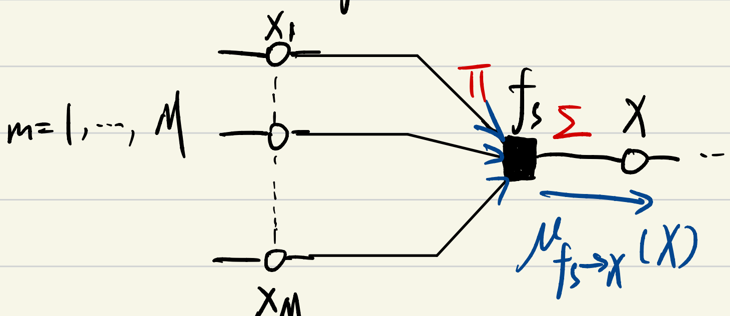


$ne(X_m) = \{f_1, \cdots, f_L, f_s\}$

$$\mu_{X_m \to f_s}(X_m) = \prod_{l \in ne(X_m) \setminus f_s} \mu_{f_l \to X_m}(X_m)$$

i.e. multiply all messages that do not come from f_s, i.e., from all the other neighboring factor nodes of X_m except f_s.

(B)(ii). Factor f_s to variable X message passing. (C.f. Figure B-8.47 and Eq. (B-8.66))



$ne(f_s) = \{X_1, \cdots, X_M, X\} \triangleq \mathcal{X}_s$

$$\mu_{f_s \to X}(X) = \underbrace{\sum_{\mathcal{X}_s \setminus X}}_{③} \underbrace{f_s(\mathcal{X}_s)}_{②} \underbrace{\prod_{m \in ne(f_s) \setminus X} \mu_{X_m \to f_s}(X_m)}_{①}$$

i.e. multiply all messages except the one from X, ① along with the factor $f_s(\mathcal{X}_s)$ ②, then marginalize out ③ all variables in $\mathcal{X}_s$ except X.

SP-FGA - IV

Once the outward sweep is completed (when all leaf nodes have processed the outward messages), then for each $x \in \mathcal{X}$, one computes the conditional marginal probabilities:

(C). $$\tilde{p}(x|y) = \prod_{s \in ne(x)} \mu_{f_s \to x}(x)$$ (c.f. Eq.-B-8.63 for non-conditional case)

→ unnormalized conditional probability.

And the normalization factor is computed as:

$$Z(y) = \sum_{x\text{-values}} \tilde{p}(x|y)$$ (only need to do for one variable node x)

→ sum over all values of the variable x.

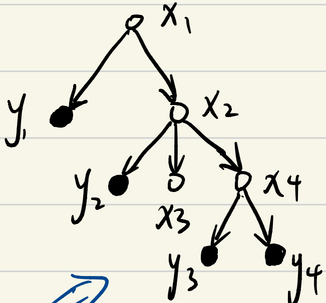
Then $p(x|y) = \frac{1}{Z(y)} \tilde{p}(x|y)$ = evidence-conditional probability.

This algorithm has linear complexity w.r.t. the number of nodes in $\mathcal{X}$.

Example of SP-FGA - I

Extension of Bishop example in Figure 8.51 to include evidence.

BN:



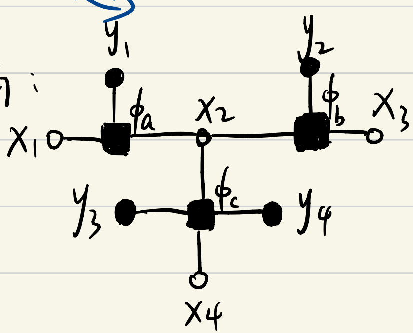
$$\tilde{p}(x|y) = p(x, y)$$

$$= \underbrace{p(y_4|x_4)\, p(y_3|x_4)\, p(x_4|x_2)}_{f_c(x_2, x_4)}\ \underbrace{p(y_2|x_2)\, p(x_3|x_2)}_{f_b(x_2, x_3)}\ \underbrace{p(y_1|x_1)\, p(x_2|x_1)\, p(x_1)}_{f_a(x_1, x_2)}$$

$$f_c(x_2, x_4) = \phi_c(x_2, x_4; y_3, y_4) \qquad f_b(x_2, x_3) = \phi_b(x_2, x_3; y_2) \qquad f_a(x_1, x_2) = \phi_a(x_1, x_2; y_1)$$

$\Rightarrow$ Condition on evidence y $\Rightarrow$

FG:



$\Longrightarrow$

Evidence-Conditional Factor Graph $\tilde{p}(x|y) \propto f_a(x_1, x_2)\, f_b(x_2, x_3)\, f_c(x_2, x_4)$

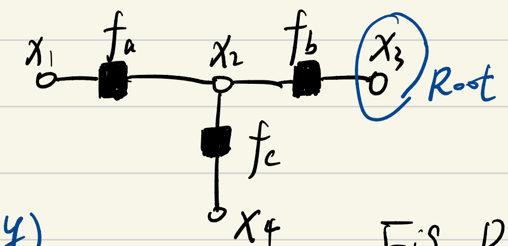
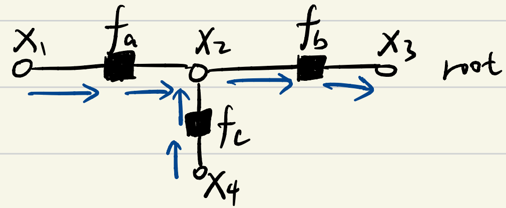


Fig. B-8.51

Example of SP-FGA - II

- Inwards sweep towards root:



$\mu_{x_1 \to f_a}(x_1) \equiv 1$, $\mu_{x_4 \to f_c}(x_4) \equiv 1$ (a) Initialization

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

(b-ii). multiply all messages that do not come from x_2, Marginalize all f_a / f_c-cluster variables except x_2.

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

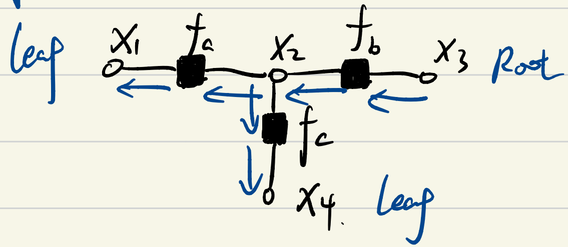
(b-i). multiply all messages that do not come from f_b.

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

(b-ii). multiply all messages that do not come from x_3, Marginalize all f_b-cluster variables except x_3.

Example of SP-FGA — III.

- Outward sweep towards leafs



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

(a). Because messages from f_b are ignored and only the initialization messages $\equiv 1$ exists.

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

(b-ii). Multiply all messages that do not come from x_2, Marginalize all f_b-cluster variables except x_2.

$$\mu_{x_2 \to f_a}(x_2) = \mu_{f_b \to x_2}(x_2)\,\underbrace{\mu_{f_c \to x_2}(x_2)}_{\text{computed in inward sweep.}}$$

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

(b-i). Multiply all messages that do not come from f_c / f_a.

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

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

(b-ii). Multiply all messages that do not come from x_1 / x_4, Marginalize all f_a / f_c-cluster variables except x_1 / x_4.

Example of SP-FGA - IV.

Now that we have computed the various messages.

$\mu_{f_l \to x_k}(x_k)$ for $l = a, b, c$, and $k = 1, 2, 3, 4$,

We can compute any observation-conditional marginal $p(x_k | y)$:

Overall, we have $Z(y)\, p(\mathcal{X} | y) = \tilde{p}(\mathcal{X} | y) = \prod_l f_l(\mathcal{X}_l)$

Then for $\tilde{p}(x_k | y)$, take $k=2$ for example:

$$Z(y)\, p(x_2 | y) = \tilde{p}(x_2 | y) = \prod_{l \in ne(x_2)} \mu_{f_l \to x_2}(x_2)$$

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

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

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

$$= \sum_{\mathcal{X} \setminus x_2} \prod_{l \in ne(x_2)} f_l(\mathcal{X}_l)$$