Lecture 12: Factor graphs and Sum-Product algorithm

Course instructor: Alexey Frolov

al.frolov@skoltech.ru

Teaching Assistant: Stanislav Kruglik

stanislav.kruglik@skolkovotech.ru

March 2, 2017

Distributive Law

$$ab + ac = a(b + c)$$

Distributive Law

$$ab + ac = a(b + c)$$

$$\textstyle\sum_{i,j} a_i b_j \qquad\qquad (\sum_i a_i)(\sum_j b_j)$$

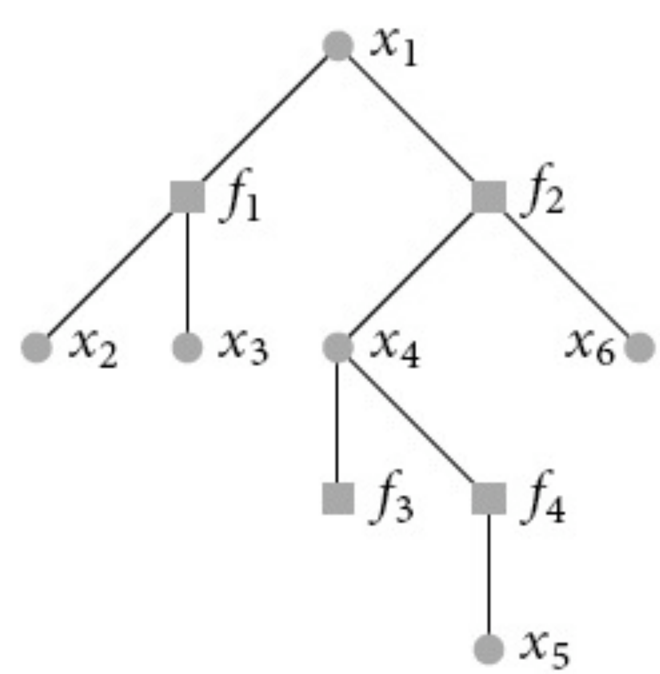
Example

Example

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = f_1(x_1, x_2, x_3) f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5)$$

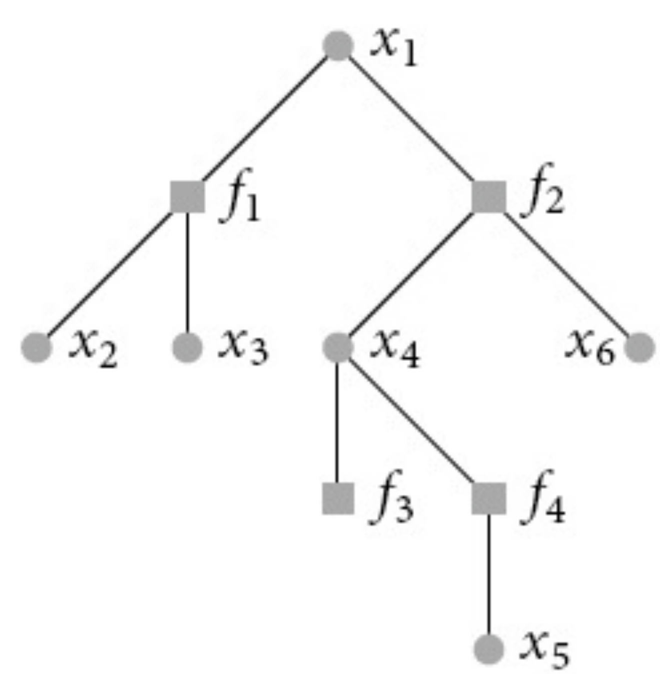
Example

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = f_1(x_1, x_2, x_3) f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5)$$



Example

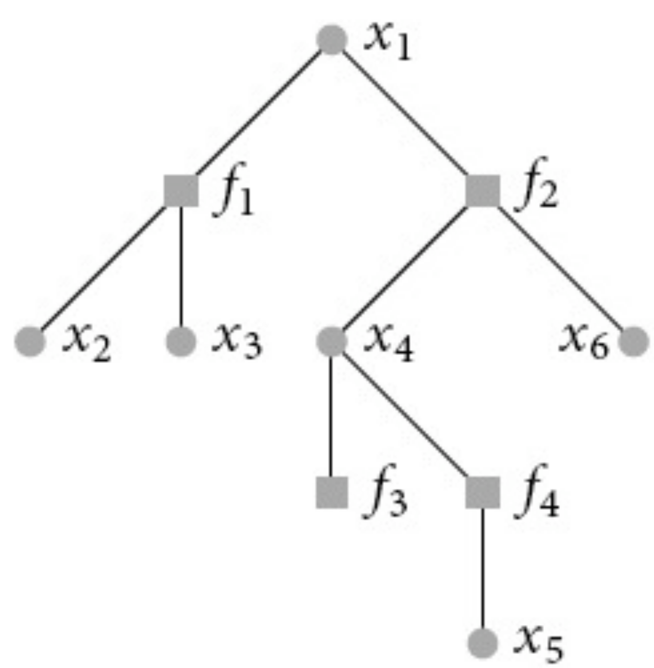
$$f(x_1, x_2, x_3, x_4, x_5, x_6) = f_1(x_1, x_2, x_3) f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5)$$



$$f(x_1) = \sum_{x_2, x_3, x_4, x_5, x_6} f(x_1, x_2, x_3, x_4, x_5, x_6) = \sum_{\sim x_1} f(x_1, x_2, x_3, x_4, x_5, x_6)$$

Example

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = f_1(x_1, x_2, x_3) f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5)$$

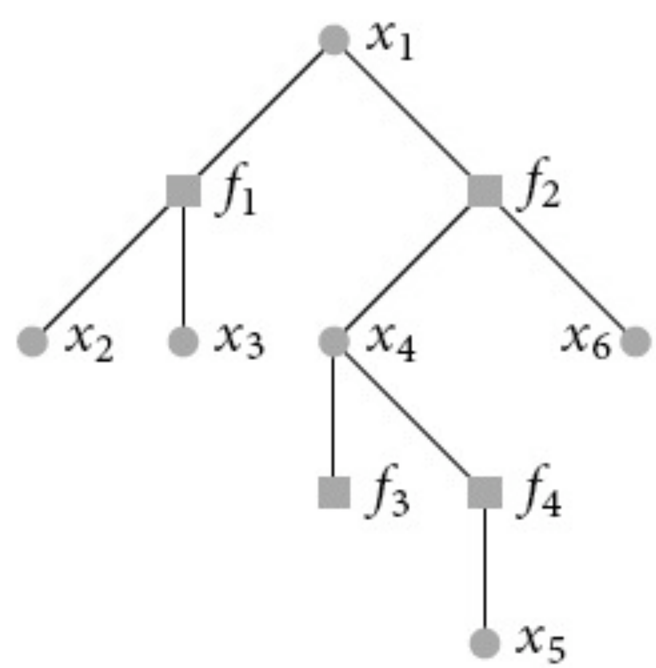


$$f(x_1) = \sum_{x_2, x_3, x_4, x_5, x_6} f(x_1, x_2, x_3, x_4, x_5, x_6) = \sum_{\sim x_1} f(x_1, x_2, x_3, x_4, x_5, x_6)$$

Note: $f(x_1)$ is a function; therefore, it takes on a distinct value for each value of x_1

Example

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = f_1(x_1, x_2, x_3) f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5)$$



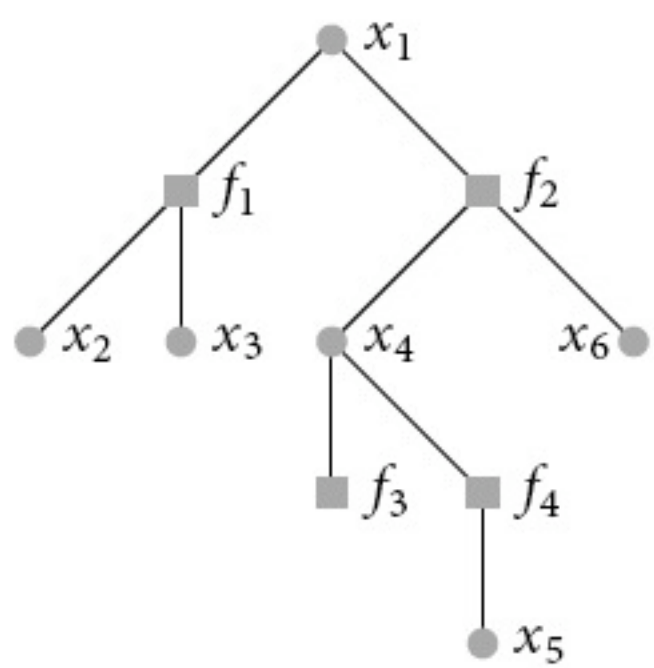
$$f(x_1) = \sum_{x_2, x_3, x_4, x_5, x_6} f(x_1, x_2, x_3, x_4, x_5, x_6) = \sum_{\sim x_1} f(x_1, x_2, x_3, x_4, x_5, x_6)$$

Note: $f(x_1)$ is a function; therefore, it takes on a distinct value for each value of x_1

$|\mathcal{X}|$ alphabet

Example

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = f_1(x_1, x_2, x_3) f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5)$$



$$f(x_1) = \sum_{x_2, x_3, x_4, x_5, x_6} f(x_1, x_2, x_3, x_4, x_5, x_6) = \sum_{\sim x_1} f(x_1, x_2, x_3, x_4, x_5, x_6)$$

Note: $f(x_1)$ is a function; therefore, it takes on a distinct value for each value of x_1

$|\mathcal{X}|$ alphabet

$\Theta\left(|\mathcal{X}|^6\right)$ brute force complexity

Example

$$f(x_1) = \Big[\sum_{x_2,x_3} f_1(x_1, x_2, x_3)\Big]\Big[\sum_{x_4} f_3(x_4)\Big(\sum_{x_6} f_2(x_1, x_4, x_6)\Big)\Big(\sum_{x_5} f_4(x_4, x_5)\Big)\Big]$$

Example

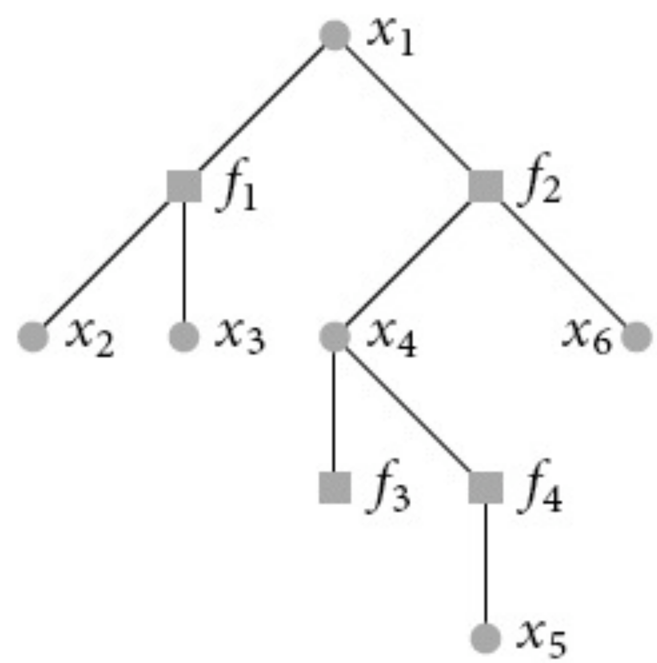
$$f(x_1) = \Big[\sum_{x_2, x_3} f_1(x_1, x_2, x_3)\Big]\Big[\sum_{x_4} f_3(x_4)\Big(\sum_{x_6} f_2(x_1, x_4, x_6)\Big)\Big(\sum_{x_5} f_4(x_4, x_5)\Big)\Big]$$

$\Theta\left(|\mathcal{X}|^3\right)$ complexity

Example

$$f(x_1) = \Big[\sum_{x_2,x_3} f_1(x_1,x_2,x_3)\Big]\Big[\sum_{x_4} f_3(x_4)\Big(\sum_{x_6} f_2(x_1,x_4,x_6)\Big)\Big(\sum_{x_5} f_4(x_4,x_5)\Big)\Big]$$

$\Theta\left(|\mathcal{X}|^3\right)$ complexity



Does there exist a systematic way to find this low complexity scheme using the structure of the graph?

Marginalization via Message Passing for Trees

Marginalization via Message Passing for Trees

$g(z) = \sum_{\sim z} g(z, \ldots)$

Marginalization via Message Passing for Trees

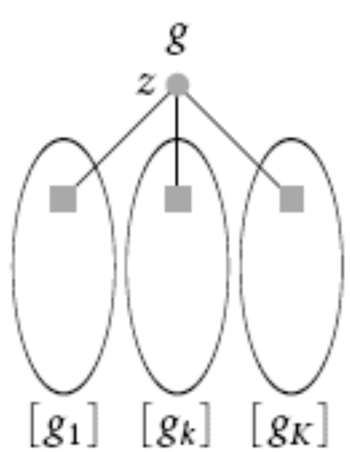
$g(z) = \sum_{\sim z} g(z, \ldots)$

$$g(z, \ldots) = \prod_{k=1}^{K} [g_k(z, \ldots)]$$

Marginalization via Message Passing for Trees

$$g(z) = \sum_{\sim z} g(z, \ldots)$$

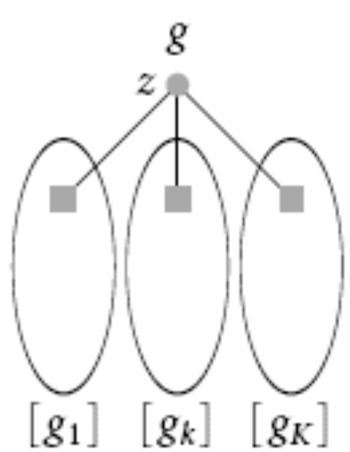
$$g(z, \ldots) = \prod_{k=1}^{K} [g_k(z, \ldots)]$$



Marginalization via Message Passing for Trees

$g(z) = \sum_{\sim z} g(z, \ldots)$

$$g(z, \ldots) = \prod_{k=1}^{K} [g_k(z, \ldots)]$$

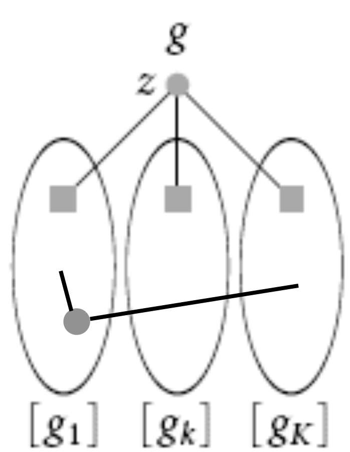


Note: the individual functions $g_k(z, \ldots)$
only share the variable z; all other
variables are "private"

Marginalization via Message Passing for Trees

$g(z) = \sum_{\sim z} g(z, \dots)$

$$g(z, \dots) = \prod_{k=1}^{K} [g_k(z, \dots)]$$



Note: the individual functions $g_k(z, \dots)$
only share the variable z; all other
variables are "private"

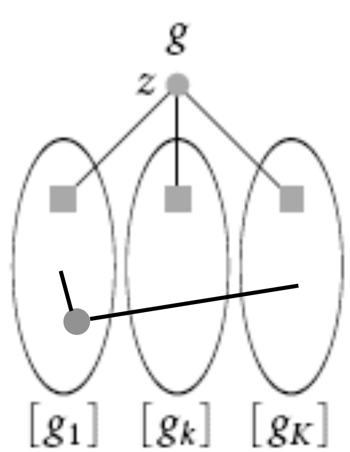
otherwise ... graph is not a tree

Marginalization via Message Passing for Trees

$$g(z) = \sum_{\sim z} g(z, \dots)$$

$$f(x_1) = \sum_{\sim x_1} f(x_1, x_2, x_3, x_4, x_5, x_6)$$

$$g(z, \dots) = \prod_{k=1}^{K} [g_k(z, \dots)]$$



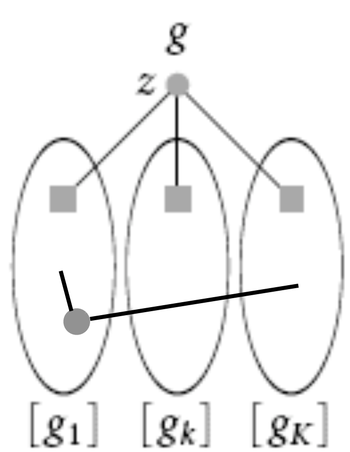
Note: the individual functions $g_k(z, \dots)$ only share the variable z; all other variables are "private"

otherwise ... graph is not a tree

Marginalization via Message Passing for Trees

$$g(z) = \sum_{\sim z} g(z, \dots)$$

$$g(z, \dots) = \prod_{k=1}^{K} [g_k(z, \dots)]$$



Note: the individual functions $g_k(z, \dots)$ only share the variable z; all other variables are "private"

otherwise ... graph is not a tree

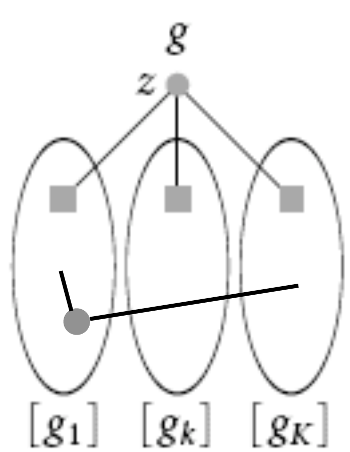
$$f(x_1) = \sum_{\sim x_1} f(x_1, x_2, x_3, x_4, x_5, x_6)$$

$$f(x_1, \dots) = [f_1(x_1, x_2, x_3)]\,[f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5)]$$

Marginalization via Message Passing for Trees

$$g(z) = \sum_{\sim z} g(z, \dots)$$

$$g(z, \dots) = \prod_{k=1}^{K} [g_k(z, \dots)]$$

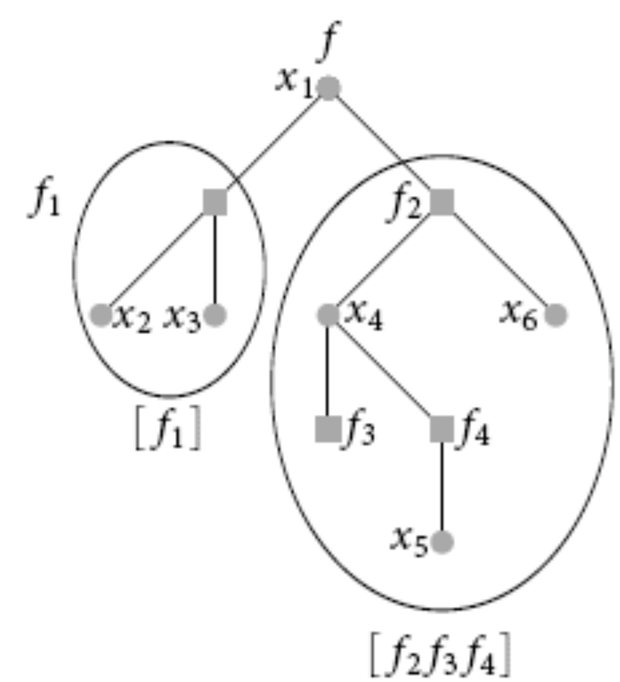


Note: the individual functions $g_k(z, \dots)$
only share the variable z; all other
variables are "private"

otherwise ... graph is not a tree

$$f(x_1) = \sum_{\sim x_1} f(x_1, x_2, x_3, x_4, x_5, x_6)$$

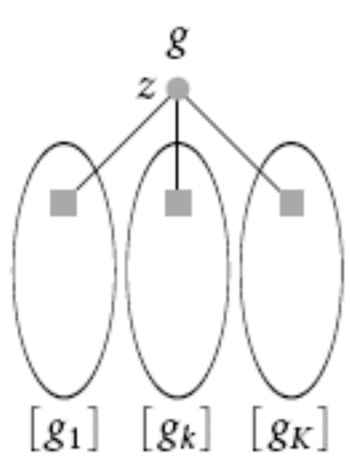
$$f(x_1, \dots) = [f_1(x_1, x_2, x_3)]\,[f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5)]$$



Marginalization via Message Passing for Trees

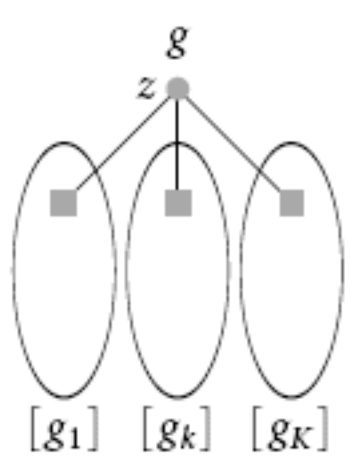
Marginalization via Message Passing for Trees

$$\sum_{\sim z} g(z, \ldots) = \underbrace{\sum_{\sim z} \prod_{k=1}^{K} [g_k(z, \ldots)]}_{\text{marginal of product}} = \underbrace{\prod_{k=1}^{K} \Big[\sum_{\sim z} g_k(z, \ldots) \Big]}_{\text{product of marginals}}$$



Marginalization via Message Passing for Trees

$$\sum_{\sim z} g(z, \dots) = \underbrace{\sum_{\sim z} \prod_{k=1}^{K} [g_k(z, \dots)]}_{\text{marginal of product}} = \underbrace{\prod_{k=1}^{K} \Big[\sum_{\sim z} g_k(z, \dots) \Big]}_{\text{product of marginals}}$$

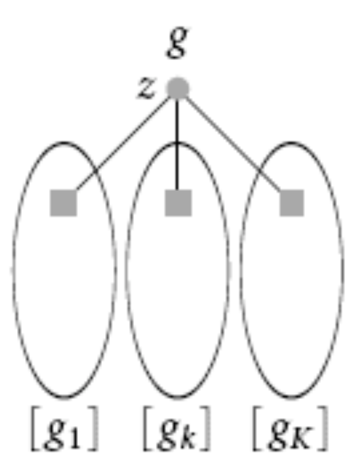


marginal $\sum_{\sim z} g(z, \dots)$ is the product of the individual marginals

Marginalization via Message Passing for Trees

recall that g(z) is a function, taking a distinct value for each value of z

$$\sum_{\sim z} g(z,\ldots) = \underbrace{\sum_{\sim z} \prod_{k=1}^{K} [g_k(z,\ldots)]}_{\text{marginal of product}} = \underbrace{\prod_{k=1}^{K} \Big[\sum_{\sim z} g_k(z,\ldots)\Big]}_{\text{product of marginals}}$$



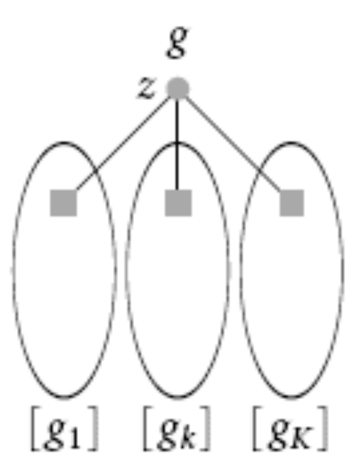
marginal $\sum_{\sim z} g(z,\ldots)$ is the product of the individual marginals

Marginalization via Message Passing for Trees

recall that g(z) is a function, taking a distinct value for each value of z

instead of computing g(z) directly by brute force we can first compute each of the functions $g_k(z)$; we then get g(z) by multiplying these functions $g_k(z)$

$$\sum_{\sim z} g(z,\dots) = \underbrace{\sum_{\sim z} \prod_{k=1}^{K} [g_k(z,\dots)]}_{\text{marginal of product}} = \underbrace{\prod_{k=1}^{K} \Big[\sum_{\sim z} g_k(z,\dots)\Big]}_{\text{product of marginals}}$$



marginal $\sum_{\sim z} g(z,\dots)$ is the product of the individual marginals

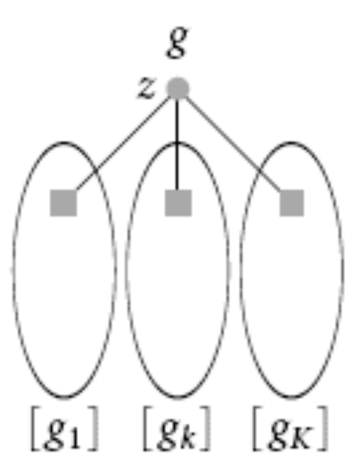
Marginalization via Message Passing for Trees

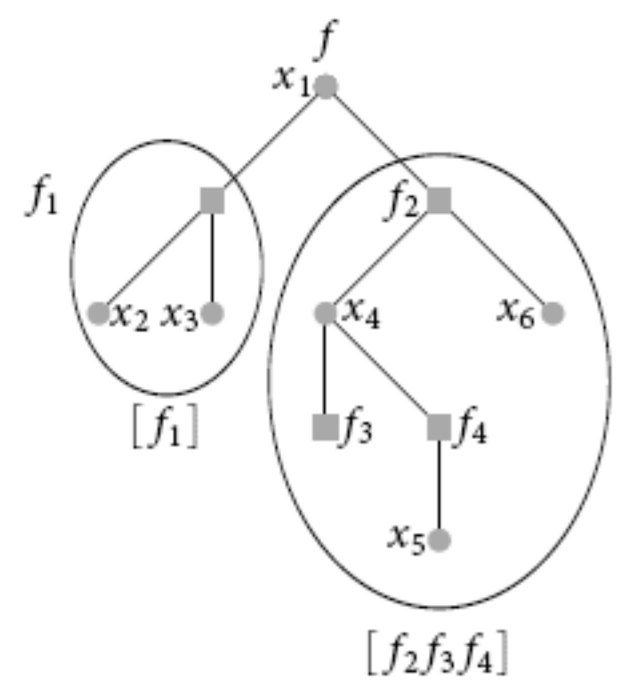
recall that g(z) is a function, taking a distinct value for each value of z

instead of computing g(z) directly by brute force we can first compute each of the functions $g_k(z)$; we then get g(z) by multiplying these functions $g_k(z)$

$$\sum_{\sim z} g(z,\ldots) = \underbrace{\sum_{\sim z} \prod_{k=1}^{K} [g_k(z,\ldots)]}_{\text{marginal of product}} = \underbrace{\prod_{k=1}^{K} \Big[\sum_{\sim z} g_k(z,\ldots)\Big]}_{\text{product of marginals}}$$

$$f(x_1) = \Big[\sum_{\sim x_1} f_1(x_1,x_2,x_3)\Big]\Big[\sum_{\sim x_1} f_2(x_1,x_4,x_6) f_3(x_4) f_4(x_4,x_5)\Big]$$





marginal $\sum_{\sim z} g(z,\ldots)$ is the product of the individual marginals

Marginalization via Message Passing for Trees

$$\sum_{\sim z} g(z, \dots) = \underbrace{\sum_{\sim z} \prod_{k=1}^{K} [g_k(z, \dots)]}_{\text{marginal of product}} = \underbrace{\prod_{k=1}^{K} \Big[\sum_{\sim z} g_k(z, \dots)\Big]}_{\text{product of marginals}}$$

Marginalization via Message Passing for Trees

$$\sum_{\sim z} g(z, \ldots) = \underbrace{\sum_{\sim z} \prod_{k=1}^{K} [g_k(z, \ldots)]}_{\text{marginal of product}} = \underbrace{\prod_{k=1}^{K} \Big[\sum_{\sim z} g_k(z, \ldots) \Big]}_{\text{product of marginals}}$$

$g_k(z, \ldots)$

Marginalization via Message Passing for Trees

$$\sum_{\sim z} g(z,\ldots) = \underbrace{\sum_{\sim z} \prod_{k=1}^{K} [g_k(z,\ldots)]}_{\text{marginal of product}} = \underbrace{\prod_{k=1}^{K} \Big[\sum_{\sim z} g_k(z,\ldots)\Big]}_{\text{product of marginals}}$$

$g_k(z,\ldots)$

$$g_k(z,\ldots) = \underbrace{h(z, z_1, \ldots, z_J)}_{\text{kernel}} \prod_{j=1}^{J} \underbrace{[h_j(z_j,\ldots)]}_{\text{factors}}$$

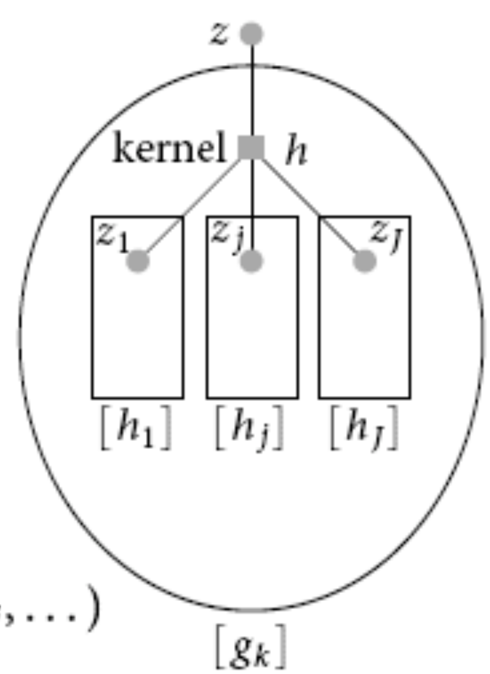
"kernel" $h(z, z_1, \ldots, z_J)$

Marginalization via Message Passing for Trees

$$g_k(z,\dots) = \underbrace{h(z,z_1,\dots,z_J)}_{\text{kernel}} \prod_{j=1}^{J} \underbrace{\left[h_j(z_j,\dots)\right]}_{\text{factors}}$$

"kernel" $h(z, z_1, \dots, z_J)$

$g_k(z, \dots)$



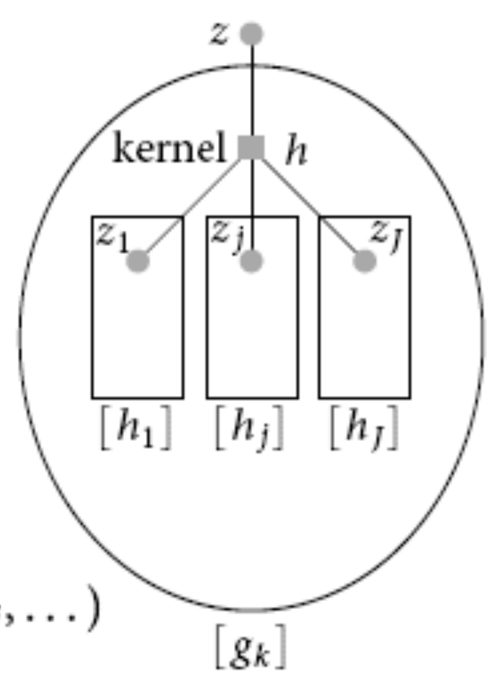
factors $h_j(z_j, \dots)$

Marginalization via Message Passing for Trees

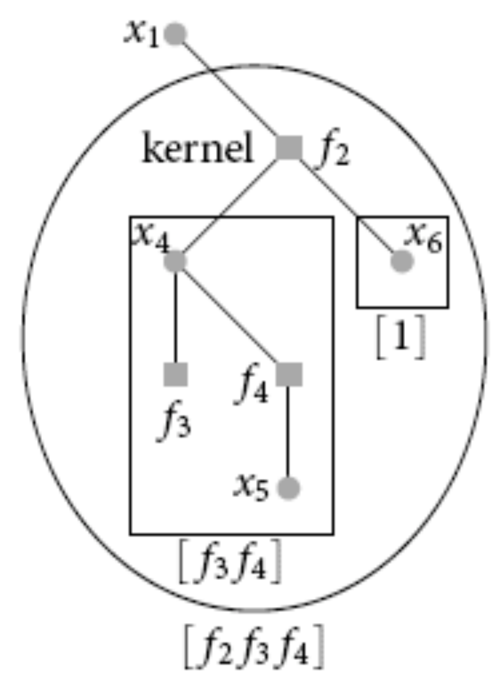
$$g_k(z, \dots) = \underbrace{h(z, z_1, \dots, z_J)}_{\text{kernel}} \prod_{j=1}^{J} \underbrace{[h_j(z_j, \dots)]}_{\text{factors}}$$

"kernel" $h(z, z_1, \dots, z_J)$

$g_k(z, \dots)$



factors $h_j(z_j, \dots)$



$$f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5) = \underbrace{f_2(x_1, x_4, x_6)}_{\text{kernel}} \underbrace{[f_3(x_4) f_4(x_4, x_5)]}_{x_4} \underbrace{[1]}_{x_6}$$

Marginalization via Message Passing for Trees

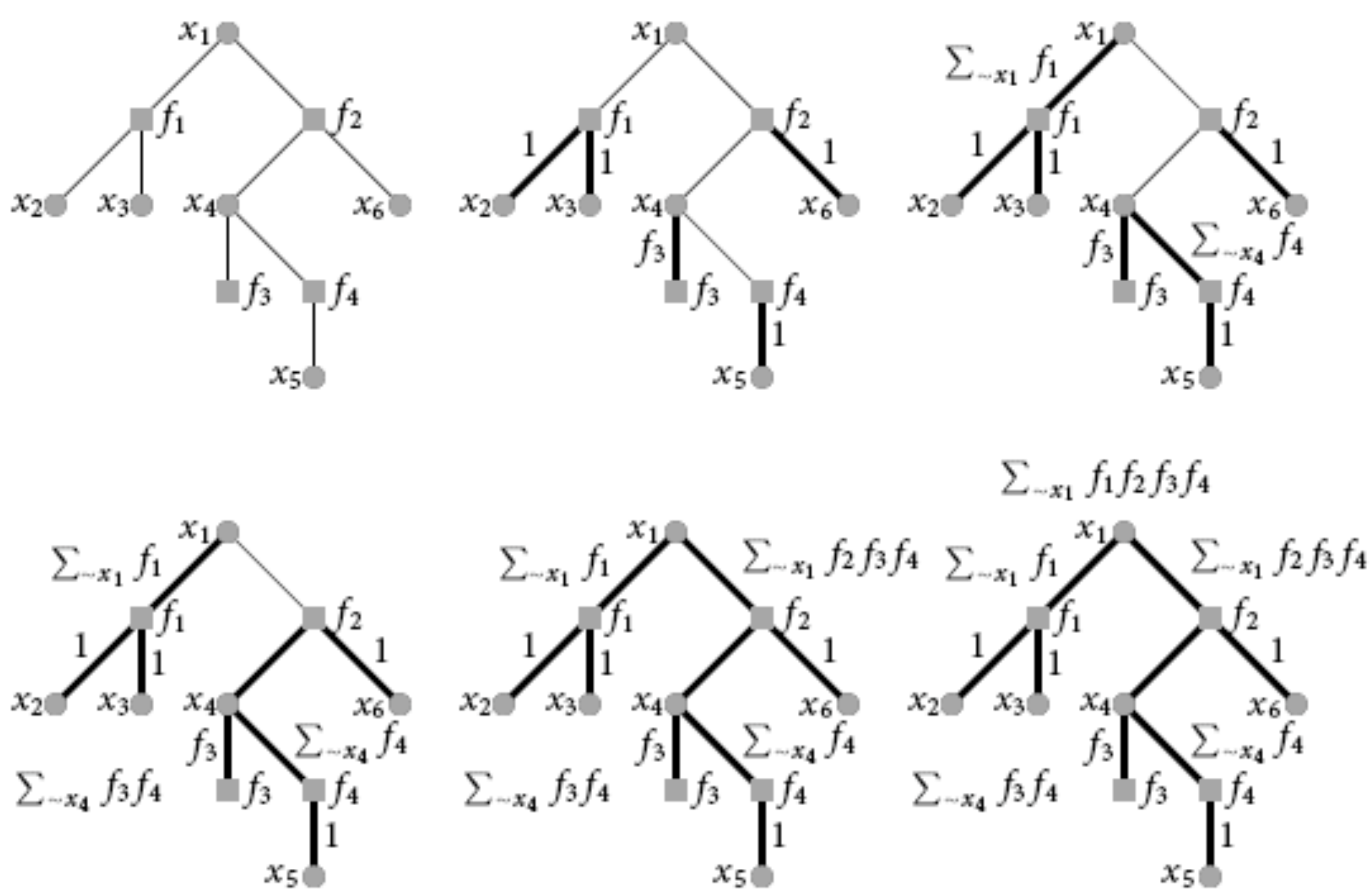
$$
\begin{aligned}
\sum_{\sim z} g_k(z, \dots) &= \sum_{\sim z} h(z, z_1, \dots, z_J) \prod_{j=1}^{J} \left[h_j(z_j, \dots) \right] \\
&= \sum_{\sim z} h(z, z_1, \dots, z_J) \underbrace{\prod_{j=1}^{J} \Big[\sum_{\sim z_j} h_j(z_j, \dots) \Big]}_{\text{product of marginals}}
\end{aligned}
$$

Marginalization via Message Passing for Trees

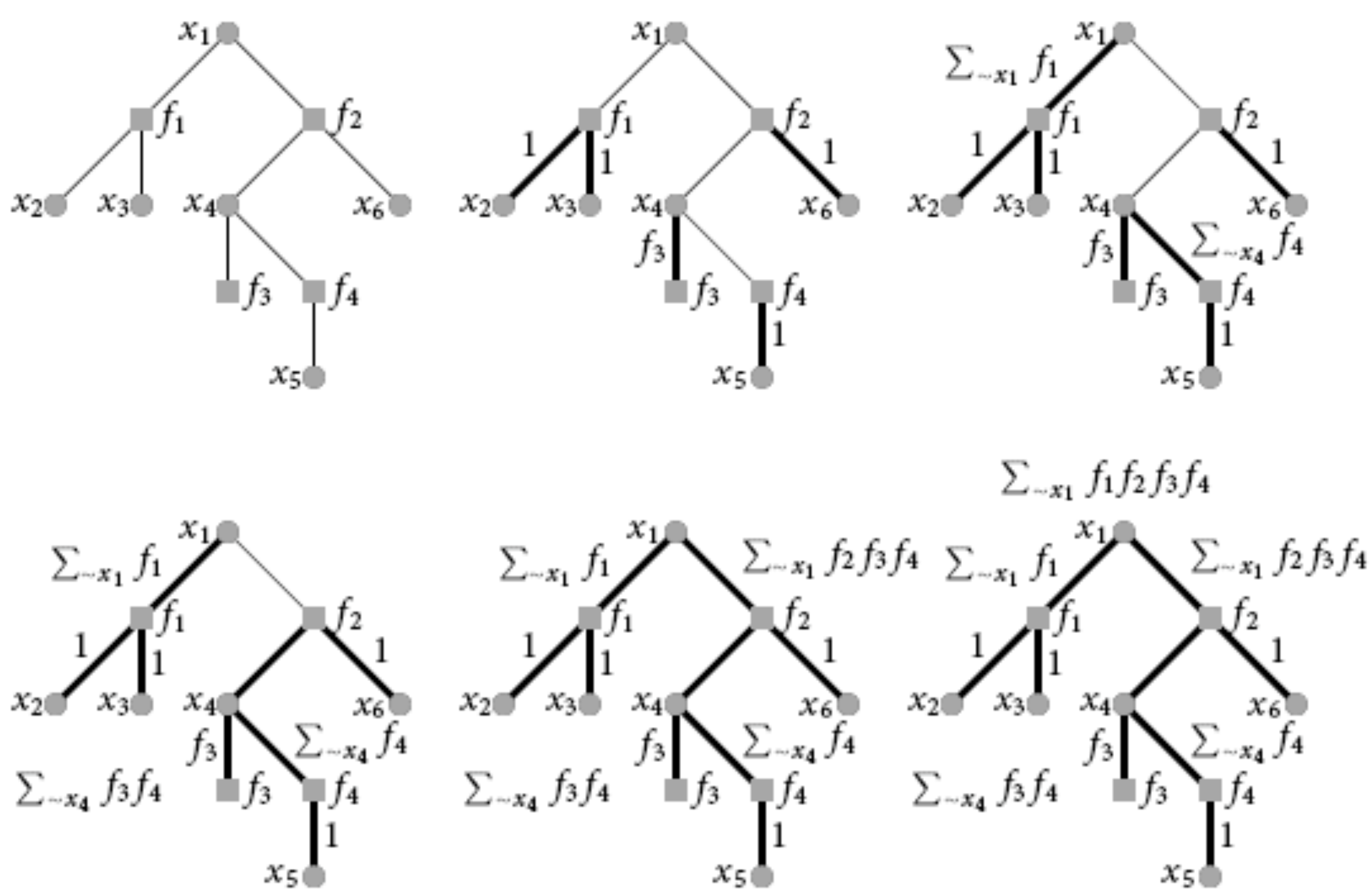
$$\sum_{\sim z} g_k(z, \ldots) = \sum_{\sim z} h(z, z_1, \ldots, z_J) \prod_{j=1}^{J} \left[h_j(z_j, \ldots) \right]$$
$$= \sum_{\sim z} h(z, z_1, \ldots, z_J) \underbrace{\prod_{j=1}^{J} \Big[\sum_{\sim z_j} h_j(z_j, \ldots) \Big]}_{\text{product of marginals}}$$

$$f(x_1) = \Big[\sum_{\sim x_1} f_1(x_1, x_2, x_3) \Big] \Big[\sum_{\sim x_1} f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5) \Big]$$

Marginalization via Message Passing for Trees

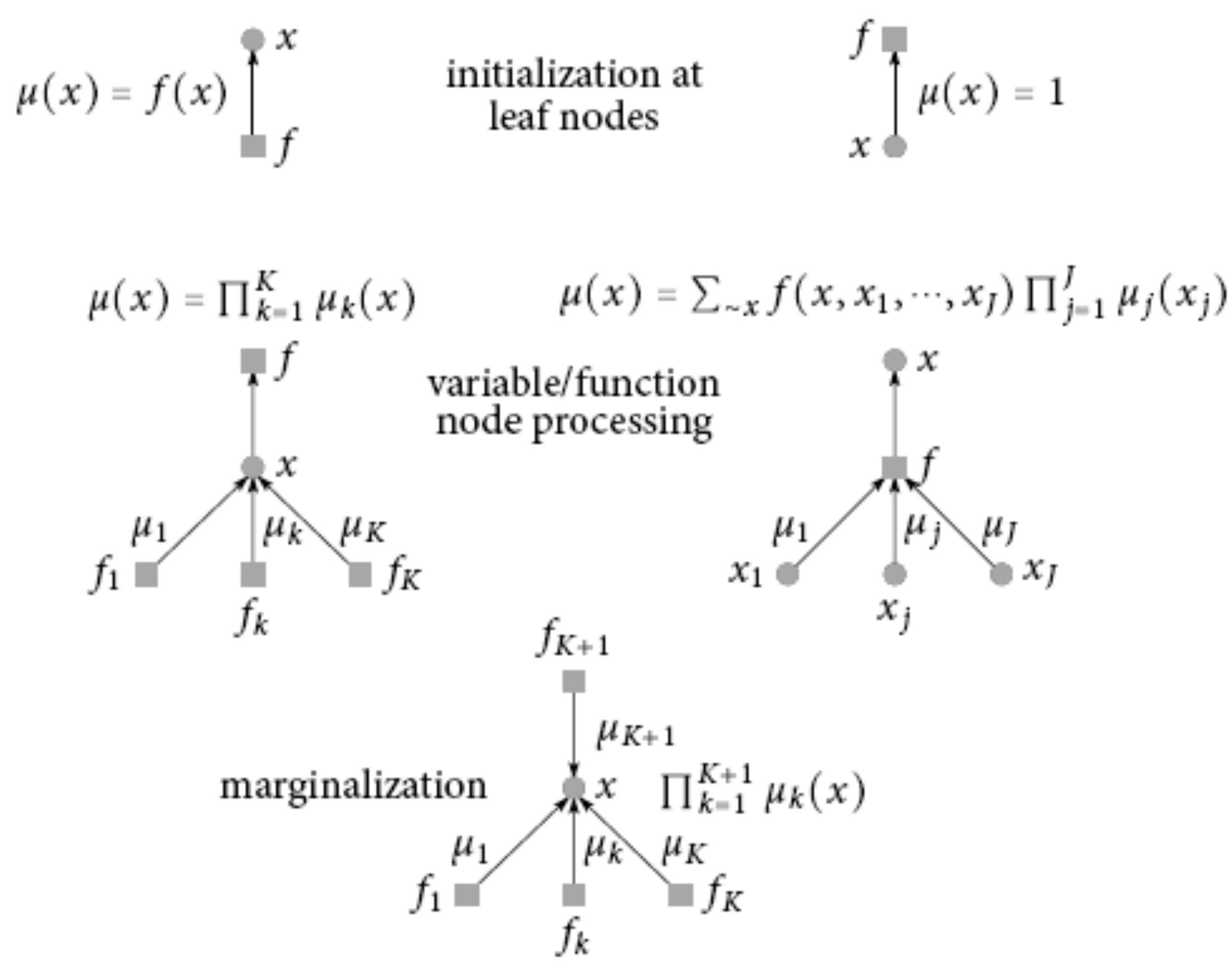


Marginalization via Message Passing for Trees



complexity proportional to highest degree

Message Passing Rules



Example

Example

$$
\begin{array}{cc}
 & \begin{array}{ccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{array} \\
H = & \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}
\end{array}
$$

Example

$$H = \begin{array}{c} \begin{array}{ccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

$$f(x_1, \ldots, x_7) = \mathbb{1}_{\{x \in C(H)\}} = \begin{cases} 1, & \text{if } Hx^T = 0^T, \\ 0, & \text{otherwise.} \end{cases}$$

Example

$$H = \begin{array}{c} \\ \\ \end{array}\begin{array}{ccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{array}$$

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

$$f(x_1, \ldots, x_7) = \mathbb{1}_{\{x \in C(H)\}} = \begin{cases} 1, & \text{if } Hx^T = 0^T, \\ 0, & \text{otherwise.} \end{cases}$$

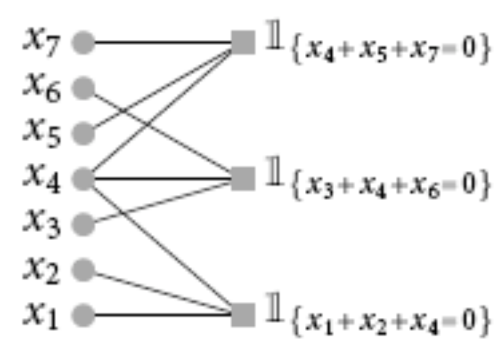
$$f(x_1, \ldots, x_7) = \mathbb{1}_{\{x_1+x_2+x_4=0\}} \mathbb{1}_{\{x_3+x_4+x_6=0\}} \mathbb{1}_{\{x_4+x_5+x_7=0\}}$$

Example

$$
H = \begin{array}{c} \begin{array}{ccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \end{array}
$$

$$
f(x_1, \ldots, x_7) = \mathbb{1}_{\{x \in C(H)\}} = \begin{cases} 1, & \text{if } Hx^T = 0^T, \\ 0, & \text{otherwise.} \end{cases}
$$

$$
f(x_1, \ldots, x_7) = \mathbb{1}_{\{x_1+x_2+x_4=0\}} \mathbb{1}_{\{x_3+x_4+x_6=0\}} \mathbb{1}_{\{x_4+x_5+x_7=0\}}
$$

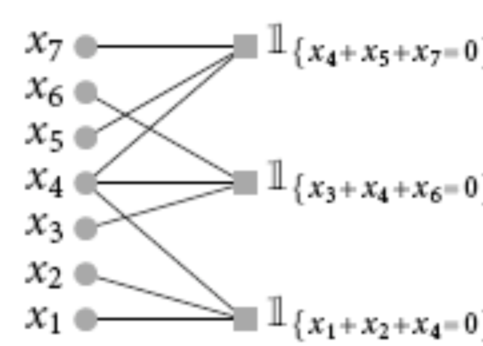


Example

$$H = \begin{array}{c} \\ \\ \end{array}\begin{array}{c} \begin{array}{ccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

$$f(x_1, \dots, x_7) = \mathbb{1}_{\{x \in C(H)\}} = \begin{cases} 1, & \text{if } Hx^T = 0^T, \\ 0, & \text{otherwise.} \end{cases}$$

$$f(x_1, \dots, x_7) = \mathbb{1}_{\{x_1+x_2+x_4=0\}} \mathbb{1}_{\{x_3+x_4+x_6=0\}} \mathbb{1}_{\{x_4+x_5+x_7=0\}}$$



again a tree

Bitwise MAP Decoding

$$
\begin{aligned}
& & \hat{x}_i^{\text{MAP}}(y) &= \operatorname{argmax}_{x_i \in \{\pm 1\}} p_{X_i \mid Y}(x_i \mid y) \\
&\text{(law of total probability)} & &= \operatorname{argmax}_{x_i \in \{\pm 1\}} \sum_{\sim x_i} p_{X \mid Y}(x \mid y) \\
&\text{(Bayes')} & &= \operatorname{argmax}_{x_i \in \{\pm 1\}} \sum_{\sim x_i} p_{Y \mid X}(y \mid x) p_X(x) \\
& & &= \operatorname{argmax}_{x_i \in \{\pm 1\}} \sum_{\sim x_i} \Big(\prod_j p_{Y_j \mid X_j}(y_j \mid x_j) \Big) \mathbb{1}_{\{x \in C\}}
\end{aligned}
$$

Bitwise MAP Decoding

$$\hat{x}_i^{\mathrm{MAP}}(y) = \mathrm{argmax}_{x_i \in \{\pm 1\}} p_{X_i \mid Y}(x_i \mid y)$$

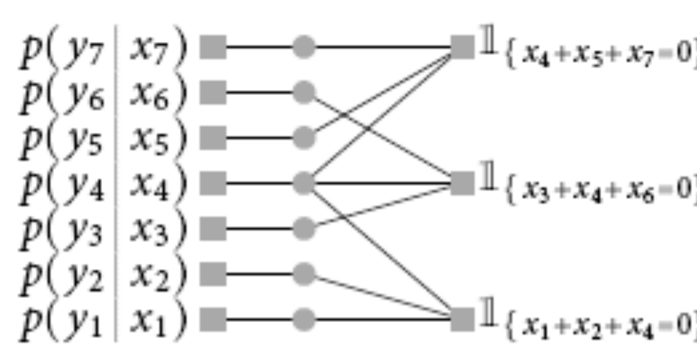
(law of total probability) $$= \mathrm{argmax}_{x_i \in \{\pm 1\}} \sum_{\sim x_i} p_{X \mid Y}(x \mid y)$$

(Bayes') $$= \mathrm{argmax}_{x_i \in \{\pm 1\}} \sum_{\sim x_i} p_{Y \mid X}(y \mid x) p_X(x)$$

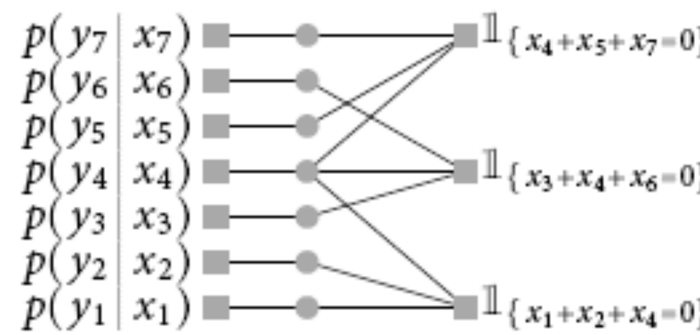
$$= \mathrm{argmax}_{x_i \in \{\pm 1\}} \sum_{\sim x_i} \Big(\prod_j p_{Y_j \mid X_j}(y_j \mid x_j) \Big) \mathbb{1}_{\{x \in C\}}$$

$$\mathrm{argmax}_{x_i \in \{\pm 1\}} \sum_{\sim x_i} \Big(\prod_{i=1}^{7} p_{Y_i \mid X_j}(y_j \mid x_j) \Big) \mathbb{1}_{\{x_1+x_2+x_4=0\}} \mathbb{1}_{\{x_3+x_4+x_6=0\}} \mathbb{1}_{\{x_4+x_5+x_7=0\}}$$

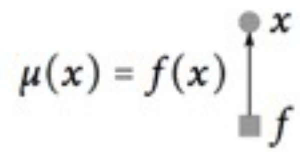
$$H = \begin{array}{c} \begin{array}{ccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \end{array}$$



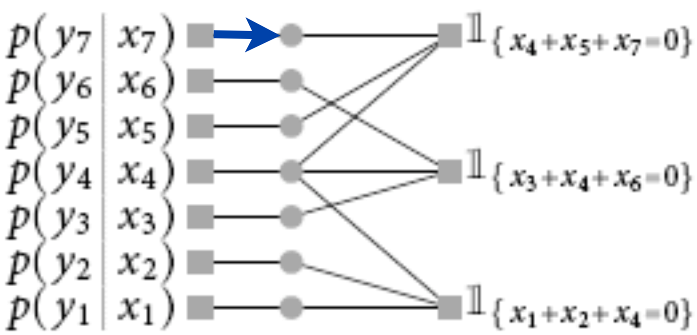
Decoding for Trees via Message Passing



Decoding for Trees via Message Passing

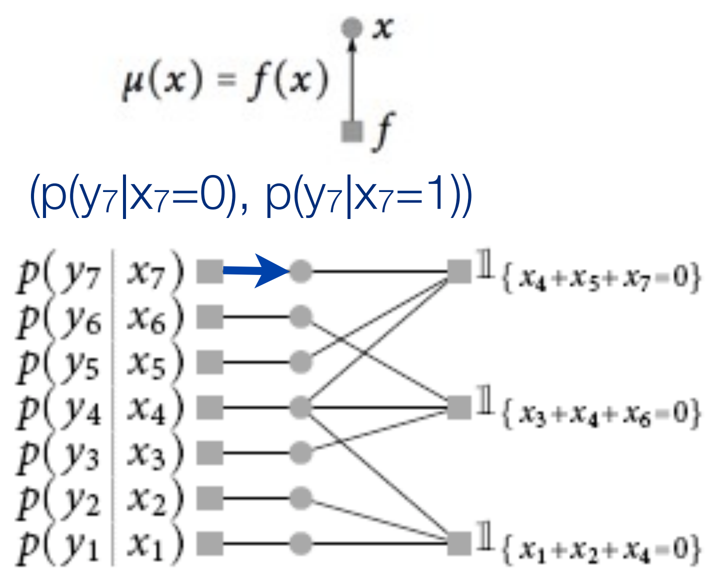


Initial messages from leaf check nodes on the left:



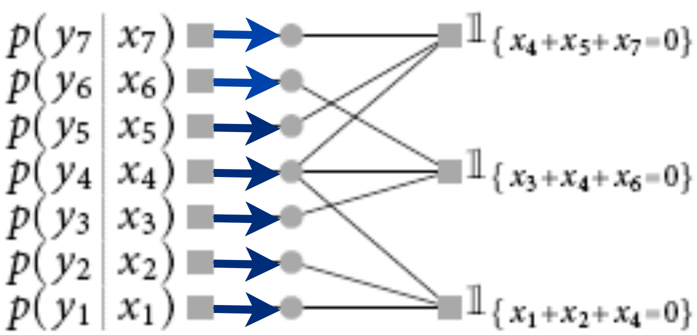
Decoding for Trees via Message Passing

Initial messages from leaf check nodes on the left:



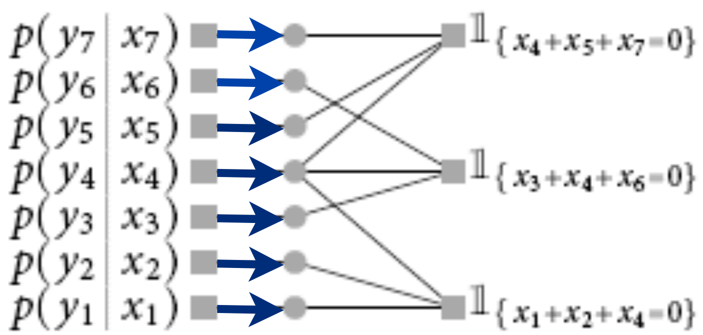
Decoding for Trees via Message Passing

same for all other leafs

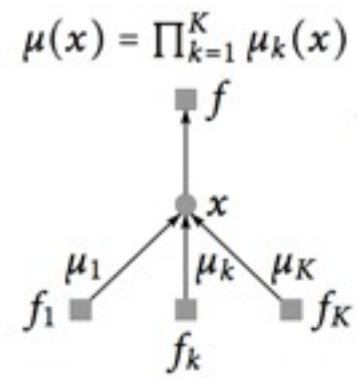


Decoding for Trees via Message Passing

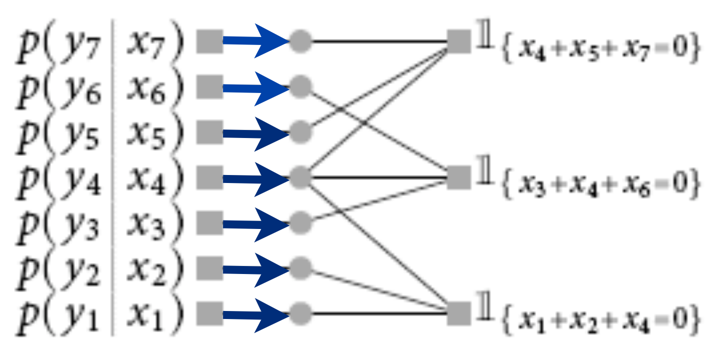
now use message passing rules



Decoding for Trees via Message Passing

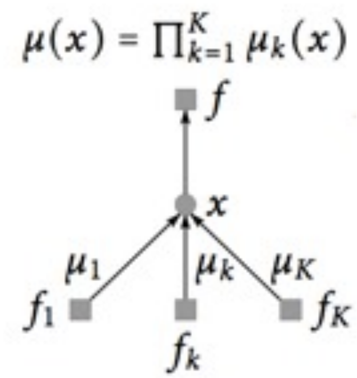


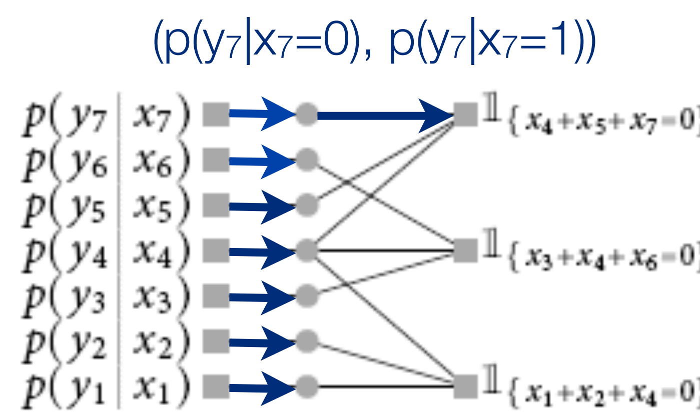
at variables we multiply messages



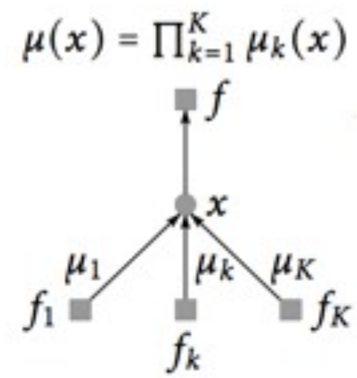
Decoding for Trees via Message Passing

at variables we multiply messages

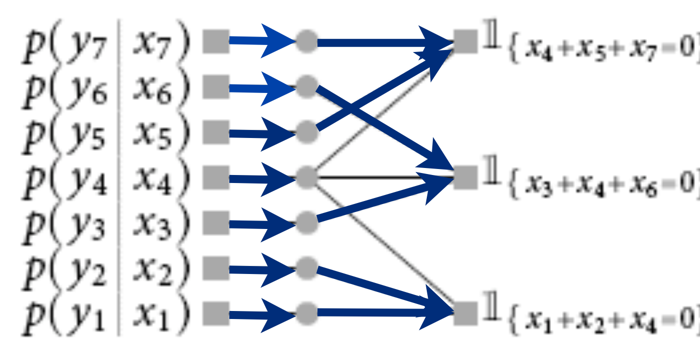




Decoding for Trees via Message Passing

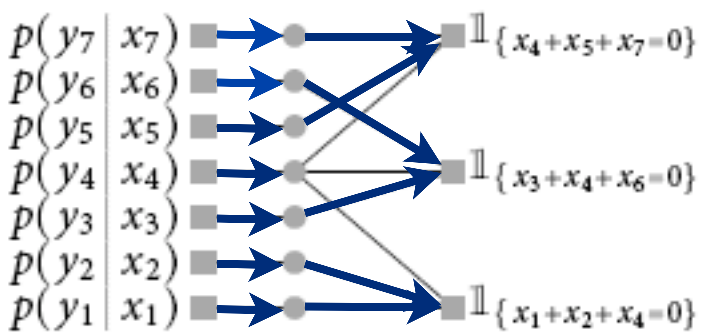


at variables we multiply messages

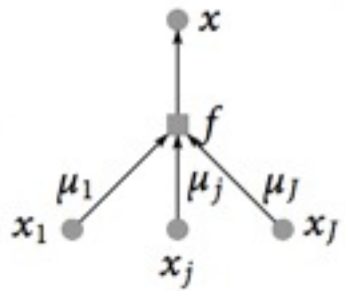


Decoding for Trees via Message Passing

at check nodes we multiply messages,
multiply with kernel and marginalize

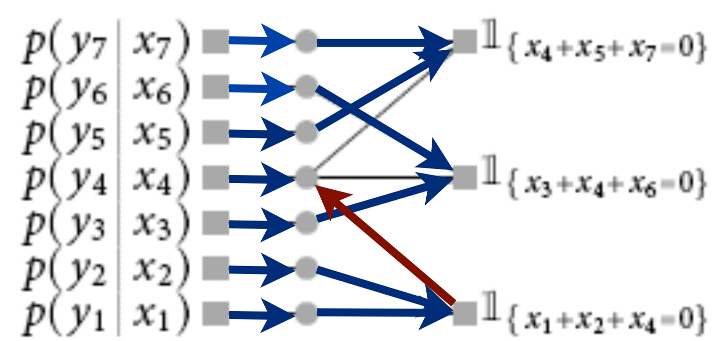


$$\mu(x) = \sum_{\sim x} f(x, x_1, \cdots, x_J) \prod_{j=1}^{J} \mu_j(x_j)$$



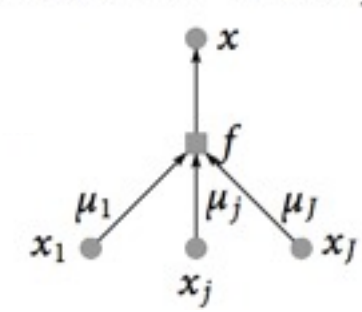
Decoding for Trees via Message Passing

at check nodes we multiply messages,
multiply with kernel and marginalize



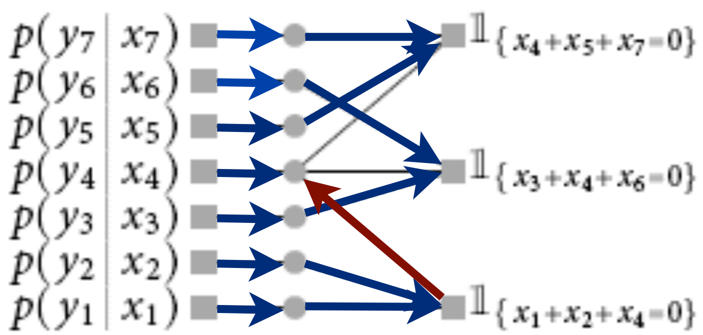
$\mu(x_4)=\sum_{\sim x4}\, 1_{\{x1+x2+x4=0\}}\, p(y_1|x_1)\; p(y_2|x_2)$

$\mu(x) = \sum_{\sim x} f(x, x_1, \cdots, x_J) \prod_{j=1}^{J} \mu_j(x_j)$



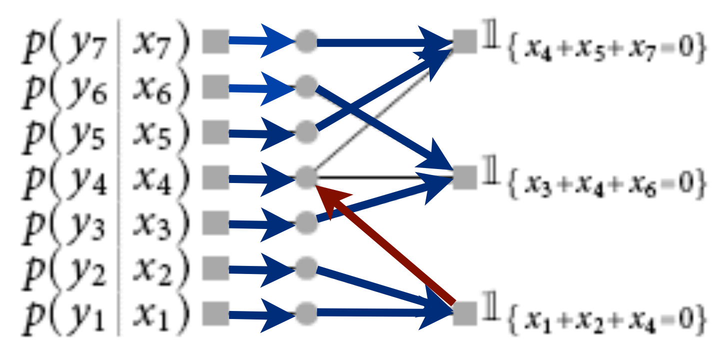
Decoding for Trees via Message Passing

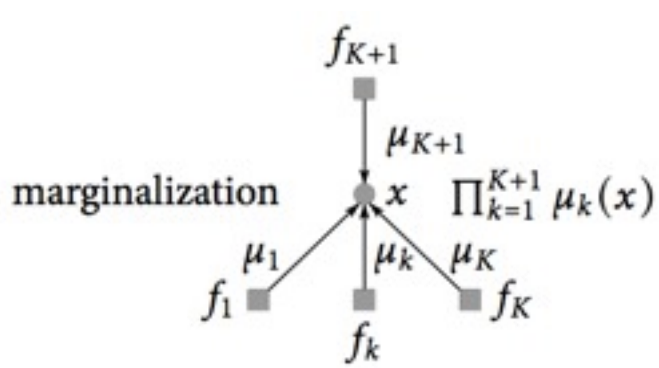
continue in this fashion until all messages along all edges have been determined



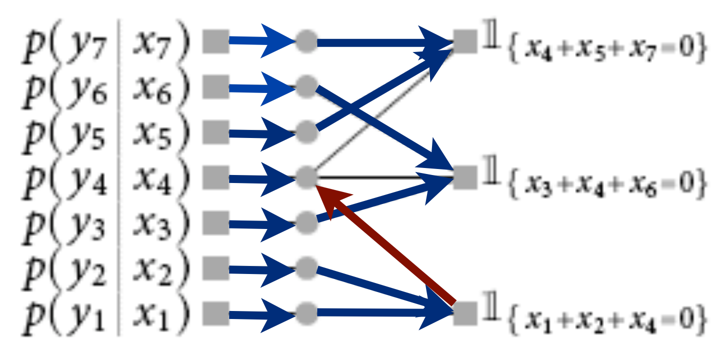
Decoding for Trees via Message Passing

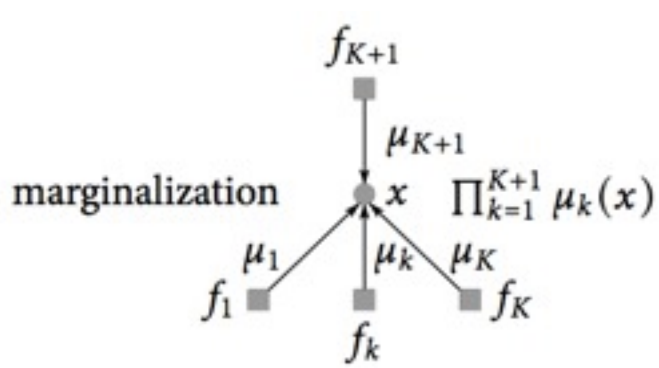
the final decision for each variable is given by the product of all incoming messages





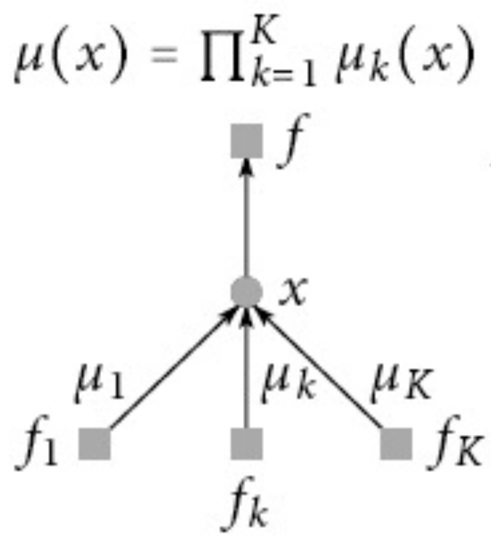
Decoding for Trees via Message Passing



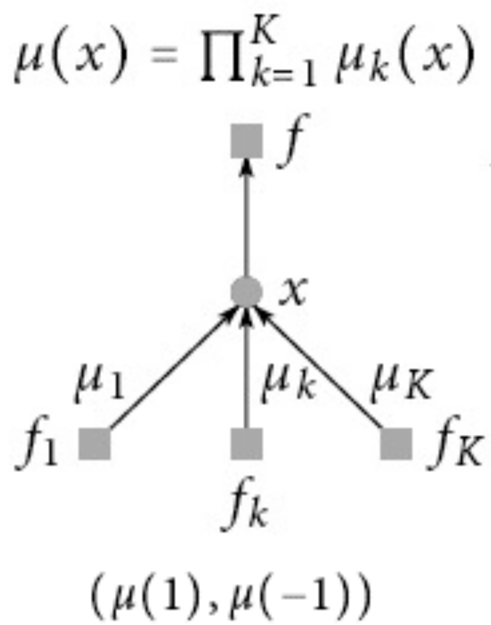


Simplification of Message-Passing Rules for Decoding

Simplification of Message-Passing Rules for Decoding

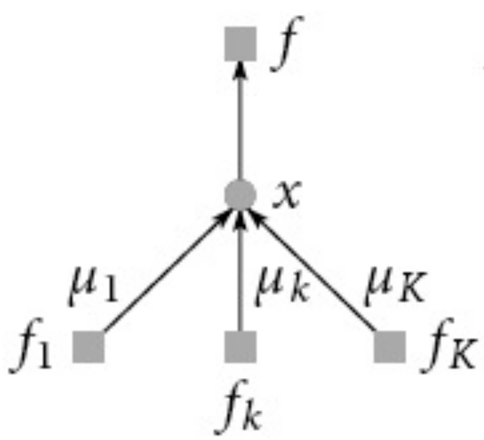


Simplification of Message-Passing Rules for Decoding



Simplification of Message-Passing Rules for Decoding

$$\mu(x) = \prod_{k=1}^{K} \mu_k(x)$$

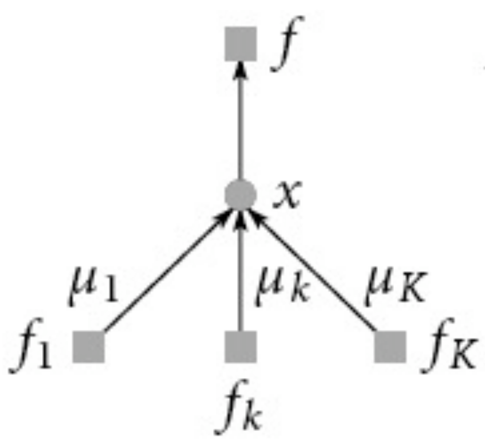


$$(\mu(1), \mu(-1))$$

$$r = \frac{\mu(1)}{\mu(-1)} = \frac{\prod_{k=1}^{K} \mu_k(1)}{\prod_{k=1}^{K} \mu_k(-1)} = \prod_{k=1}^{K} r_k$$

Simplification of Message-Passing Rules for Decoding

$$\mu(x) = \prod_{k=1}^{K} \mu_k(x)$$



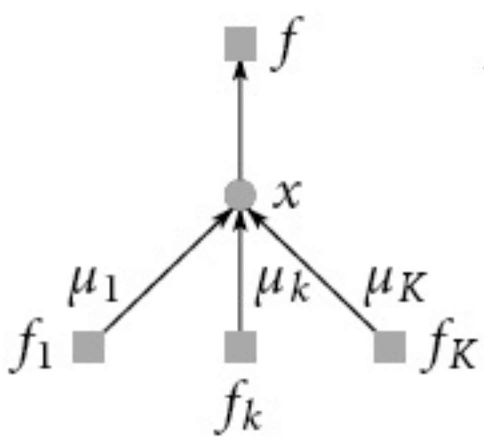
$$(\mu(1), \mu(-1))$$

$$r = \frac{\mu(1)}{\mu(-1)} = \frac{\prod_{k=1}^{K} \mu_k(1)}{\prod_{k=1}^{K} \mu_k(-1)} = \prod_{k=1}^{K} r_k$$

$$l = \sum_{k=1}^{K} l_k$$

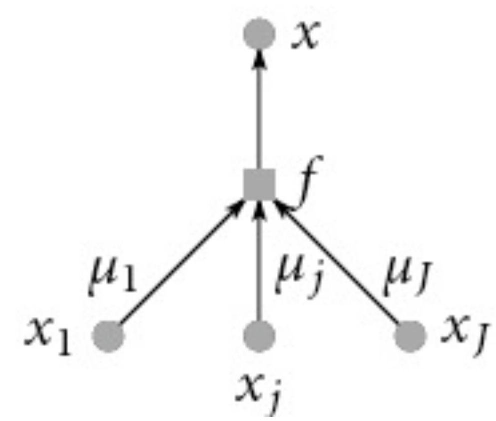
Simplification of Message-Passing Rules for Decoding

$$\mu(x) = \prod_{k=1}^{K} \mu_k(x)$$



$$(\mu(1), \mu(-1))$$

$$\mu(x) = \sum_{\sim x} f(x, x_1, \cdots, x_J) \prod_{j=1}^{J} \mu_j(x_j)$$

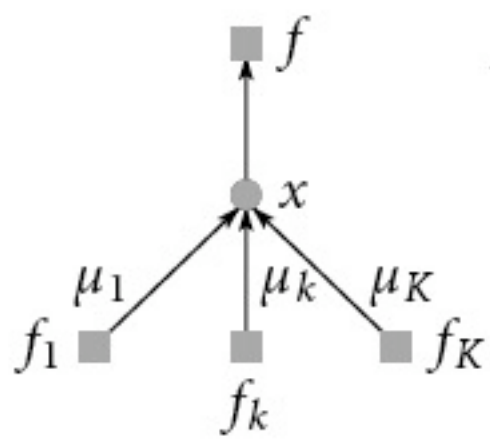


$$r = \frac{\mu(1)}{\mu(-1)} = \frac{\prod_{k=1}^{K} \mu_k(1)}{\prod_{k=1}^{K} \mu_k(-1)} = \prod_{k=1}^{K} r_k$$

$$l = \sum_{k=1}^{K} l_k$$

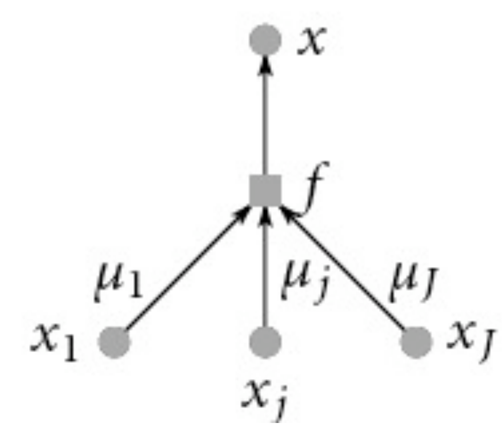
Simplification of Message-Passing Rules for Decoding

$\mu(x) = \prod_{k=1}^{K} \mu_k(x)$



$(\mu(1), \mu(-1))$

$\mu(x) = \sum_{\sim x} f(x, x_1, \cdots, x_J) \prod_{j=1}^{J} \mu_j(x_j)$



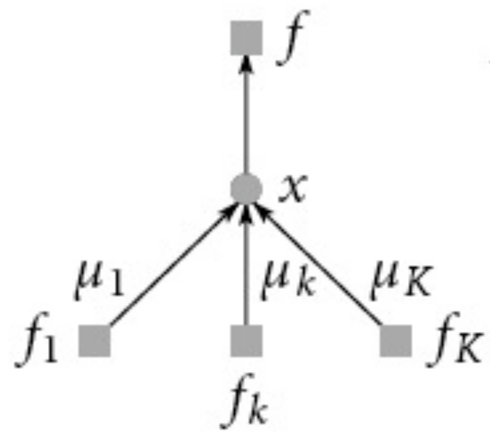
$(p_{Y_i|X_i}(y_i|1), p_{Y_i|X_i}(y_i|-1))$

$$r = \frac{\mu(1)}{\mu(-1)} = \frac{\prod_{k=1}^{K} \mu_k(1)}{\prod_{k=1}^{K} \mu_k(-1)} = \prod_{k=1}^{K} r_k$$

$$l = \sum_{k=1}^{K} l_k$$

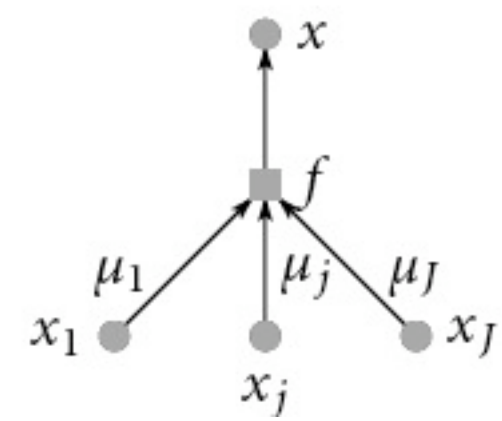
Simplification of Message-Passing Rules for Decoding

$$\mu(x) = \prod_{k=1}^{K} \mu_k(x)$$



$$(\mu(1), \mu(-1))$$

$$\mu(x) = \sum_{\sim x} f(x, x_1, \cdots, x_J) \prod_{j=1}^{J} \mu_j(x_j)$$



$$(p_{Y_i \mid X_i}(y_i \mid 1), p_{Y_i \mid X_i}(y_i \mid -1))$$

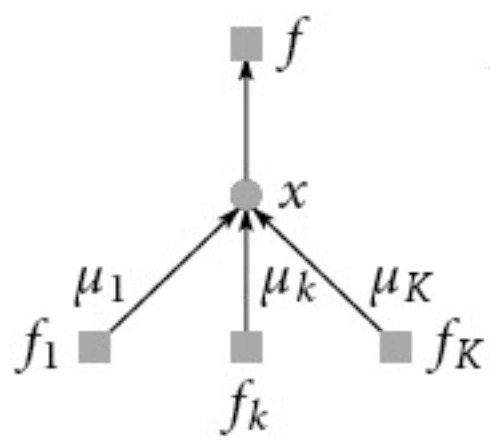
$$r = \frac{\mu(1)}{\mu(-1)} = \frac{\prod_{k=1}^{K} \mu_k(1)}{\prod_{k=1}^{K} \mu_k(-1)} = \prod_{k=1}^{K} r_k$$

$$r = \frac{1 + \prod_j \frac{r_j - 1}{r_j + 1}}{1 - \prod_j \frac{r_j - 1}{r_j + 1}}$$

$$l = \sum_{k=1}^{K} l_k$$

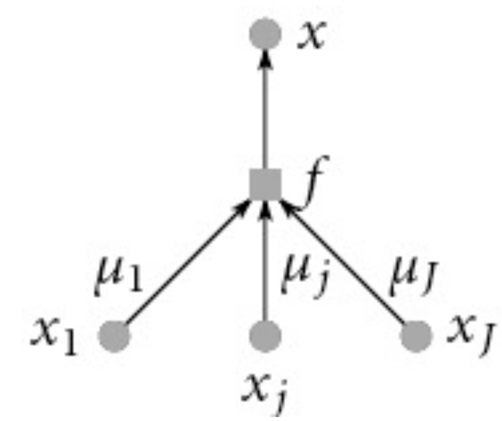
Simplification of Message-Passing Rules for Decoding

$$\mu(x) = \prod_{k=1}^{K} \mu_k(x)$$



$$(\mu(1), \mu(-1))$$

$$\mu(x) = \sum_{\sim x} f(x, x_1, \cdots, x_J) \prod_{j=1}^{J} \mu_j(x_j)$$



$$(p_{Y_i \mid X_i}(y_i \mid 1), p_{Y_i \mid X_i}(y_i \mid -1))$$

$$r = \frac{\mu(1)}{\mu(-1)} = \frac{\prod_{k=1}^{K} \mu_k(1)}{\prod_{k=1}^{K} \mu_k(-1)} = \prod_{k=1}^{K} r_k$$

$$r = \frac{1 + \prod_j \frac{r_j - 1}{r_j + 1}}{1 - \prod_j \frac{r_j - 1}{r_j + 1}}$$

$$l = \sum_{k=1}^{K} l_k$$

$$l = 2 \tanh^{-1}\left(\prod_{j=1}^{J} \tanh(l_j/2)\right)$$

Aside

Aside

$$\prod_{j=1}^{3}(r_j + 1) = 1 + r_1 + r_2 + r_3 + r_1 r_2 + r_1 r_3 + r_2 r_3 + r_1 r_2 r_3$$

Aside

$$\prod_{j=1}^{3}(r_j + 1) = 1 + r_1 + r_2 + r_3 + r_1r_2 + r_1r_3 + r_2r_3 + r_1r_2r_3$$

$$\prod_{j=1}^{3}(r_j - 1) = -1 + r_1 + r_2 + r_3 - r_1r_2 - r_1r_3 - r_2r_3 + r_1r_2r_3$$

Aside

$$\prod_{j=1}^{3}(r_j+1) = 1 + r_1 + r_2 + r_3 + r_1r_2 + r_1r_3 + r_2r_3 + r_1r_2r_3$$

$$\prod_{j=1}^{3}(r_j-1) = -1 + r_1 + r_2 + r_3 - r_1r_2 - r_1r_3 - r_2r_3 + r_1r_2r_3$$

$$\prod_{j=1}^{J}(r_j+1) + \prod_{j=1}^{J}(r_j-1) = 2 \sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} r_j^{(1+x_j)/2}.$$

Simplification of Message-Passing Rules for Decoding

$$r = \frac{\mu(1)}{\mu(-1)} = \frac{\sum_{\sim x} f(1, x_1, \ldots, x_J) \prod_{j=1}^{J} \mu_j(x_j)}{\sum_{\sim x} f(-1, x_1, \ldots, x_J) \prod_{j=1}^{J} \mu_j(x_j)}$$

$$= \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} \mu_j(x_j)}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} \mu_j(x_j)} = \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} \frac{\mu_j(x_j)}{\mu_j(-1)}}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} \frac{\mu_j(x_j)}{\mu_j(-1)}}$$

$$= \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} r_j^{(1+x_j)/2}}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} r_j^{(1+x_j)/2}} = \frac{\prod_{j=1}^{J} (r_j + 1) + \prod_{j=1}^{J} (r_j - 1)}{\prod_{j=1}^{J} (r_j + 1) - \prod_{j=1}^{J} (r_j - 1)}$$

$$f(x, x_1, \ldots, x_J) = \mathbb{1}_{\{\prod_{j=1}^{J} x_j = x\}}$$

Simplification of Message-Passing Rules for Decoding

$$r = \frac{\mu(1)}{\mu(-1)} = \frac{\sum_{\sim x} f(1, x_1, \ldots, x_J) \prod_{j=1}^{J} \mu_j(x_j)}{\sum_{\sim x} f(-1, x_1, \ldots, x_J) \prod_{j=1}^{J} \mu_j(x_j)}$$

$$f(x, x_1, \ldots, x_J) = \mathbb{1}_{\{\prod_{j=1}^{J} x_j = x\}}$$

$$= \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} \mu_j(x_j)}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} \mu_j(x_j)} = \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} \frac{\mu_j(x_j)}{\mu_j(-1)}}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} \frac{\mu_j(x_j)}{\mu_j(-1)}}$$

$$= \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} r_j^{(1+x_j)/2}}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} r_j^{(1+x_j)/2}} = \frac{\prod_{j=1}^{J}(r_j + 1) + \prod_{j=1}^{J}(r_j - 1)}{\prod_{j=1}^{J}(r_j + 1) - \prod_{j=1}^{J}(r_j - 1)}$$

$$r = \frac{1 + \prod_j \frac{r_j - 1}{r_j + 1}}{1 - \prod_j \frac{r_j - 1}{r_j + 1}}$$

Simplification of Message-Passing Rules for Decoding

$$r = \frac{\mu(1)}{\mu(-1)} = \frac{\sum_{\sim x} f(1, x_1, \ldots, x_J) \prod_{j=1}^{J} \mu_j(x_j)}{\sum_{\sim x} f(-1, x_1, \ldots, x_J) \prod_{j=1}^{J} \mu_j(x_j)}$$

$$f(x, x_1, \ldots, x_J) = \mathbb{1}_{\{\prod_{j=1}^{J} x_j = x\}}$$

$$= \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} \mu_j(x_j)}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} \mu_j(x_j)} = \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} \frac{\mu_j(x_j)}{\mu_j(-1)}}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} \frac{\mu_j(x_j)}{\mu_j(-1)}}$$

$$= \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} r_j^{(1+x_j)/2}}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} r_j^{(1+x_j)/2}} = \frac{\prod_{j=1}^{J}(r_j + 1) + \prod_{j=1}^{J}(r_j - 1)}{\prod_{j=1}^{J}(r_j + 1) - \prod_{j=1}^{J}(r_j - 1)}$$

$$r = \frac{1 + \prod_j \frac{r_j - 1}{r_j + 1}}{1 - \prod_j \frac{r_j - 1}{r_j + 1}} \qquad \frac{r-1}{r+1} = \prod_j \frac{r_j - 1}{r_j + 1}$$

Simplification of Message-Passing Rules for Decoding

$$r = \frac{\mu(1)}{\mu(-1)} = \frac{\sum_{\sim x} f(1, x_1, \ldots, x_J) \prod_{j=1}^{J} \mu_j(x_j)}{\sum_{\sim x} f(-1, x_1, \ldots, x_J) \prod_{j=1}^{J} \mu_j(x_j)}$$

$$f(x, x_1, \ldots, x_J) = \mathbb{1}_{\{\prod_{j=1}^{J} x_j = x\}}$$

$$= \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} \mu_j(x_j)}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} \mu_j(x_j)} = \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} \frac{\mu_j(x_j)}{\mu_j(-1)}}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} \frac{\mu_j(x_j)}{\mu_j(-1)}}$$

$$= \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} r_j^{(1+x_j)/2}}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} r_j^{(1+x_j)/2}} = \frac{\prod_{j=1}^{J}(r_j + 1) + \prod_{j=1}^{J}(r_j - 1)}{\prod_{j=1}^{J}(r_j + 1) - \prod_{j=1}^{J}(r_j - 1)}$$

$$r = \frac{1 + \prod_j \frac{r_j - 1}{r_j + 1}}{1 - \prod_j \frac{r_j - 1}{r_j + 1}} \qquad \frac{r-1}{r+1} = \prod_j \frac{r_j - 1}{r_j + 1} \qquad r = e^l$$

Simplification of Message-Passing Rules for Decoding

$$r = \frac{\mu(1)}{\mu(-1)} = \frac{\sum_{\sim x} f(1, x_1, \ldots, x_J) \prod_{j=1}^{J} \mu_j(x_j)}{\sum_{\sim x} f(-1, x_1, \ldots, x_J) \prod_{j=1}^{J} \mu_j(x_j)}$$

$$f(x, x_1, \ldots, x_J) = \mathbb{1}_{\{\prod_{j=1}^{J} x_j = x\}}$$

$$= \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} \mu_j(x_j)}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} \mu_j(x_j)} = \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} \frac{\mu_j(x_j)}{\mu_j(-1)}}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} \frac{\mu_j(x_j)}{\mu_j(-1)}}$$

$$= \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} r_j^{(1+x_j)/2}}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} r_j^{(1+x_j)/2}} = \frac{\prod_{j=1}^{J}(r_j + 1) + \prod_{j=1}^{J}(r_j - 1)}{\prod_{j=1}^{J}(r_j + 1) - \prod_{j=1}^{J}(r_j - 1)}$$

$$r = \frac{1 + \prod_j \frac{r_j - 1}{r_j + 1}}{1 - \prod_j \frac{r_j - 1}{r_j + 1}} \qquad \frac{r-1}{r+1} = \prod_j \frac{r_j - 1}{r_j + 1} \qquad r = e^{l} \qquad \frac{r-1}{r+1} = \tanh(l/2).$$

Simplification of Message-Passing Rules for Decoding

$$r = \frac{\mu(1)}{\mu(-1)} = \frac{\sum_{\sim x} f(1, x_1, \ldots, x_J) \prod_{j=1}^{J} \mu_j(x_j)}{\sum_{\sim x} f(-1, x_1, \ldots, x_J) \prod_{j=1}^{J} \mu_j(x_j)}$$

$$f(x, x_1, \ldots, x_J) = \mathbb{1}_{\{\prod_{j=1}^{J} x_j = x\}}$$

$$= \frac{\sum_{x_1, \ldots, x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} \mu_j(x_j)}{\sum_{x_1, \ldots x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} \mu_j(x_j)} = \frac{\sum_{x_1, \ldots, x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} \frac{\mu_j(x_j)}{\mu_j(-1)}}{\sum_{x_1, \ldots, x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} \frac{\mu_j(x_j)}{\mu_j(-1)}}$$

$$= \frac{\sum_{x_1, \ldots, x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} r_j^{(1+x_j)/2}}{\sum_{x_1, \ldots, x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} r_j^{(1+x_j)/2}} = \frac{\prod_{j=1}^{J}(r_j + 1) + \prod_{j=1}^{J}(r_j - 1)}{\prod_{j=1}^{J}(r_j + 1) - \prod_{j=1}^{J}(r_j - 1)}$$

$$r = \frac{1 + \prod_j \frac{r_j - 1}{r_j + 1}}{1 - \prod_j \frac{r_j - 1}{r_j + 1}} \qquad \frac{r-1}{r+1} = \prod_j \frac{r_j - 1}{r_j + 1} \qquad r = e^l \qquad \frac{r-1}{r+1} = \tanh(l/2)$$

$$\tanh(l/2) = \frac{r-1}{r+1} = \prod_{j=1}^{J} \frac{r_j - 1}{r_j + 1} = \prod_{j=1}^{J} \tanh(l_j/2)$$

Simplification of Message-Passing Rules for Decoding

$$r = \frac{\mu(1)}{\mu(-1)} = \frac{\sum_{\sim x} f(1, x_1, \ldots, x_J) \prod_{j=1}^{J} \mu_j(x_j)}{\sum_{\sim x} f(-1, x_1, \ldots, x_J) \prod_{j=1}^{J} \mu_j(x_j)}$$

$$f(x, x_1, \ldots, x_J) = \mathbb{1}_{\{\prod_{j=1}^{J} x_j = x\}}$$

$$= \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} \mu_j(x_j)}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} \mu_j(x_j)} = \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} \frac{\mu_j(x_j)}{\mu_j(-1)}}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} \frac{\mu_j(x_j)}{\mu_j(-1)}}$$

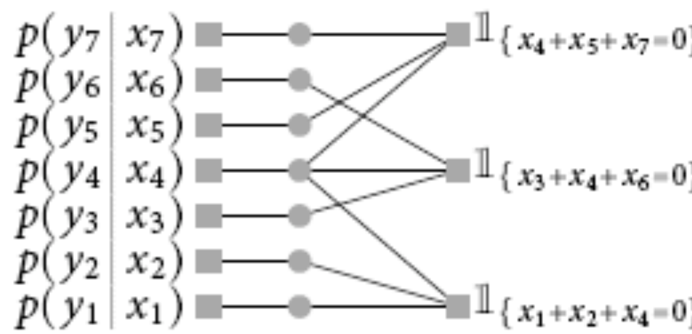
$$= \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = 1} \prod_{j=1}^{J} r_j^{(1+x_j)/2}}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^{J} x_j = -1} \prod_{j=1}^{J} r_j^{(1+x_j)/2}} = \frac{\prod_{j=1}^{J}(r_j + 1) + \prod_{j=1}^{J}(r_j - 1)}{\prod_{j=1}^{J}(r_j + 1) - \prod_{j=1}^{J}(r_j - 1)}$$

$$r = \frac{1 + \prod_j \frac{r_j - 1}{r_j + 1}}{1 - \prod_j \frac{r_j - 1}{r_j + 1}} \qquad \frac{r-1}{r+1} = \prod_j \frac{r_j - 1}{r_j + 1} \qquad r = e^l \qquad \frac{r-1}{r+1} = \tanh(l/2).$$

$$\tanh(l/2) = \frac{r-1}{r+1} = \prod_{j=1}^{J} \frac{r_j - 1}{r_j + 1} = \prod_{j=1}^{J} \tanh(l_j/2). \qquad l = 2\tanh^{-1}\left(\prod_{j=1}^{J} \tanh(l_j/2)\right).$$

Decoding for Trees via Message Passing

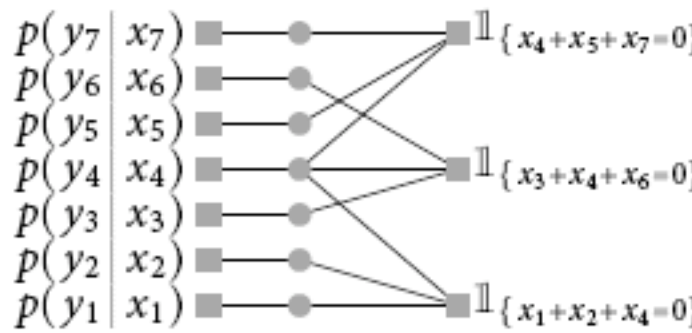
$$l = \sum_{k=1}^{K} l_k$$



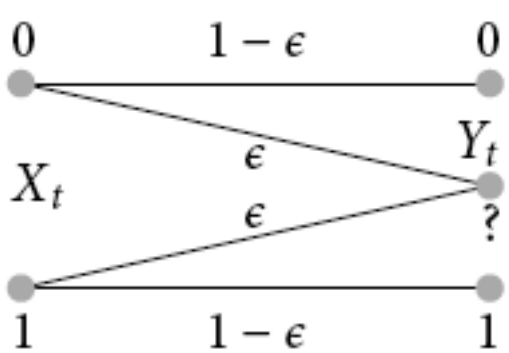
$$l = 2\tanh^{-1}\left(\prod_{j=1}^{J} \tanh(l_j/2)\right)$$

Simplification of Message-Passing Rules for Transmission over the BEC

$l = \sum_{k=1}^{K} l_k$

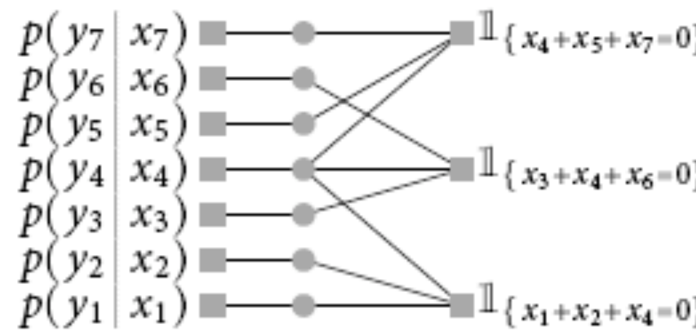


$l = 2\tanh^{-1}\left(\prod_{j=1}^{J} \tanh(l_j/2)\right)$



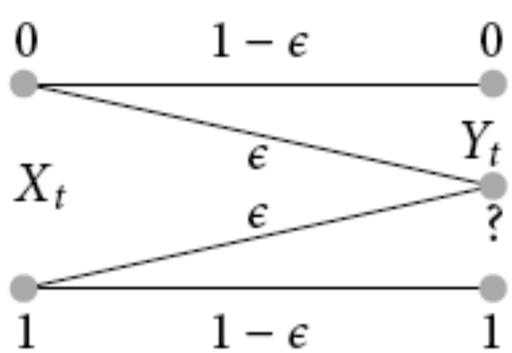
Simplification of Message-Passing Rules for Transmission over the BEC

$$l = \sum_{k=1}^{K} l_k$$



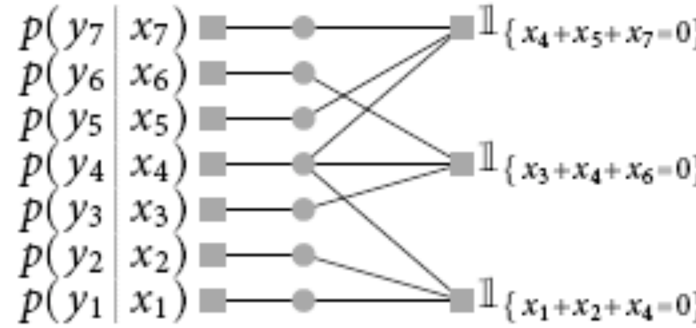
$$l = 2\tanh^{-1}\Big(\prod_{j=1}^{J} \tanh(l_j/2)\Big)$$

What are the initial log-likelihood values?



Simplification of Message-Passing Rules for Transmission over the BEC

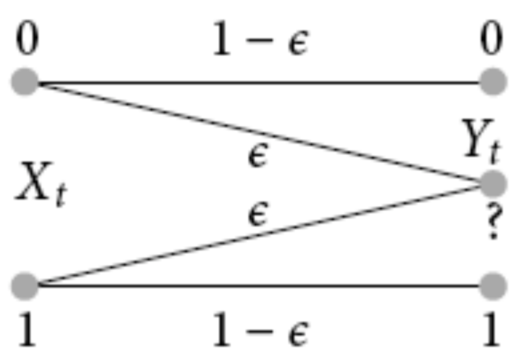
$$l = \sum_{k=1}^{K} l_k$$



$$l = 2\tanh^{-1}\left(\prod_{j=1}^{J} \tanh(l_j/2)\right)$$

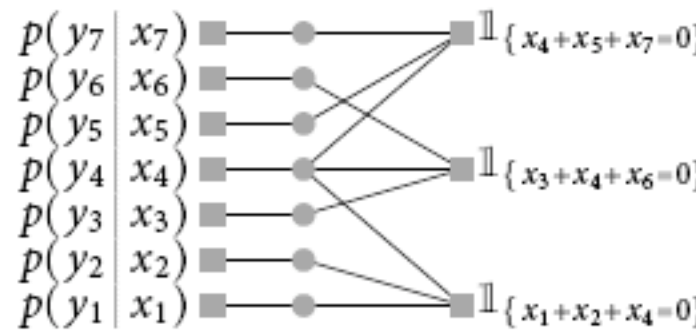
What are the initial log-likelihood values?

Assume that we send x=0; we then either receive y=0 or we receive y=?.

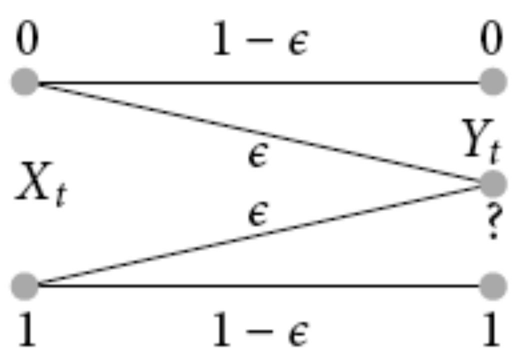


Simplification of Message-Passing Rules for Transmission over the BEC

$$l = \sum_{k=1}^{K} l_k$$



$$l = 2\tanh^{-1}\Big(\prod_{j=1}^{J} \tanh(l_j/2)\Big)$$



What are the initial log-likelihood values?

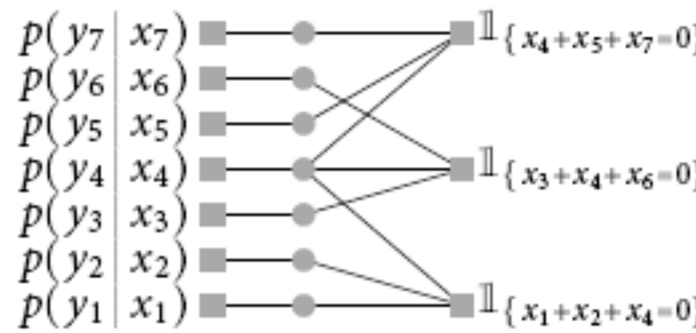
Assume that we send x=0; we then either receive y=0 or we receive y=?.

The corresponding log-likelihood values are:

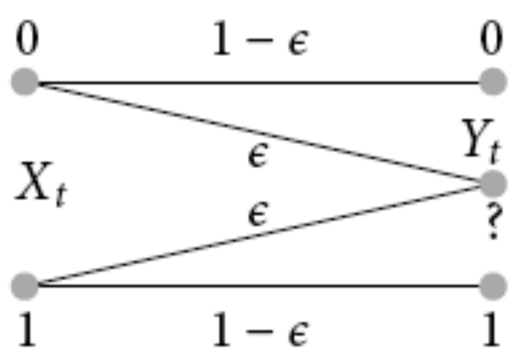
$$l(y=0) = \log(p(y=0|x=0)/p(y=0|x=1))$$
$$= \log((1-\varepsilon)/0) = +\infty$$
$$l(y=?) = \log(p(y=?|x=0)/p(y=?|x=1))$$
$$= \log(\varepsilon/\varepsilon) = 0$$

Simplification of Message-Passing Rules for Transmission over the BEC

$$l = \sum_{k=1}^{K} l_k$$



$$l = 2\tanh^{-1}\Big(\prod_{j=1}^{J} \tanh(l_j/2)\Big)$$



What are the initial log-likelihood values?

Assume that we send x=0; we then either receive y=0 or we receive y=?.

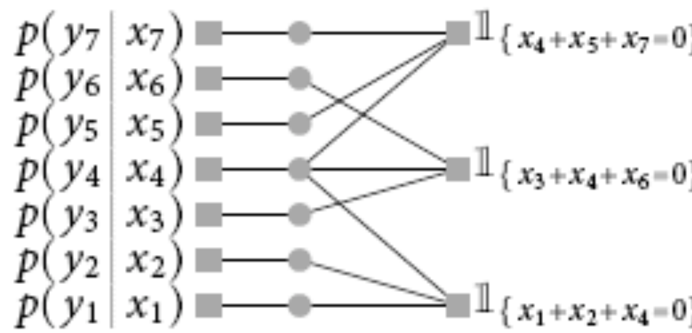
The corresponding log-likelihood values are:

$$l(y=0) = \log(p(y=0|x=0)/p(y=0|x=1))$$
$$= \log((1-\varepsilon)/0) = +\infty$$
$$l(y=?) = \log(p(y=?|x=0)/p(y=?|x=1))$$
$$= \log(\varepsilon/\varepsilon) = 0$$

This corresponds to the two options; we either are completely sure about the received bit or have no knowledge about it.

Simplification of Message-Passing Rules for Transmission over the BEC

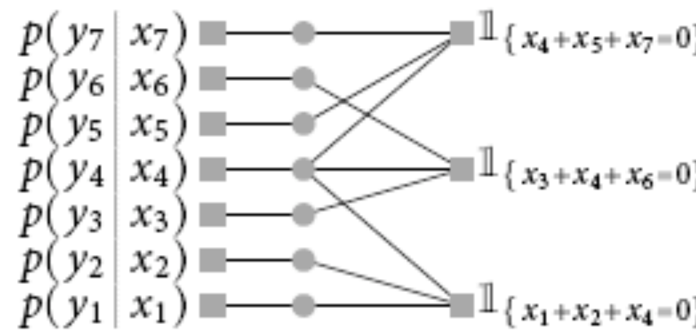
$l = \sum_{k=1}^{K} l_k$



$l = 2\tanh^{-1}\left(\prod_{j=1}^{J} \tanh(l_j/2)\right)$

Simplification of Message-Passing Rules for Transmission over the BEC

$$l = \sum_{k=1}^{K} l_k$$

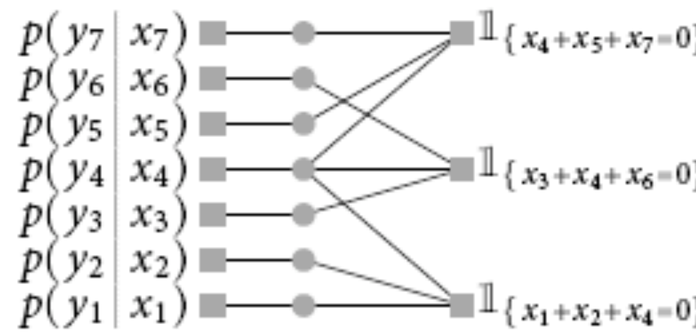


$$l = 2\tanh^{-1}\Big(\prod_{j=1}^{J} \tanh(l_j/2)\Big)$$

What does this mean for the message-passing rules?

Simplification of Message-Passing Rules for Transmission over the BEC

$l = \sum_{k=1}^{K} l_k$



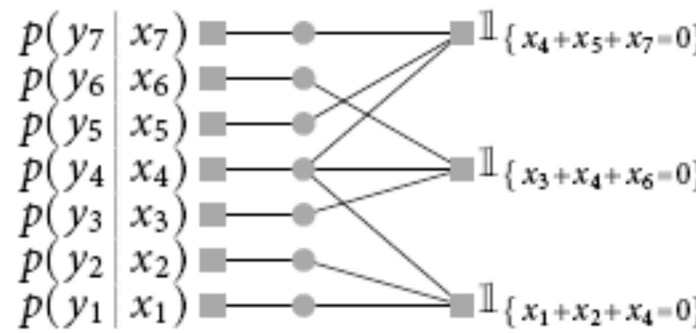
$l = 2\tanh^{-1}\left(\prod_{j=1}^{J}\tanh(l_j/2)\right)$

What does this mean for the message-passing rules?

variables: $l = \sum_{k=1}^{K} l_k$

Simplification of Message-Passing Rules for Transmission over the BEC

$l = \sum_{k=1}^{K} l_k$



$l = 2\tanh^{-1}\left(\prod_{j=1}^{J} \tanh(l_j/2)\right)$

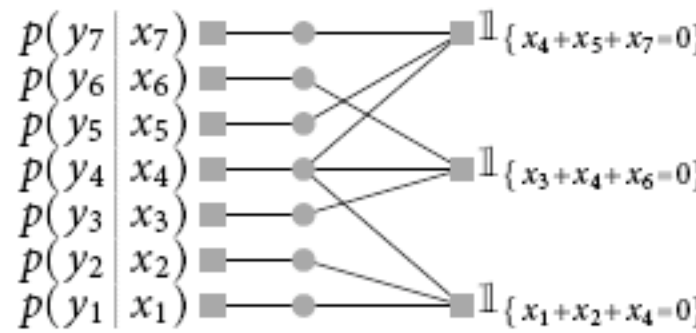
What does this mean for the message-passing rules?

variables: $l = \sum_{k=1}^{K} l_k$

If any of the inputs is $+\infty$ then the output is $+\infty$ otherwise the output is 0.

Simplification of Message-Passing Rules for Transmission over the BEC

$l = \sum_{k=1}^{K} l_k$



$l = 2\tanh^{-1}\left(\prod_{j=1}^{J} \tanh(l_j/2)\right)$

What does this mean for the message-passing rules?

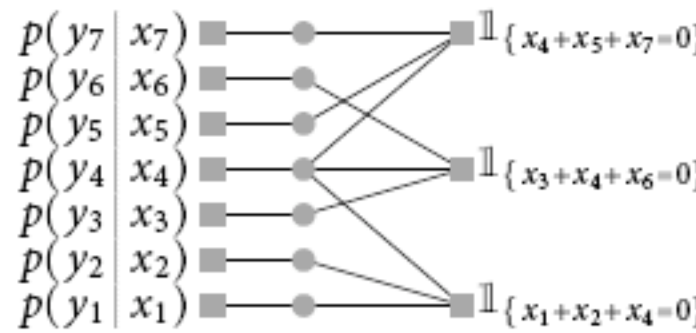
variables: $l = \sum_{k=1}^{K} l_k$

If any of the inputs is $+\infty$ then the output is $+\infty$ otherwise the output is 0.

checks: $l = 2\tanh^{-1}\left(\prod_{j=1}^{J} \tanh(l_j/2)\right)$

Simplification of Message-Passing Rules for Transmission over the BEC

$$l = \sum_{k=1}^{K} l_k$$



$$l = 2\tanh^{-1}\Big(\prod_{j=1}^{J} \tanh(l_j/2)\Big)$$

What does this mean for the message-passing rules?

variables: $l = \sum_{k=1}^{K} l_k$

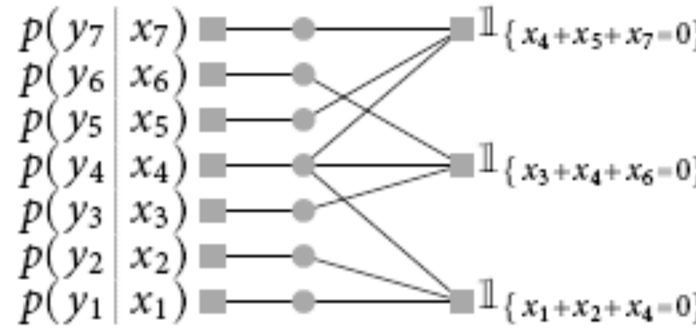
If any of the inputs is $+\infty$ then the output is $+\infty$ otherwise the output is 0.

checks: $l = 2\tanh^{-1}\Big(\prod_{j=1}^{J} \tanh(l_j/2)\Big)$

If all of the inputs are $+\infty$ then the output is $+\infty$ otherwise the output is 0.

Simplification of Message-Passing Rules for Transmission over the BEC

$l = \sum_{k=1}^{K} l_k$



$l = 2\tanh^{-1}\left(\prod_{j=1}^{J}\tanh(l_j/2)\right)$

What does this mean for the message-passing rules?

variables: $l = \sum_{k=1}^{K} l_k$

If any of the inputs is $+\infty$ then the output is $+\infty$ otherwise the output is 0.

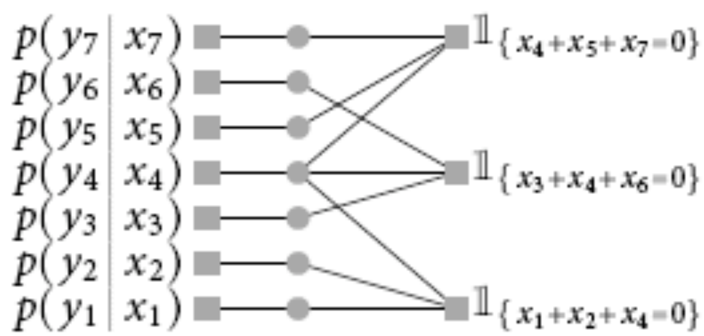
checks: $l = 2\tanh^{-1}\left(\prod_{j=1}^{J}\tanh(l_j/2)\right)$

If all of the inputs are $+\infty$ then the output is $+\infty$ otherwise the output is 0.

Instead of log-likelihood we can send the value of bit if log-likelihood is $+\infty$ or ? in case the log-likelihood is 0.

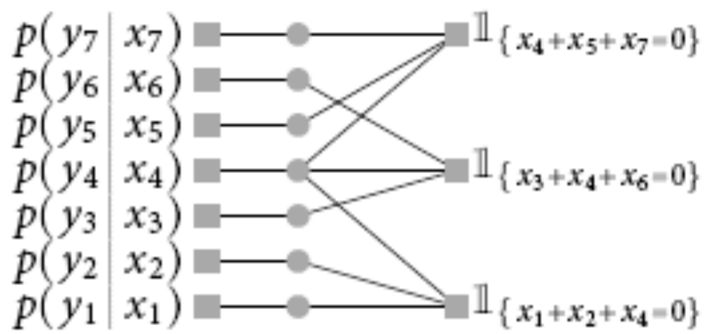
Summary and Limitations

Summary and Limitations



Summary and Limitations

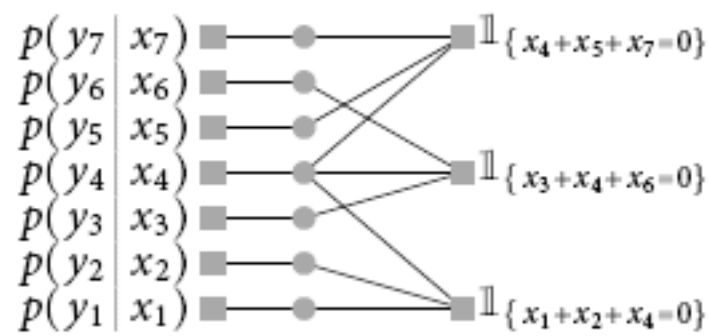
$$l = \sum_{k=1}^{K} l_k$$



$$l = 2\tanh^{-1}\Big(\prod_{j=1}^{J} \tanh(l_j/2)\Big)$$

Summary and Limitations

$$l = \sum_{k=1}^{K} l_k$$



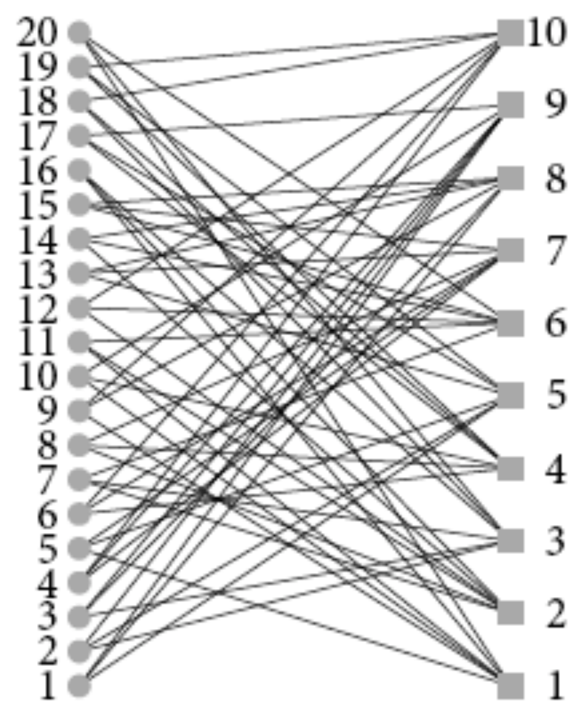
$$l = 2\tanh^{-1}\Big(\prod_{j=1}^{J}\tanh(l_j/2)\Big)$$

LEMMA 2.24 (BAD NEWS ABOUT CYCLE-FREE CODES). Let C be a binary linear code of rate r that admits a binary Tanner graph that is a forest. Then C contains at least $\frac{2r-1}{2}n$ codewords of weight 2.

Approach: Apply Algorithm to Graph with Loops

schedule

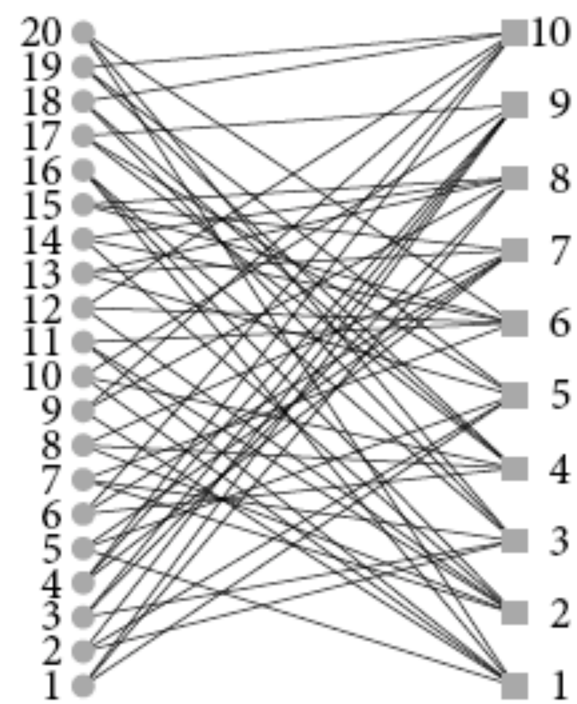
Approach: Apply Algorithm to Graph with Loops



schedule

Approach: Apply Algorithm to Graph with Loops

$l = \sum_{k=1}^{K} l_k$



$$l = 2\tanh^{-1}\left(\prod_{j=1}^{J} \tanh(l_j/2)\right)$$

schedule

Questions

- ✦ Can we determine performance of BP?
- ✦ How should we design graphs?
- ✦ How much loss of BP versus MAP?

References

R. G. GALLAGER, *Low-density parity-check codes*, IRE Trans. Inform. Theory, 8 (1962), pp. 21–28. [33, 66, 159, 264, 419, 434]

A. J. VITERBI, *Error bounds of convolutional codes and an asymptotically optimum decoding algorithm*, IEEE Trans. Inform. Theory, 13 (1967), pp. 260–269. [33, 34, 66]

L. BAHL, J. COCKE, F. JELINEK, AND J. RAVIV, *Optimal decoding of linear codes for minimizing symbol error rate*, IEEE Trans. Inform. Theory, 20 (1974), pp. 284–287. [34, 39, 66]

R. M. TANNER, *A recursive approach to low complexity codes*, IEEE Trans. Inform. Theory, 27 (1981), pp. 533–547. [65, 161, 261, 263]

N. WIBERG, *Codes and Decoding on General Graphs*, PhD thesis, Linköping University, S-581 83, Linköping, Sweden, 1996. [65, 66, 263, 266, 366]

S. M. AJI AND R. J. MCELIECE, *The generalized distributive law*, IEEE Trans. Inform. Theory, 46 (2000), pp. 325–343. [65, 66]

F. R. KSCHISCHANG, B. J. FREY, AND H.-A. LOELIGER, *Factor graphs and the sum-product algorithm*, IEEE Trans. Inform. Theory, 47 (2001), pp. 498–519. [65, 66]

Thank you for your attention!