# Inference with Graphical Models – Propagation Algorithms

Kenji Fukumizu
The Institute of Statistical Mathematics

Computational Methodology in Statistical Inference II

# Inference with Graphical Model

- Assumption in this part

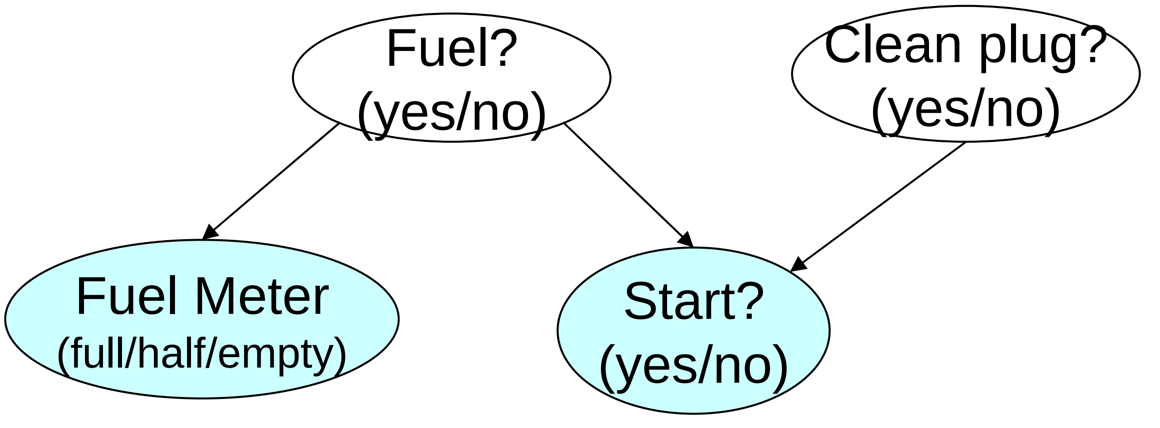
  Every variable takes values in a finite set.

- Probabilistic Inference

  $$p(Y \mid X)$$

  $X$: observed (evidence)
  $Y$: variable for inference

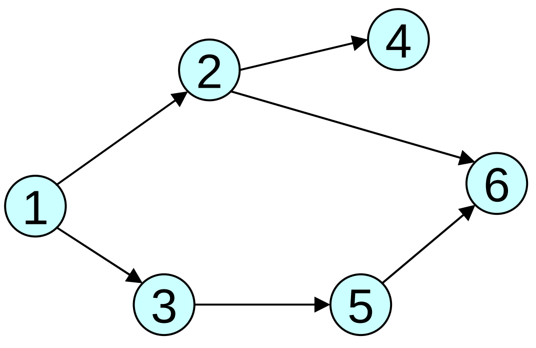
  - Example: diagnosis for car start




P(Clean plug = no | No start, Fuel meter = half)

# Inference with Graphical Model

- Probabilistic inference with graphical model




$$p(X) = p(X_1)p(X_2 \mid X_1)p(X_3 \mid X_1)$$
$$\times p(X_4 \mid X_2)p(X_5 \mid X_3)p(X_6 \mid X_2, X_5)$$

Given a value of $X_6 = e$, compute the probability of $X_1$

$$p(X_1 \mid X_6 = e) = \frac{p(X_1, X_6 = e)}{p(X_6 = e)}$$

# Inference with Graphical Model

Assume each variable takes K values

- Naïve method

$$p(X_1, X_6 = e) = \sum_{X_2}\sum_{X_3}\sum_{X_4}\sum_{X_5} p(X_1, X_2, X_3, X_4, X_5, X_6 = e)$$

($K^5$ operations)

$$p(X_6 = e) = \sum_{X_1} p(X_1, X_6 = e)$$

(K operations)

$$p(X_1 \mid X_6 = e) = \frac{p(X_1, X_6 = e)}{p(X_6 = e)}$$

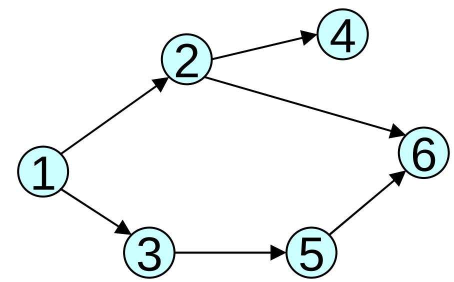
(K operations)

In total: $K^5 + 2K$ operations are needed.

# Inference with Graphical Model

- Efficient method:

  Elimination or successive marginalization




$$p(X_1, X_6 = e) = \sum_{X_2, X_3, X_4, X_5} p(X_1) p(X_2 \mid X_1) p(X_3 \mid X_1) p(X_4 \mid X_2) \times p(X_5 \mid X_3) p(X_6 = e \mid X_2, X_5)$$

$$= p(X_1) \sum_{X_2} p(X_2 \mid X_1) \sum_{X_3} p(X_3 \mid X_1) \sum_{X_4} p(X_4 \mid X_2) \boxed{\sum_{X_5} p(X_5 \mid X_3) p(X_6 = e \mid X_2, X_5)}$$

$$= p(X_1) \sum_{X_2} p(X_2 \mid X_1) \sum_{X_3} p(X_3 \mid X_1) m_5(X_2, X_3, X_6 = e) \boxed{\sum_{X_4} p(X_4 \mid X_2)}$$

$$= p(X_1) \sum_{X_2} p(X_2 \mid X_1) \boxed{\sum_{X_3} p(X_3 \mid X_1) m_5(X_2, X_3, X_6 = e)}$$

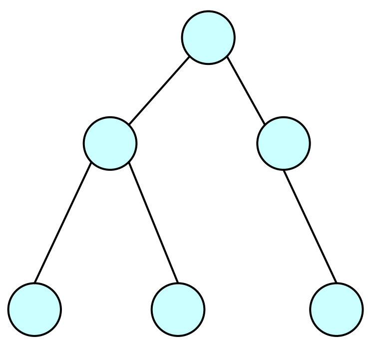
$$= p(X_1) \boxed{\sum_{X_2} p(X_2 \mid X_1) m_3(X_1, X_2, X_6 = e)}$$

In total: $K^3$ (+ K) + $K^3$ + $K^2$ + 2K operations are needed.

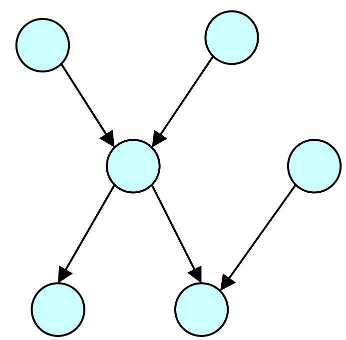
The efficiency depends on the number of variables in the factors.

# Tree

- The previous elimination method works most efficiently for trees.
- Tree: a (directed or undirected) graph such that for any two nodes there is a unique (undirected) path connecting them.
- Tree is connected, and has no loop.



undirected tree



directed tree

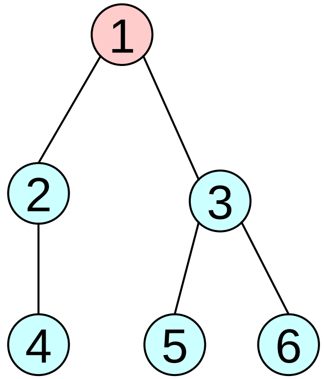
- |E| = |V|-1

# Inference with Undirected Tree

# Inference with Undirected Tree

- Propagation in a tree

  Marginalization in an undirected tree




$$p(X_1) = \frac{1}{Z} \sum_{X_2, X_3, X_4, X_5, X_6} \psi_{12}(X_1, X_2)\psi_{13}(X_1, X_3)\psi_{24}(X_2, X_4) \times \psi_{35}(X_3, X_5)\psi_{36}(X_3, X_6)$$

$$= \frac{1}{Z} \sum_{X_2} \psi_{12}(X_1, X_2) \sum_{X_3} \psi_{13}(X_1, X_3) \underbrace{\boxed{\sum_{X_4} \psi_{24}(X_2, X_4)}}_{m_{42}(X_2)} \underbrace{\boxed{\sum_{X_5} \psi_{35}(X_3, X_5)}}_{m_{53}(X_3)} \underbrace{\boxed{\sum_{X_6} \psi_{36}(X_3, X_6)}}_{m_{63}(X_3)}$$

$$= \frac{1}{Z} \sum_{X_2} \psi_{12}(X_1, X_2) \sum_{X_3} \psi_{13}(X_1, X_3) m_{42}(X_2) m_{53}(X_3) m_{63}(X_3)$$

$$= \frac{1}{Z} \underbrace{\boxed{\sum_{X_2} \psi_{12}(X_1, X_2) m_{42}(X_2)}}_{m_{21}(X_1)} \underbrace{\boxed{\sum_{X_3} \psi_{13}(X_1, X_3) m_{53}(X_3) m_{63}(X_3)}}_{m_{31}(X_1)}$$

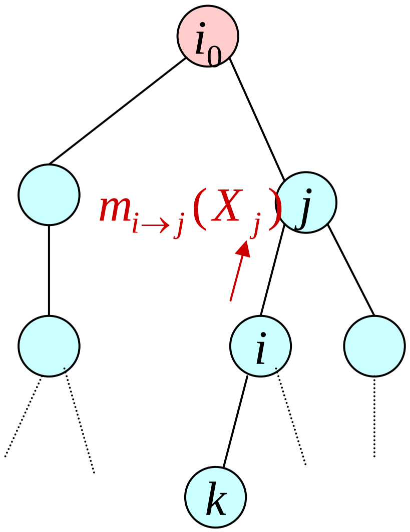
$$= \frac{1}{Z} m_{21}(X_1) m_{31}(X_1)$$

$5K^2$ operations

# Inference with Undirected Tree

Propagate messages from the bottom nodes to an upper level.

$$m_{i\to j}(X_j) = \sum_{X_i} \psi_{ji}(X_j, X_i) \prod_{k\in ne(i)\setminus\{j\}} m_{k\to i}(X_i)$$




K[2] operations

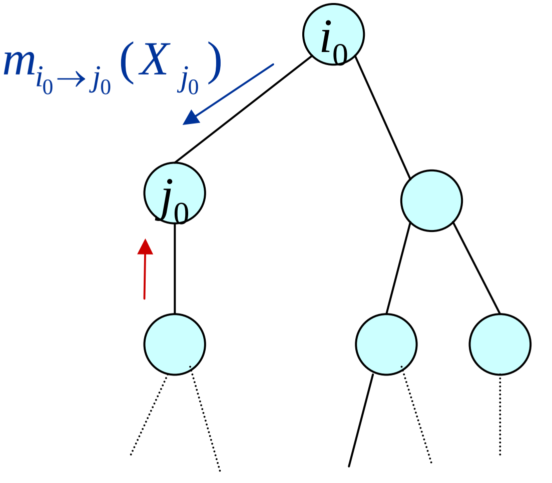
When all the messages are propagated to $i_0$,
the marginal of $X_{i_0}$ is given by

$$p(X_{i_0}) = \frac{b(X_{i_0})}{\sum_{X_{i_0}} b(X_{i_0})}, \qquad b(X_{i_0}) = \prod_{j\in ne(i_0)} m_{j\to i_0}(X_{i_0})$$

Note: normalization factor 1/Z in the joint probability is not needed.

# Inference with Undirected Tree

- Computation of all the marginals
  - We DO NOT need to repeat the process for every node.
  - Propagate the messages downward after the upward propagations are done.
  - When all the upward and downward messages are computed, every marginal can be obtained.



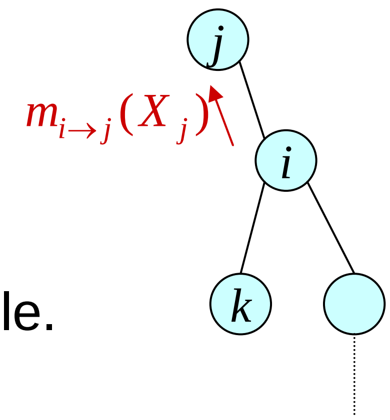
# Belief Propagation for Undirected Tree

- Belief propagation algorithm for undirected tree (sum-product algorithm)

(1) Fix a root of the tree

(2) [Upward] Propagate the messages from to bottom nodes to the root according to

$$m_{i\to j}(X_j) = \sum_{X_i} \psi_{ji}(X_j, X_i) \prod_{k \in ne(i)\setminus\{j\}} m_{k\to i}(X_i)$$




(3) [Downward] Propagate the messages from the root to the bottom nodes by the same rule.

(4) The marginals are obtained by

$$p(X_i) = \frac{b(X_i)}{\sum_{X_i} b(X_i)}, \qquad b(X_i) = \prod_{j \in ne(i)} m_{j\to i}(X_i)$$

( $b(X_i)$: belief )

# Belief Propagation for Undirected Tree

- Message passing protocol
  - The order of updates may be different, but should keep the following message passing protocol:

    "The message to a node must be propagated after the messages from all the other neighbors are received".

- Efficient algorithm
  - Reuse of messages to compute all the marginals.
  - The cost for computing all the marginals

    $= (\text{Upward} + \text{Downward}) \times K^2 = 2|E| \times K^2 = 2(|V|-1) \times K^2$

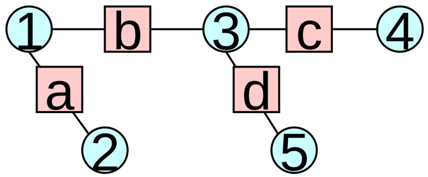
    Linear in the number of nodes or edges

- Use of evidence
  - If some nodes have evidence, just fix the values in computing the messages.

# Inference with Factor Tree

# Factor Tree

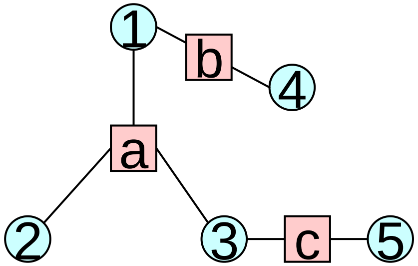
- Factor tree: tree as a bipartite graph



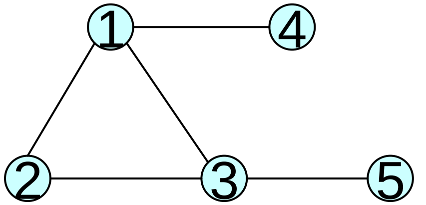

$$p(X) = \frac{1}{Z}\psi_a(X_1, X_2)\psi_b(X_1, X_3)\psi_c(X_3, X_4)$$
$$\times\psi_d(X_3, X_5)$$

- An undirected graph, which is not a tree, may be represented by a factor tree. {factor tree} $\supset$ {undirected tree}

$$p(X) = \frac{1}{Z}\psi_a(X_1, X_2, X_3)\psi_b(X_1, X_4)\psi_c(X_3, X_5)$$

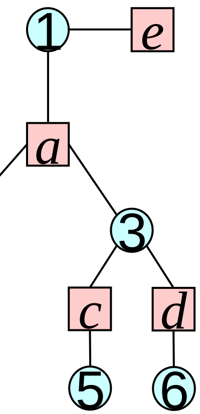



factor tree




not an undirected tree

# Marginalization for Factor tree




- Example

$$p(X) = \frac{1}{Z} f_e(X_1) f_a(X_1, X_2, X_3) f_b(X_2, X_4) f_c(X_3, X_5) f_d(X_3, X_6)$$

$$p(X_1) = \frac{1}{Z} \sum_{X_2, X_3, X_4, X_5, X_6} f_e(X_1) f_a(X_1, X_2, X_3) f_b(X_2, X_4) f_c(X_3, X_5) f_d(X_3, X_6)$$

$$= \frac{1}{Z} \sum_{X_2, X_3} f_e(X_1) f_a(X_1, X_2, X_3) \overbrace{\boxed{\sum_{X_4} f_b(X_2, X_4)}}^{m_{b\to 2}(X_2)} \overbrace{\boxed{\sum_{X_5} f_c(X_3, X_5)}}^{m_{c\to 3}(X_3)} \overbrace{\boxed{\sum_{X_6} f_d(X_3, X_6)}}^{m_{d\to 3}(X_3)}$$

$$= \frac{1}{Z} \sum_{X_2, X_3} f_e(X_1) f_a(X_1, X_2, X_3) \underbrace{\boxed{m_{b\to 2}(X_2)}}_{\mu_{2\to a}(X_2)} \underbrace{\boxed{m_{c\to 3}(X_3) m_{d\to 3}(X_3)}}_{\mu_{3\to a}(X_3)}$$

$$= \frac{1}{Z} \overbrace{\boxed{f_e(X_1)}}^{m_{e\to 1}(X_1)} \underbrace{\boxed{\sum_{X_2, X_3} f_a(X_1, X_2, X_3) \mu_{2\to a}(X_2) \mu_{3\to a}(X_3)}}_{m_{a\to 1}(X_1)}$$

$$= \frac{1}{Z} m_{e\to 1}(X_1) m_{a\to 1}(X_1)$$

# Belief Propagation for Factor Tree

Propagate messages from the bottom nodes to an upper level.

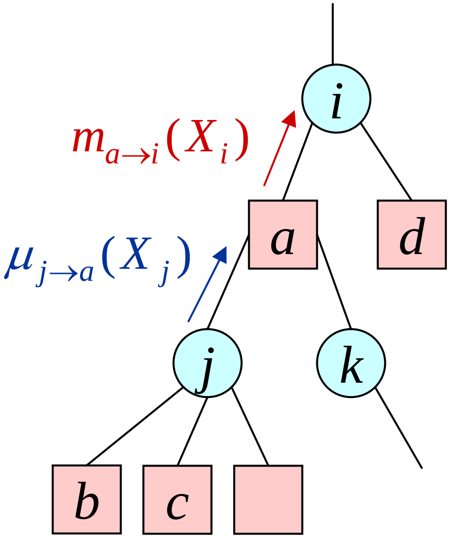
Separate the 'sum' process and 'product' process.

- Message from a factor node to a variable node:

$$m_{a\to i}(X_i) = \sum_{X_{N(a)\backslash\{i\}}} f_a(X_{N(a)}) \prod_{j\in N(a)\backslash\{i\}} \mu_{j\to a}(X_j)$$

- Message from a variable node to a factor node:

$$\mu_{j\to a}(X_j) = \prod_{b\in N(j)\backslash\{a\}} m_{b\to j}(X_j)$$




$N(i)$ : the factor nodes connected to the variable node $i$

$N(a)$ : the variable nodes connected to the factor node $a$

# Belief Propagation for Factor Tree

- Belief propagation algorithm for factor tree

(1) Fix a root variable node of the tree

(2) [Upward] Propagate the messages from bottom nodes to the root.

Factor to variable:

$$m_{a\to i}(X_i) = \sum_{X_{N(a)\backslash\{i\}}} f_a(X_{N(a)}) \prod_{j\in N(a)\backslash\{i\}} \mu_{j\to a}(X_j)$$

Variable to factor

$$\mu_{j\to a}(X_j) = \prod_{b\in N(j)\backslash\{a\}} m_{b\to j}(X_j)$$

(3) [Downward] Propagate the messages from the root to the bottom nodes by the same rules.

(4) The marginals are obtained by

$$p(X_i) \propto \prod_{a\in N(i)} m_{a\to i}(X_i)$$

# Belief Propagation for Factor Tree

- Message passing protocol
  - The order of updates may be different, but should keep the following message passing protocol:

    "The message to a node must be propagated after the messages from all the other neighbors are received".

- Efficient algorithm
  - Reuse of messages to compute all the marginals.
  - The computational cost $= 2|E| \times K^m = 2(|V|-1) \times K^m$
    $m = \max \#(\text{variables in a factor})$

    Linear in the number of nodes or edges.

- Use of evidence
  - If some nodes have evidence, just fix the values in computing the messages.

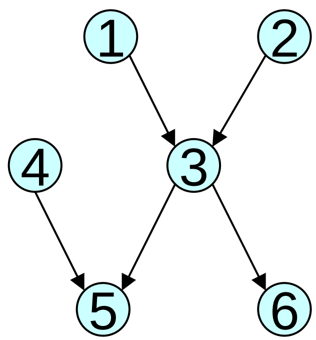
# Inference with Directed Tree

# Factor Tree Representation

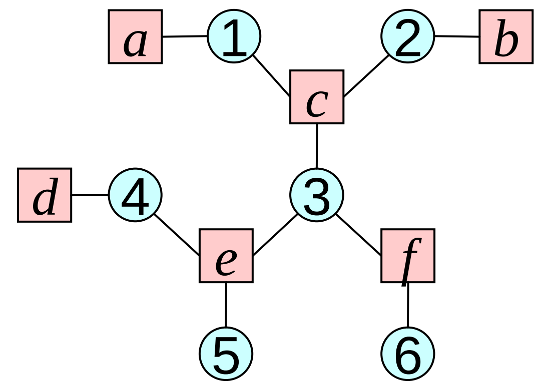
- Directed tree to factor tree
  - A directed tree (polytree) can be converted to a factor tree

    Example

$$p(X) = p(X_1)p(X_2)p(X_3 \mid X_1, X_2)p(X_4)p(X_5 \mid X_3, X_4)p(X_6 \mid X_3)$$




directed tree (polytree)




factor tree

  - Note: in the factor tree representation, each factor node is connected to a unique child node, since the factor is of the form $p(X_i \mid X_{\mathrm{pa}(i)})$.

# Belief Propagation for Directed Tree

- Belief propagation for directed tree
  - BP in the factor tree representation

    Factor nodes can be indexed by the corresponding child nodes.

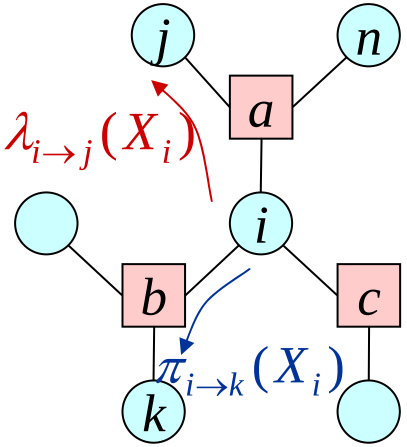
    Parent to child (← variable to factor)

$$\mu_{i\to b}(X_i) = \prod_{c\in N(i)\setminus\{b\}} m_{c\to i}(X_i) \quad \equiv \pi_{i\to k}(X_i)$$

Child to parent (← factor to variable)

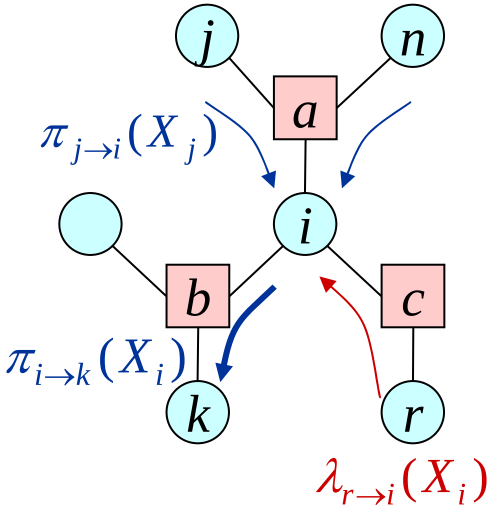
$$m_{a\to j}(X_j) = \sum_{X_{N(a)\setminus\{j\}}} f_a(X_{N(a)}) \prod_{n\in N(a)\setminus\{j\}} \mu_{n\to a}(X_n)$$

$$\equiv \lambda_{i\to j}(X_j)$$



# Belief Propagation for Directed Tree

- Parent to child




$$\pi_{i\to k}(X_i) = \prod_{c\in N(i)\setminus\{b\}} m_{c\to i}(X_i)$$

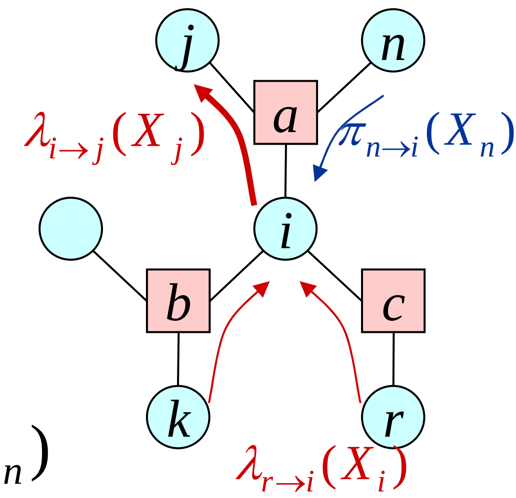
$$= \underbrace{m_{a\to i}(X_i)}_{\text{parent-side}} \underbrace{\prod_{r\in ch(i)\setminus\{k\}} \lambda_{r\to i}(X_i)}_{\text{child-side}}$$

$$= \sum_{X_{pa(i)}} p(X_i \mid X_{pa(i)}) \prod_{j\in pa(i)} \mu_{j\to a}(X_j) \prod_{r\in ch(i)\setminus\{k\}} \lambda_{r\to i}(X_i)$$

$$= \sum_{X_{pa(i)}} p(X_i \mid X_{pa(i)}) \prod_{j\in pa(i)} \pi_{j\to i}(X_j) \prod_{r\in ch(i)\setminus\{k\}} \lambda_{r\to i}(X_i)$$

# Belief Propagation for Directed Tree

- Child to parent




$$\lambda_{i\to j}(X_j)$$

$$= \sum_{X_{N(a)\backslash\{j\}}} p(X_i \mid X_{pa(i)}) \prod_{n\in N(a)\backslash\{j\}} \mu_{n\to a}(X_n)$$

$$= \sum_{X_{N(a)\backslash\{j\}}} p(X_i \mid X_{pa(i)}) \underbrace{\mu_{i\to a}(X_i)}_{\text{child-side}} \underbrace{\prod_{n\in pa(i)\backslash\{j\}} \mu_{n\to a}(X_n)}_{\text{parent-side}}$$

$$= \sum_{X_{N(a)\backslash\{j\}}} p(X_i \mid X_{pa(i)}) \prod_{b\in N(i)\backslash\{a\}} m_{b\to i}(X_i) \prod_{n\in pa(i)\backslash\{j\}} \pi_{n\to a}(X_n)$$

$$= \sum_{X_i, X_{pa(i)\backslash\{j\}}} p(X_i \mid X_{pa(i)}) \prod_{k\in ch(i)} \lambda_{k\to i}(X_i) \prod_{n\in pa(i)\backslash\{j\}} \pi_{n\to a}(X_n)$$

# Belief Propagation for Directed Tree

- πλ-algorithm (Kim & Pearl 1983)

  Parent to child:

$$\pi_{i\to k}(X_i) = \sum_{X_{pa(i)}} p(X_i \mid X_{pa(i)}) \prod_{j\in pa(i)} \pi_{j\to i}(X_j) \prod_{r\in ch(i)\backslash\{k\}} \lambda_{r\to i}(X_i)$$

  Child to parent:

$$\lambda_{i\to j}(X_j) = \sum_{X_i, X_{pa(i)\backslash\{j\}}} p(X_i \mid X_{pa(i)}) \prod_{k\in ch(i)} \lambda_{k\to i}(X_i) \prod_{n\in pa(i)\backslash\{j\}} \pi_{n\to a}(X_n)$$

  Marginal:

$$p(X_i) \propto \lambda(X_i)\pi(X_i)$$

$$\lambda(X_i) = \prod_{k\in ch(i)} \lambda_{k\to i}(X_i), \quad \pi(X_i) = \sum_{X_{pa(i)}} p(X_i \mid X_{pa(i)}) \prod_{j\in pa(i)} \pi_{j\to i}(X_j)$$

  - πλ-algorithm is the first general belief propagation algorithm.

# Mini Summary

- Belief propagation / sum-product algorithm
  - All the marginals are exactly calculated for trees.
    - Undirected tree, factor tree, polytree.
    - Non-tree cases will be discussed later.
  - The computational cost is linear w.r.t. the tree size (number of variables).
    - Basic idea is successive marginal-out, but the messages are reused to compute all the marginals.
    - Messages are passed upward and then downward.
    - In general, the order of the message passing should keep the message passing protocol.
  - The equations of message passing is local:

    product of the messages from the neighbors and sum over local variables.

# Mini Summary

- Constant factor is not necessary.

  To given the joint probability density, the form

  $$p(X) \propto \prod f_a(X_a)$$

  is sufficient to apply the belief propagation.

  Just normalize after the unnormalized marginal is computed.

- Normalization factor can be computed by belief propagation.

  For

  $$p(X) \propto \prod f_a(X_a)$$

  Normalization factor $Z$ is given by marginal-out:

  $$Z = \sum_X \prod f_a(X_a)$$