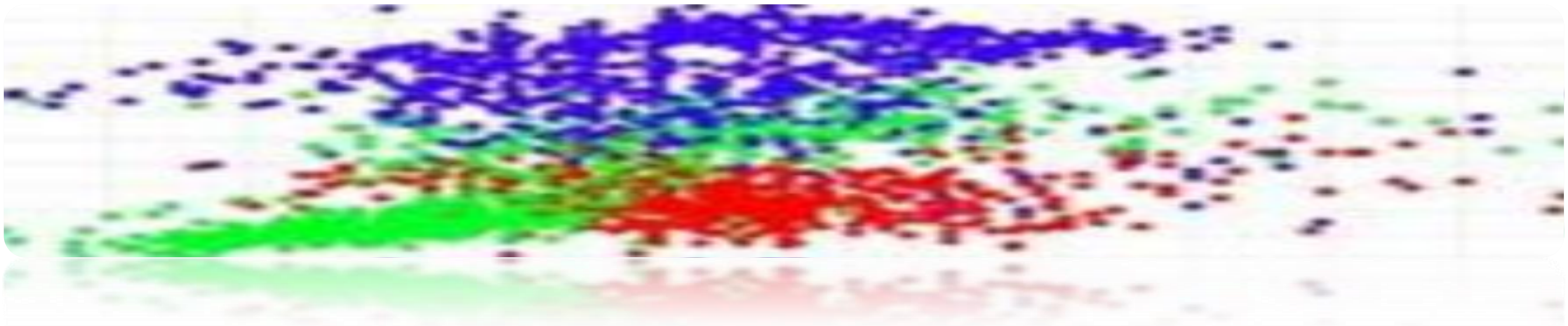


INFERENCE IN GRAPHICAL MODELS

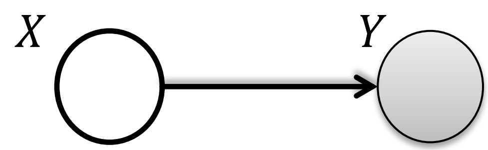
Outline

- Inference in chains
- Inference in tree structures
  - Factor graphs
  - Sum-product algorithm
  - Max-sum algorithm
- Inference in general graphs
  - Junction tree algorithm
  - Loopy belief propagation

Inference on chains (1)

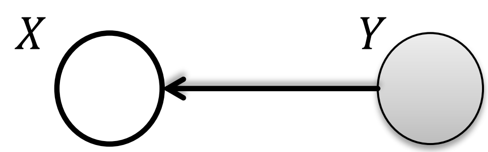
What is the marginal distribution on Y?

$$P(Y) = \sum_{X=x} P(X,Y) = \sum_{X=x} P(Y|X)P(X)$$



Now that we know $P(Y)$, what is $P(X|Y)$?

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}$$



Given $P(X|Y)$ and $P(Y)$, we can find $P(X,Y)$ (the joint distribution)

Inference on chains (2)

Directed chain can easily be converted into equivalent (w.r.t. conditional independence properties) undirected chain (by omitting arrows)

$$P(\mathbf{X}) = \frac{1}{Z}\psi_{1,2}(X_1, X_2)\psi_{2,3}(X_2, X_3) \ldots \psi_{n-1,n}(X_{n-1}, X_n)$$

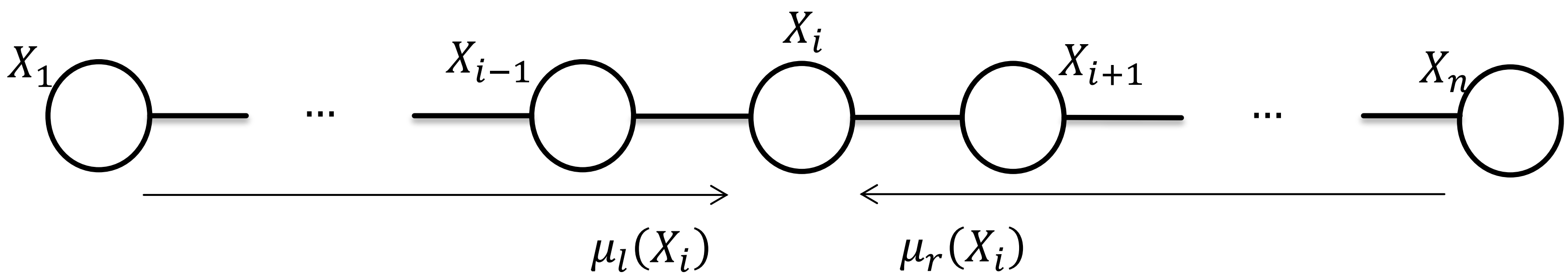
Marginal distribution at a given X_i:

$$P(X_i) = \sum_{X_1} \ldots \sum_{X_{i-1}} \sum_{X_{i+1}} \ldots \sum_{X_n} P(\mathbf{X})$$

$$= \sum_{X_1} \ldots \sum_{X_{i-1}} \sum_{X_{i+1}} \ldots \sum_{X_n} \frac{1}{Z}\psi_{1,2}(X_1, X_2)\psi_{2,3}(X_2, X_3) \ldots \psi_{n-1,n}(X_{n-1}, X_n)$$

Observations: 1. Summations depend on certain factors
2. Summations can be rearranged by distribution law, i.e., ab+ac=a(b+c)

Inference on chains (3)



$$P(\mathbf{X}) = \frac{1}{Z}\psi_{1,2}(X_1, X_2)\psi_{2,3}(X_2, X_3)\dots\psi_{n-1,n}(X_{n-1}, X_n)$$

Marginal distribution at a given X_i:

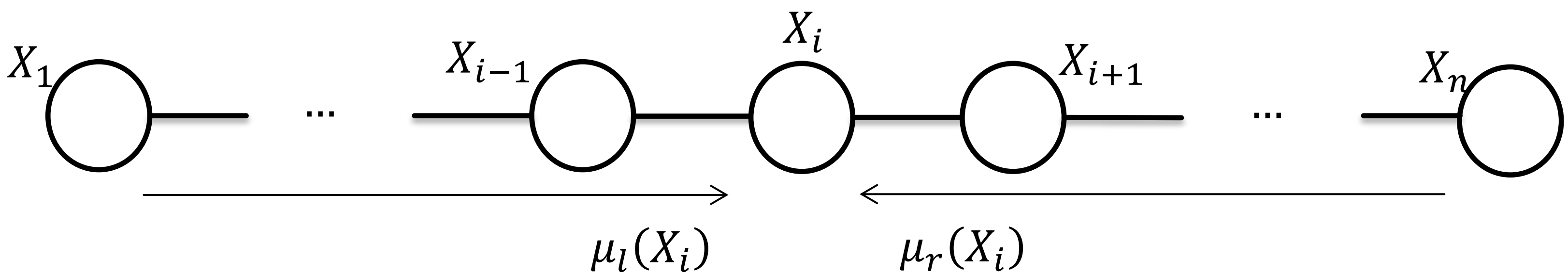
$$P(X_i) = \frac{1}{Z}\left[\sum_{X_{i-1}}\psi_{i-1,i}(X_{i-1}, X_i)\dots\left[\sum_{X_1}\psi_{1,2}(X_1, X_2)\right]\dots\right]$$

$$\mu_l(X_i)$$

$$\left[\sum_{X_{i+1}}\psi_{i,i+1}(X_{i+1}, X_i)\dots\left[\sum_{X_n}\psi_{n-1,n}(X_{n-1}, X_n)\right]\dots\right]$$

$$\mu_r(X_i)$$

Inference on chains (4)



$$P(X_i) = \frac{1}{Z}\mu_l(X_i)\mu_r(X_i)$$

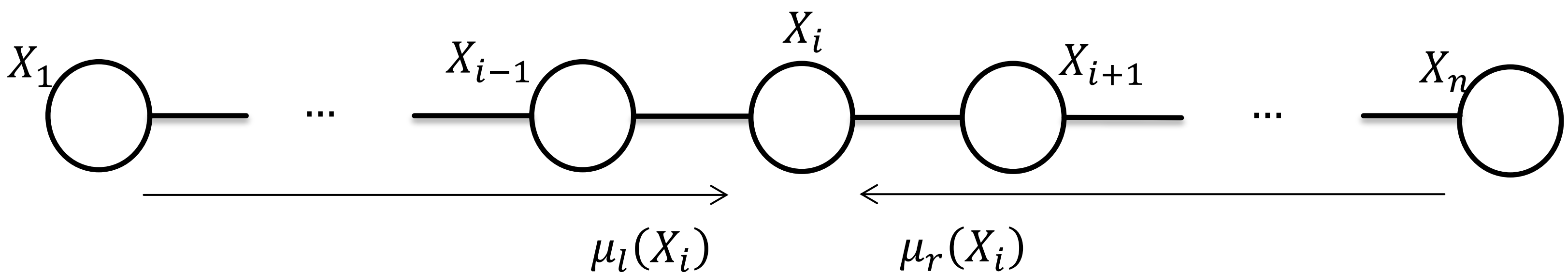
Recurrence:

$$\mu_l(X_i) = \sum_{X_{i-1}} \psi_{i-1,i}(X_{i-1}, X_i)\,[\mu_l(X_{i-1})]$$

$$\mu_r(X_i) = \sum_{X_{i+1}} \psi_{i+1,i}(X_{i+1}, X_i)\,[\mu_r(X_{i+1})]$$

Note: each message comprises k values (where k is the number of states of X_i), one for every value of X_i

Inference on chains (5)



$$P(X_i) = \frac{1}{Z}\mu_l(X_i)\mu_r(X_i)$$

Recurrence:

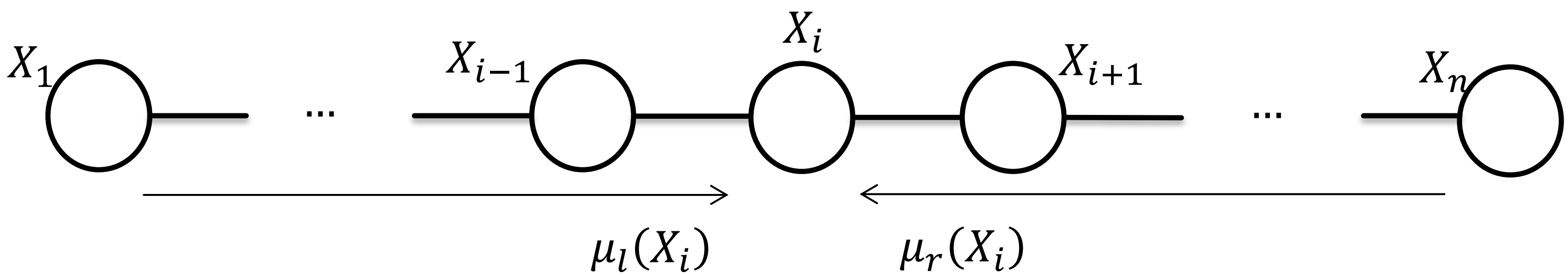
$$\mu_l(X_i) = \sum\nolimits_{X_{i-1}} \psi_{i-1,i}(X_{i-1}, X_i)\,[\mu_l(X_{i-1})]$$

Start with $\mu_l(X_2) = \sum_{X_1} \psi_{1,2}(X_1, X_2)$ and compute $\mu_l(X_i)$ forwards on the chain

$$\mu_r(X_i) = \sum\nolimits_{X_{i+1}} \psi_{i+1,i}(X_{i+1}, X_i)\,[\mu_r(X_{i+1})]$$

Start with $\mu_l(X_{n-1}) = \sum_{X_n} \psi_{n-1,n}(X_{n-1}, X_n)$ and compute $\mu_l(X_i)$ backwards on the chain

Inference on chains (6)



$$P(X_i) = \frac{1}{Z}\mu_l(X_i)\mu_r(X_i)$$

What about Z?

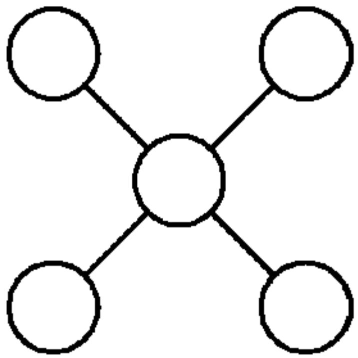
$$Z = \sum_{X_i} \mu_l(X_i)\mu_r(X_i)$$

Forward-backward algorithm

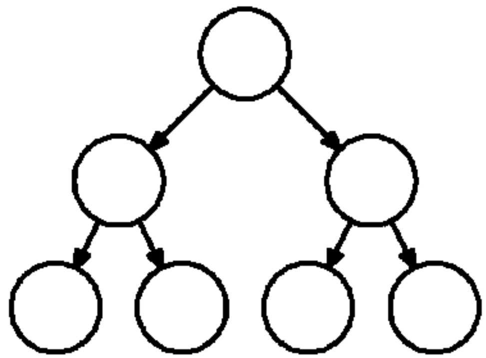
(1) Start with $\mu_l(X_2) = \sum_{X_1} \psi_{1,2}(X_1, X_2)$ and compute $\mu_l(X_n)$ forwards on the chain (store $\mu_l(X_i)$ for each variable X_i)
(2) Start with $\mu_l(X_{n-1}) = \sum_{X_n} \psi_{n-1,n}(X_{n-1}, X_n)$ and compute $\mu_l(X_1)$ backwards on the chain (store $\mu_r(X_i)$ for each variable X_i)

Inference in trees (1)

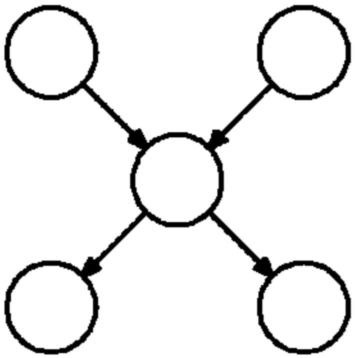
- Examples from C.Bishop: PRML



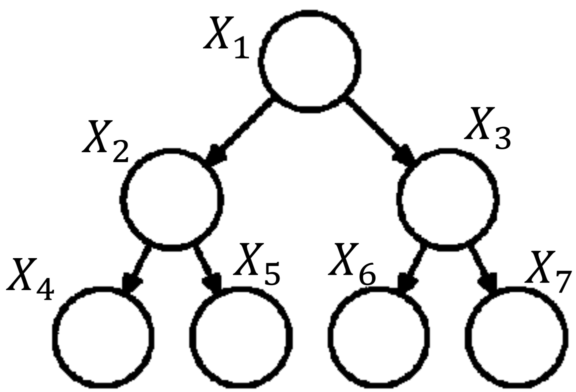
Undirected tree



Directed tree



Polytree



Joint distribution:

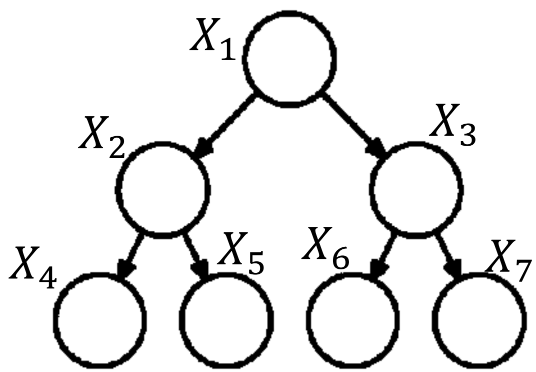
$$P(\mathbf{X}) = P(X_1)P(X_2|X_1)P(X_3|X_1)$$
$$P(X_4|X_2)P(X_5|X_2)$$
$$P(X_6|X_3)P(X_7|X_3)$$

Generally:

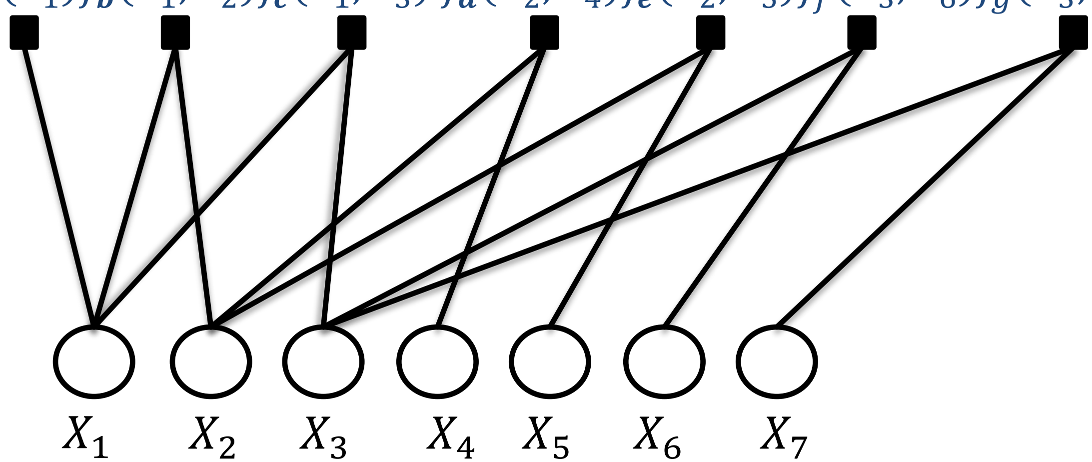
$$P(\mathbf{X}) = \prod_s f_s(\mathbf{X}_s)$$

Inference in trees: Factor graphs (1)

$$P(\mathbf{X}) = P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|X_2)P(X_5|X_2)P(X_6|X_3)P(X_7|X_3)$$



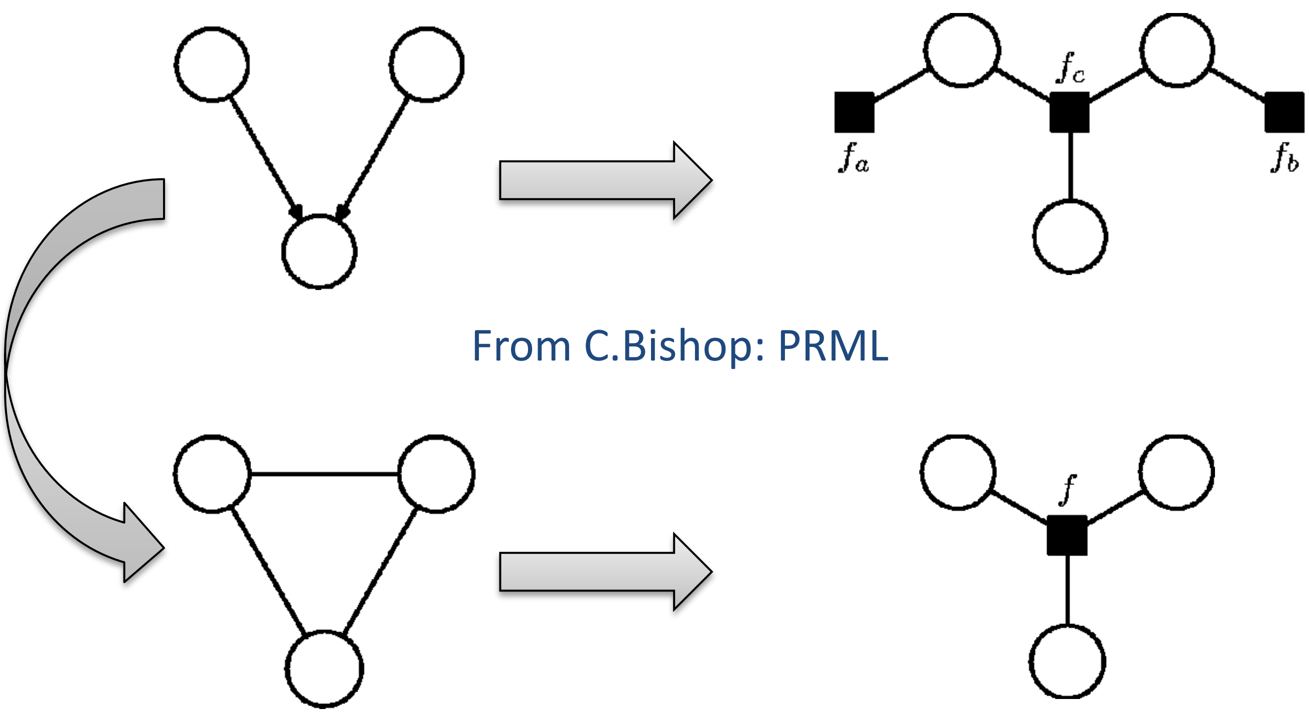
$$P(\mathbf{X}) = f_{\boldsymbol{a}}(X_1)f_{\boldsymbol{b}}(X_1, X_2)f_{\boldsymbol{c}}(X_1, X_3)\, f_{\boldsymbol{d}}(X_2, X_4)f_{\boldsymbol{e}}(X_2, X_5)f_f(X_3, X_6)f_g(X_3, X_7)$$



Inference in trees: Factor graphs (2)

- **Factor graph:** bipartite graph $G(V \cup F, E)$, where $\{v, f\} \in E$ if and only if variable $v \in V$ occurs in factor $f \in F$

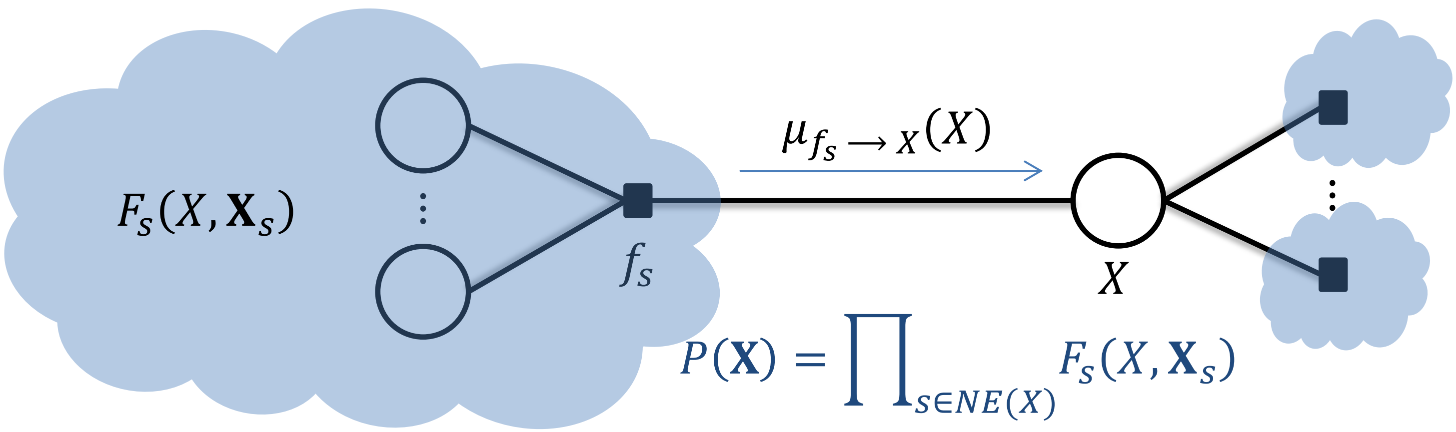
- Examples:



From C.Bishop: PRML

Inference in trees: Sum-product algorithm (1)

- Exact inference for finding marginal distributions based on message passing in factor graphs



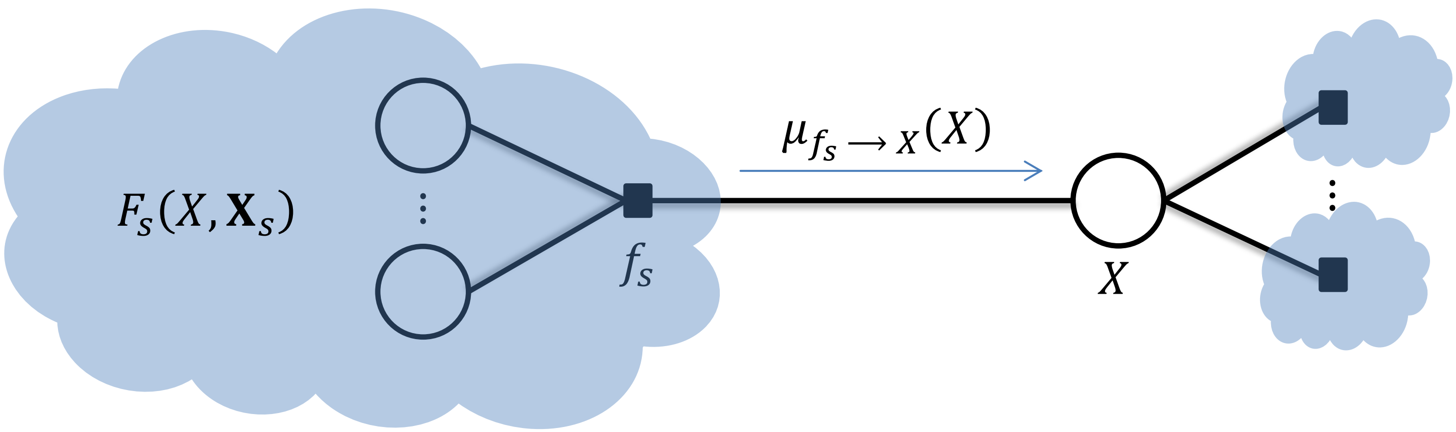
$$P(\mathbf{X}) = \prod_{s \in NE(X)} F_s(X, \mathbf{X}_s)$$

$$P(X) = \sum_{\mathbf{X} \setminus X} P(\mathbf{X}) = \prod_{s \in NE(X)} \left[\sum_{\mathbf{X}_s} F_s(X, \mathbf{X}_s) \right]$$
$$= \prod_{s \in NE(X)} \mu_{f_s \to X}(X)$$

$$\mu_{f_s \to X}(X) \equiv \sum_{\mathbf{X}_s} F_s(X, \mathbf{X}_s)$$

Inference in trees: Sum-product algorithm (2)

- Exact inference for finding marginal distributions based on message passing in factor graphs

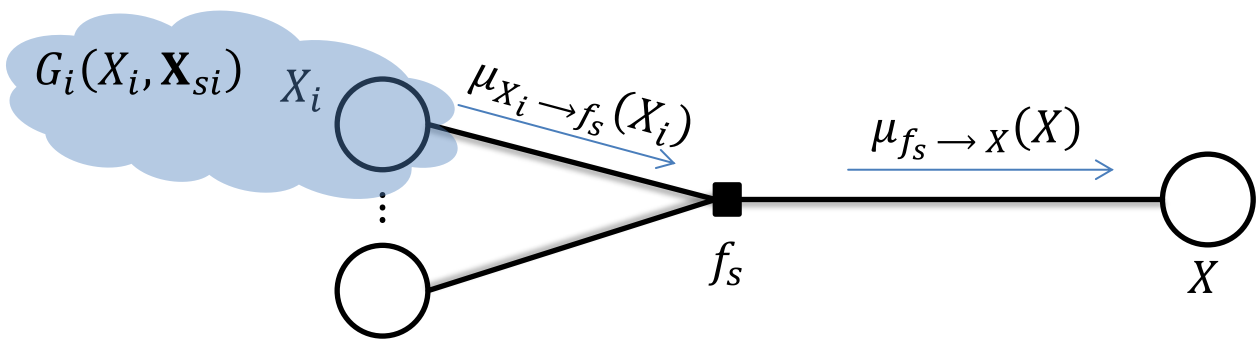


Compute recursively:

$$\mu_{f_s \to X}(X) \equiv \sum_{\mathbf{X}_s} F_s(X, \mathbf{X}_s) = \underbrace{\sum_{X_1} \dots \sum_{X_m} f_s(X, X_1, \dots, X_m)}_{\text{Local marginal distribution at } X \text{ with respect to } f_s} \prod_{i \in NE(f_s) \setminus X} \mu_{X_i \to f_s}(X_i)$$

Inference in trees: Sum-product algorithm (3)

- Exact inference for finding marginal distributions based on message passing in factor graphs



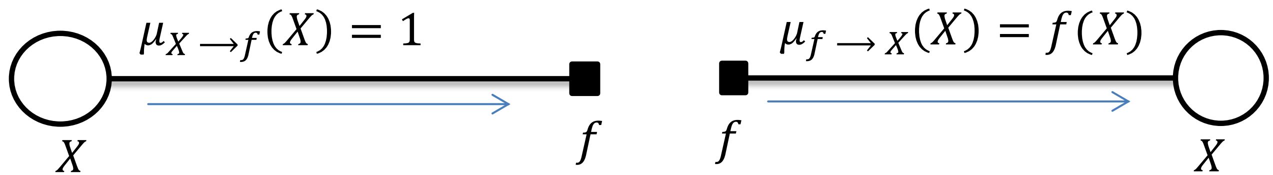
$$\mu_{X_i \to f_s}(X_i) = \underbrace{\sum\nolimits_{\mathbf{X}_{si}} G_i(X_i, \mathbf{X}_{si})}_{\text{Marginal distribution of } X_i \text{ with respect to } G_i} \qquad G_i(X_i, \mathbf{X}_{si}) = \prod\nolimits_{l \in NE(X_i) \backslash f_s} F_l(X_i, \mathbf{X}_{si})$$

Compute recursively:

$$\mu_{X_i \to f_s}(X_i) = \prod\nolimits_{l \in NE(X_i) \backslash f_s} \left[\sum\nolimits_{\mathbf{X}_{si}} F_l(X_i, \mathbf{X}_{si}) \right] = \prod\nolimits_{l \in NE(X_i) \backslash f_s} \mu_{f_l \to X_i}(X_i)$$

Inference in trees: Sum-product algorithm (4)

First message from leaves

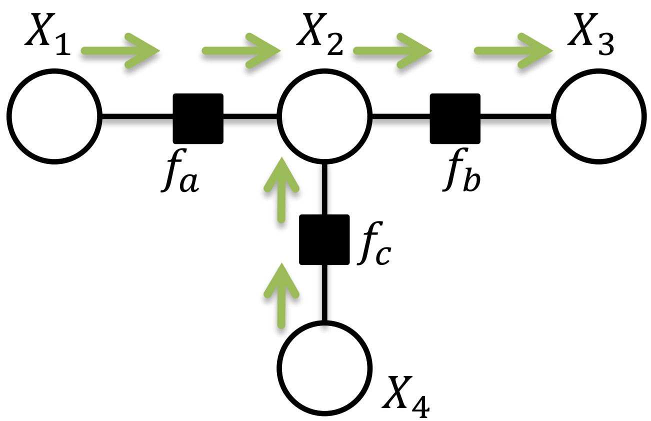


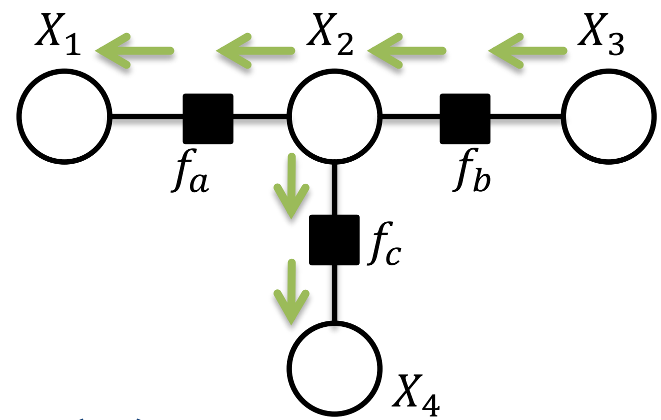

```
(1) Pick arbitrary node as root
(2) Propagate messages from the leaves to root and store
received messages at every node
(3) Propagate messages from root to leaves and store
received messages at every node.

Compute the product of received messages at a given node
for which the marginal is required (normalize if
necessary)
```

Sum-product algorithm example

- From C. Bishop, PRML





$$\mu_{X_1 \to f_a}(X_1) = 1$$
$$\mu_{f_a \to X_2}(X_2) = \sum_{X_1} f_a(X_1, X_2)$$

$$\mu_{X_4 \to f_c}(X_4) = 1$$
$$\mu_{f_c \to X_2}(X_2) = \sum_{X_4} f_c(X_2, X_4)$$

$$\mu_{X_2 \to f_b}(X_2) = \mu_{f_a \to X_2}(X_2)\mu_{f_c \to X_2}(X_2)$$
$$\mu_{f_b \to X_3}(X_3) = \sum_{X_2} f_b(X_2, X_3)\, \mu_{X_2 \to f_b}(X_2)$$

$$\mu_{X_3 \to f_b}(X_3) = 1$$
$$\mu_{f_b \to X_2}(X_2) = \sum_{X_3} f_b(X_2, X_3)$$

$$\mu_{X_2 \to f_a}(X_2) = \mu_{f_b \to X_2}(X_2)\mu_{f_c \to X_2}(X_2)$$
$$\mu_{f_a \to X_1}(X_1) = \sum_{X_2} f_a(X_1, X_2)\, \mu_{X_2 \to f_b}(X_2)$$

$$\mu_{X_2 \to f_c}(X_2) = \mu_{f_a \to X_2}(X_2)\mu_{f_b \to X_2}(X_2)$$
$$\mu_{f_c \to X_4}(X_4) = \sum_{X_3} f_c(X_2, X_4)\, \mu_{X_2 \to f_c}(X_2)$$

Inference in trees: Max-sum algorithm (1)

- Finds $\mathbf{X}$ that maximizes $P(\mathbf{X})$
- Computes $\max_{\mathbf{X}} P(\mathbf{X})$

Generally: $\underset{X_1}{\operatorname{argmax}} P(X_1, X_2) \neq \underset{X_1}{\operatorname{argmax}} P(X_1)$

The value of X_1 that maximizes the joint probability distribution over X_1, X_2 is not the same as the value of X_1 that maximizes the marginal distribution over X_1.

For any tree-structured factor graph:

$$\max_{\mathbf{X}} P(\mathbf{X}) = \max_{X_n} \prod_{f_s \in NE(X_n)} \max_{\mathbf{X}_s} f_s(X_n, \mathbf{X}_s)$$

➔ Max-product (…so, what about the max-sum algorithm)

Inference in trees: Max-sum algorithm (2)

- Numerically it is safer to compute $\ln\left(\max_{\mathbf{X}} P(\mathbf{X})\right)$
- We know that: $\operatorname{argmax}_{\mathbf{X}} P(\mathbf{X}) = \operatorname{argmax}_{\mathbf{X}} \ \ln P(\mathbf{X})$

Fact: $\ln\left(\max_{\mathbf{X}} P(\mathbf{X})\right) = \max_{\mathbf{X}} \ \ln P(\mathbf{X})$

Moreover: $\max(a + b, a + c) = a + \max(b, c)$ (distributive law)

Analogous message passing as in sum-product algorithm …

Inference in trees: Max-sum algorithm (3)

- Analogously to sum-product…

  Initialization at leaf nodes:

  $\mu_{X\to f}(X) = 0$, if X is a leaf

  $\mu_{f\to X}(X) = \ln f(X)$, if f is a leaf

  Recurrence:

  $\mu_{f_s\to X}(X) = \max_{\mathbf{X}_s\setminus X}\left(\ln f_s(\mathbf{X}_s) + \sum_{Y\in\mathbf{X}_s\setminus X}\mu_{Y\to f_s}(Y)\right)$

  Store $\operatorname{argmax}_{\mathbf{X}_s\setminus X}\left(\ln f_s(\mathbf{X}_s) + \sum_{Y\in\mathbf{X}_s\setminus X}\mu_{Y\to f_s}(Y)\right)$ for each of these messages (this is the best configuration of the other variables in f_s)

  $\mu_{X\to f_s}(X) = \sum_{f_t\in NE(X)\setminus f_s}\mu_{f_s\to X}(X)$

  Termination at root node X:

  $\max_{\mathbf{X}}\ln P(\mathbf{X}) = \max_{X}\left(\sum_{f_s\in NE(X)}\mu_{f_s\to X}(X)\right)$, and remember $\operatorname{argmax}_{X}\left(\sum_{f_s\in NE(X)}\mu_{f_s\to X}(X)\right)$ (the best configuration of X)

Inference in general graphs: Junction tree algorithm

If graph is directed make it undirected through **moralization**

Triangulate the undirected graph

Let the **maximal cliques** be the nodes of a new graph, where two cliques are connected if they have at least one node in common

In this new clique graph, let the weight on an edge between two cliques be the number of nodes they have in common

Return the **maximum spanning tree** on the clique graph

- Exact inference through the max-sum or sum-product algorithm
- Intractable when the given graph contains large cliques…

Loopy belief propagation for approximate inference

- Sum-product and Max-sum algorithms can be also applied to general graphs by “looping” through the cycles of nodes
- The algorithm is then called "loopy" belief propagation
- Initialization and scheduling of message updates must be adjusted slightly because graphs might not contain any leaves
- E.g., initialize all variable messages to 1 and use the same message passing strategies as above in every iteration
- The precise conditions under which loopy belief propagation will converge are still not well understood (on graphs containing a single loop it converges in most cases, but the probabilities obtained might be incorrect)
- There are other approximate methods for marginalization including, Expectation Propagation, Variational Bayes, Markov Chain Monte Carlo methods, etc.