Inference on graphs

- Given a graph and its conditionals or potentials compute

$p(\theta|e)$ (particular θ and e, marginalizing out other variables)

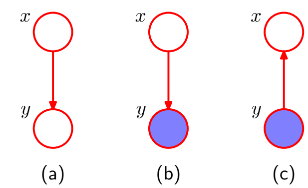
$p(X)$ (particular event, marginalizing out other variables)

argmax $p(\theta|e)$ (particular θ and e; marginalizing other variables)

argmax $p(\theta,\theta_N,e_N|e)$ (all variables, will nuisance / unobserved)

Inference on graphs

- Simplest example (Bayes' rule)
  - (a) model
  - (b) illustrates observed
  - (c) inference **reverses** the arrow



- Computationally

$$p(x|y)=\frac{p(y|x)\,p(x)}{\sum_{x'} p(y|x')\,p(x')}$$

Marginals on a chain

Recall

$$p(\mathbf{x})=p(x_1)\,p(x_2|x_1)\,p(x_3|x_2)\;\cdots\;p(x_{N-1}|x_{N-2})\,p(x_N|x_{N-1})$$

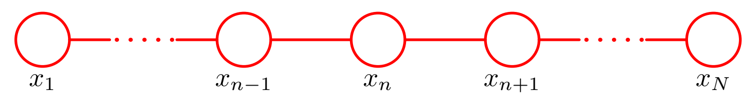
Converted to

$$p(\mathbf{x})=\psi_{1,2}(x_1,x_2)\,\psi_{2,3}(x_2,x_3)\;\ldots\;\psi_{N-2,N-1}(x_{N-2},x_{N-1})\,\psi_{N-1,N}(x_{N-1},x_N)$$

Assume N discrete variables, with K values each.

Compute the marginal of a node in the middle, $p(x_n)$

Marginals on a chain



$$p(\mathbf{x})=\psi_{1,2}(x_1,x_2)\,\psi_{2,3}(x_2,x_3)\;\ldots\;\psi_{N-2,N-1}(x_{N-2},x_{N-1})\,\psi_{N-1,N}(x_{N-1},x_N)$$

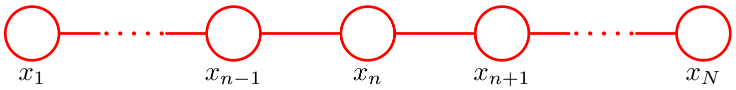
Direct calculation of $p(x_n)$

$$p(x_n)=\sum_{x_1}\sum_{x_2}\ldots\sum_{x_{n-1}}\sum_{x_{n+1}}\ldots\sum_{x_{N-1}}\sum_{x_N}p(\mathbf{x})$$

↑

Skip x_n

Marginals on a chain



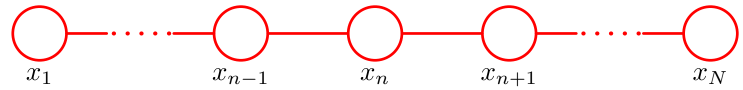
$$p(\mathbf{x}) = \psi_{1,2}(x_1,x_2)\psi_{2,3}(x_2,x_3) \; \dots \; \psi_{N-2,N-1}(x_{N-2},x_{N-1})\psi_{N-1,N}(x_{N-1},x_N)$$

Direct calculation of $p(x_n)$

$$p(x_n) = \sum_{x_1}\sum_{x_2} \dots \sum_{x_{n-1}}\sum_{x_{n+1}} \dots \sum_{x_{N-1}}\sum_{x_N} p(\mathbf{x})$$

Computational complexity is $O(K^N)$. Way too slow!

Computing marginals on a chain efficiently



$$p(\mathbf{x}) = \psi_{1,2}(x_1,x_2)\psi_{2,3}(x_2,x_3) \; \dots \; \psi_{N-2,N-1}(x_{N-2},x_{N-1})\psi_{N-1,N}(x_{N-1},x_N)$$

Main idea is to rearrange terms to exploit conditional independence.

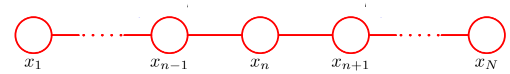
Fancy formulas from algebra

$$\underbrace{\sum_{x_1, x_2, \dots, x_N}}_{\text{all values of each}} f(x_1, x_2, \dots, x_N) = \underbrace{\sum_{x_1}\sum_{x_2}\dots\sum_{x_N}}_{\text{any order you like}} f(x_1, x_2, \dots, x_N)$$

(essentially a definition)

$$\left(\sum a_i\right)\left(\sum b_j\right) = \sum\sum a_i b_j$$

Back to marginals on a chain



$$p(\mathrm{x}) = \psi_{1,2}(x_1,x_2)\psi_{2,3}(x_2,x_3) \; \dots \; \psi_{N-2,N-1}(x_{N-2},x_{N-1})\psi_{N-1,N}(x_{N-1},x_N)$$

$$p(x_n) = \sum_{x_1}\sum_{x_2}\dots\sum_{x_{n-1}}\sum_{x_{n+1}}\dots\sum_{x_{N-1}}\sum_{x_N} \underbrace{\psi_{1,2}(x_1,x_2)\dots\psi_{n-1,n}(x_{n-1},x_n)}_{f_L(x_1,x_1,\dots,x_n)} \underbrace{\psi_{n,n+1}(x_n,x_{n+1})\dots\psi_{N-1,N}(x_{N-1},x_N)}_{f_R(x_n,x_{n+1},\dots,x_N)}$$

$$= \sum_{x_1}\sum_{x_2}\dots\sum_{x_{n-1}}\sum_{x_{n+1}}\dots\sum_{x_{N-1}}\sum_{x_N} f_L(x_1,x_1,\dots,x_n) f_R(x_n,x_{n+1},\dots,x_N)$$

$$= \left(\sum_{x_1}\sum_{x_2}\dots\sum_{x_{n-1}} f_L(x_1,x_1,\dots,x_n)\right)\left(\sum_{x_{n+1}}\dots\sum_{x_{N-1}}\sum_{x_N} f_R(x_n,x_{n+1},\dots,x_N)\right)$$

$$= \left(\sum_{x_1}\sum_{x_2}\dots\sum_{x_{n-1}}\prod_{i=1}^{n-1}\psi_{i,i+1}(x_i,x_{i+1})\right)\left(\sum_{x_{n+1}}\dots\sum_{x_{N-1}}\sum_{x_N}\prod_{i=n}^{N-1}\psi_{i,i+1}(x_i,x_{i+1})\right)$$

$$= \left(\sum_{x_{n-1}}\sum_{x_{n-2}}\dots\sum_{x_1}\prod_{i=1}^{n-1}\psi_{n-i,n-i+1}(x_{n-i},x_{n-i+1})\right)\left(\sum_{x_{n+1}}\dots\sum_{x_{N-1}}\sum_{x_N}\prod_{i=n}^{N-1}\psi_{i,i+1}(x_i,x_{i+1})\right)$$

(from previous slide)

$$p(x_n)=\left(\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_1}\prod_{i=1}^{n-1}\psi_{n-i,\,n-i+1}(x_{n-i},x_{n-i+1})\right)\left(\sum_{x_{n+1}}\cdots\sum_{x_{N-1}}\sum_{x_N}\prod_{i=n}^{N-1}\psi_{i,\,i+1}(x_i,x_{i+1})\right)$$

Notice that swapping the order (last step on previous) makes it so that the sums are symmetric, both ordered from outside to inside.

$$p(x_n)=\left(\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_1}\prod_{i=1}^{n-1}\psi_{n-i,\,n-i+1}(x_{n-i},x_{n-i+1})\right)\left(\sum_{x_{n+1}}\cdots\sum_{x_{N-1}}\sum_{x_N}\prod_{i=n}^{N-1}\psi_{i,\,i+1}(x_i,x_{i+1})\right)$$

$$\prod_{i=1}^{n-1}\psi_{n-i,\,n-i+1}(x_{n-i},x_{n-i+1})=\psi_{n-1,\,n}(x_{n-1},x_n)\psi_{n-2,\,n-1}(x_{n-2},x_{n-2})\ \dots\ \psi_{2,\,3}(x_2,x_3)\psi_{1,\,2}(x_1,x_2)$$

More warmup

$$\sum_{x_2}\sum_{x_1}\underbrace{\psi(x_2,x_3)}_{\substack{\text{No dependency on } x_1\\ \text{hence we can move}\\ \text{factor outside sum}\\ \text{over } x_1.}}\psi(x_1,x_2)=\sum_{x_2}\psi(x_2,x_3)\underbrace{\sum_{x_1}\psi(x_1,x_2)}_{\substack{\text{Vector of size over the}\\ \text{components of } x_2}}$$

(Recall the distibutive law: $ba+ca=a(b+c)$)

For example, in gory detail, for K=2, the first component of the sum, x_3^1, is:

$$\sum_{x_2}\sum_{x_1}\psi(x_2,x_3^1)\psi(x_1,x_2)$$
$$=\psi(x_2^1,x_3^1)\psi(x_1^1,x_2^1)+\psi(x_2^1,x_3^1)\psi(x_1^2,x_2^1)+\psi(x_2^2,x_3^1)\psi(x_1^1,x_2^2)+\psi(x_2^2,x_3^1)\psi(x_1^2,x_2^2)$$
$$=\psi(x_2^1,x_3^1)\bullet\left(\psi(x_1^1,x_2^1)+\psi(x_1^2,x_2^1)\right)+\psi(x_2^2,x_3^1)\bullet\left(\psi(x_1^1,x_2^2)+\psi(x_1^2,x_2^2)\right)$$
$$=\psi(x_2^1,x_3^1)\left(\sum_{\{x_1^i\}}\psi(x_1^i,x_2^1)\right)+\psi(x_2^2,x_3^1)\left(\sum_{\{x_1^i\}}\psi(x_1^i,x_2^1)\right)$$
$$=\sum_{\{x_2^i\}}\psi(x_2^i,x_3^1)\sum_{\{x_1^i\}}\psi(x_1^i,x_2^1)$$

More warmup

$$\sum_{x_2}\sum_{x_1}\underbrace{\psi(x_2,x_3)}_{\substack{\text{No dependency on } x_1\\ \text{hence we can move}\\ \text{factor outside sum}\\ \text{over } x_1.}}\psi(x_1,x_2)=\sum_{x_2}\psi(x_2,x_3)\underbrace{\sum_{x_1}\psi(x_1,x_2)}_{\substack{\text{Vector of size over the}\\ \text{components of } x_2}}$$

(Recall the distibutive law: $ba+ca=a(b+c)$)

This rule enables us to move sums “inwards” (or equivalently factors “outward”) to break big sums over big products into smaller pieces.

This works as long as what is being shuffled do not have variables in common (e.g., sum over x_1 and potential over x_2 and x_3).

$$p(x_n)=\left(\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_1}\prod_{i=1}^{n-1}\psi_{n-i,\,n-i+1}(x_{n-i},x_{n-i+1})\right)\left(\sum_{x_{n+1}}\cdots\sum_{x_{N-1}}\sum_{x_N}\prod_{i=n}^{N-1}\psi_{i,\,i+1}(x_i,x_{i+1})\right)$$

$$\prod_{i=1}^{n-1}\psi_{n-i,\,n-i+1}(x_{n-i},x_{n-i+1})=\psi_{n-1,\,n}(x_{n-1},x_n)\,\psi_{n-2,\,n-1}(x_{n-2},x_{n-2})\cdots\psi_{2,\,3}(x_2,x_3)\,\psi_{1,\,2}(x_1,x_2)$$

$$\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_1}\prod_{i=1}^{n-1}\psi_{n-i,n-i+1}(x_{n-i},x_{n-i+1})=\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_2}\prod_{i=1}^{n-2}\psi_{n-i,n-i+1}(x_{n-i},x_{n-i+1})\underbrace{\left\{\sum_{x_1}\psi_{1,2}(x_1,x_2)\right\}}_{\text{This has K elements, indexed by } x_2}$$

$$p(x_n)=\left(\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_1}\prod_{i=1}^{n-1}\psi_{n-i,\,n-i+1}(x_{n-i},x_{n-i+1})\right)\left(\sum_{x_{n+1}}\cdots\sum_{x_{N-1}}\sum_{x_N}\prod_{i=n}^{N-1}\psi_{i,\,i+1}(x_i,x_{i+1})\right)$$

$$\prod_{i=1}^{n-1}\psi_{n-i,\,n-i+1}(x_{n-i},x_{n-i+1})=\psi_{n-1,\,n}(x_{n-1},x_n)\,\psi_{n-2,\,n-1}(x_{n-2},x_{n-2})\cdots\psi_{2,\,3}(x_2,x_3)\,\psi_{1,\,2}(x_1,x_2)$$

$$\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_1}\prod_{i=1}^{n-1}\psi_{n-i,n-i+1}(x_{n-i},x_{n-i+1})=\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_2}\prod_{i=1}^{n-2}\psi_{n-i,n-i+1}(x_{n-i},x_{n-i+1})\left\{\sum_{x_1}\psi_{1,2}(x_1,x_2)\right\}$$

$$=\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_3}\prod_{i=1}^{n-3}\psi_{n-i,n-i+1}(x_{n-i-1},x_{n-i})\left\{\underbrace{\sum_{x_2}\psi_{2,3}(x_2,x_3)\left\{\sum_{x_1}\psi_{1,2}(x_1,x_2)\right\}}_{\text{Like a dot product for each } x_3\text{, with elements indexed by } x_2}\right\}$$

$$p(x_n)=\left(\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_1}\prod_{i=1}^{n-1}\psi_{n-i,\,n-i+1}(x_{n-i},x_{n-i+1})\right)\left(\sum_{x_{n+1}}\cdots\sum_{x_{N-1}}\sum_{x_N}\prod_{i=n}^{N-1}\psi_{i,\,i+1}(x_i,x_{i+1})\right)$$

$$\prod_{i=1}^{n-1}\psi_{n-i,\,n-i+1}(x_{n-i},x_{n-i+1})=\psi_{n-1,\,n}(x_{n-1},x_n)\,\psi_{n-2,\,n-1}(x_{n-2},x_{n-2})\cdots\psi_{2,\,3}(x_2,x_3)\,\psi_{1,\,2}(x_1,x_2)$$

$$\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_1}\prod_{i=1}^{n-1}\psi_{n-i,n-i+1}(x_{n-i},x_{n-i+1})=\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_2}\prod_{i=1}^{n-2}\psi_{n-i,n-i+1}(x_{n-i},x_{n-i+1})\left\{\sum_{x_1}\psi_{1,2}(x_1,x_2)\right\}$$

$$=\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_3}\prod_{i=1}^{n-3}\psi_{n-i,n-i+1}(x_{n-i-1},x_{n-i})\left\{\sum_{x_2}\psi_{2,3}(x_2,x_3)\left\{\sum_{x_1}\psi_{1,2}(x_1,x_2)\right\}\right\}$$

$$\cdots$$

$$=\left\{\sum_{x_{n-1}}\psi_{n-1,n}(x_{n-1},x_n)\cdots\left\{\sum_{x_3}\psi_{3,4}(x_3,x_4)\left\{\sum_{x_2}\psi_{2,3}(x_2,x_3)\left\{\sum_{x_1}\psi_{1,2}(x_1,x_2)\right\}\right\}\right\}\right\}$$

$$p(x_n)=\left(\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_1}\prod_{i=1}^{n-1}\psi_{n-i,\,n-i+1}(x_{n-i},x_{n-i+1})\right)\left(\sum_{x_{n+1}}\cdots\sum_{x_{N-1}}\sum_{x_N}\prod_{i=n}^{N-1}\psi_{i,\,i+1}(x_i,x_{i+1})\right)$$

where

$$\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_1}\prod_{i=1}^{n-1}\psi_{n-i,n-i+1}(x_{n-i},x_{n-i+1})=\left\{\sum_{x_{n-1}}\psi_{n-1,n}(x_{n-1},x_n)\cdots\left\{\sum_{x_3}\psi_{3,4}(x_3,x_4)\left\{\sum_{x_2}\psi_{2,3}(x_2,x_3)\left\{\sum_{x_1}\psi_{1,2}(x_1,x_2)\right\}\right\}\right\}\cdots\right\}$$

and

$$\sum_{x_{n+1}}\cdots\sum_{x_{N-1}}\sum_{x_N}\prod_{i=n}^{N-1}\psi_{i,i+1}(x_i,x_{i+1})=\left\{\sum_{x_{n+1}}\psi_{n,n+1}(x_n,x_{n+1})\cdots\left\{\sum_{x_{N-1}}\psi_{N-2,N-1}(x_{N-2},x_{N-1})\left\{\sum_{x_N}\psi_{N-1,N}(x_{N-1},x_N)\right\}\right\}\cdots\right\}$$

(Deriving the right factor is similar to doing
the left one which we did in detail.)

Matrix interpretation (for two variables)

$$\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_1}\prod_{i=1}^{n-1}\psi_{n-i,n-i+1}(x_{n-i},x_{n-i+1}) = \left\{\sum_{x_{n-1}}\psi_{n-1,n}(x_{n-1},x_n)\cdots\left\{\sum_{x_3}\psi_{3,4}(x_3,x_4)\left\{\sum_{x_2}\psi_{2,3}(x_2,x_3)\left\{\sum_{x_1}\psi_{1,2}(x_1,x_2)\right\}\right\}\right\}\cdots\right\}$$

$\psi_{i,\,i+1}(x_i,x_{i+1}) \Leftrightarrow Q_{i+1,\,i}$ (note transposition!)

$\sum_i \psi_{i,\,i+1}(x_i,x_{i+1}) \Leftrightarrow$ sums columns to get a vector V_{i+1}

$$\sum_{x_{i+1}}\psi_{i+1,\,i+2}(x_{i+1},x_{i+2})\left\{\sum_{x_i}\psi_{i,\,i+1}(x_i,x_{i+1})\right\} \Leftrightarrow Q_{i+2,\,i+1}\bullet \mathrm{V}_i$$

Computational Complexity

Suppose each variable has K values
What is the cost of evaluating the first factor?

$$\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_1}\prod_{i=1}^{n-1}\psi_{n-i,n-i+1}(x_{n-i},x_{n-i+1}) = \left\{\sum_{x_{n-1}}\psi_{n-1,n}(x_{n-1},x_n)\cdots\left\{\sum_{x_3}\psi_{3,4}(x_3,x_4)\left\{\underbrace{\sum_{x_2}\underbrace{\psi_{2,3}(x_2,x_3)\left\{\underbrace{\sum_{x_1}\psi_{1,2}(x_1,x_2)}_{\text{K sums of K values}}\right\}}_{\text{K evaluations of K products}}}_{\text{K sums of K values}}\right\}\right\}\cdots\right\}$$

The cost for computing the part shown in orange is $\mathrm{O}(K^2)$.

Computational Complexity

Suppose each variable has K values
What is the cost of evaluating the first factor?

$$\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_1}\prod_{i=1}^{n-1}\psi_{n-i,n-i+1}(x_{n-i},x_{n-i+1}) = \left\{\sum_{x_{n-1}}\psi_{n-1,n}(x_{n-1},x_n)\cdots\left\{\sum_{x_3}\psi_{3,4}(x_3,x_4)\left\{\underbrace{\sum_{x_2}\underbrace{\psi_{2,3}(x_2,x_3)\left\{\underbrace{\sum_{x_1}\psi_{1,2}(x_1,x_2)}_{\text{K sums of K values}}\right\}}_{\text{K evaluations of K products}}}_{\text{K sums of K values}}\right\}\right\}\cdots\right\}$$

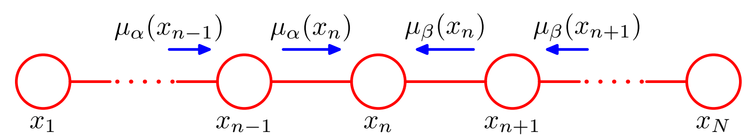
The cost of the left factor is $\mathrm{O}(N\cdot K^2)$.

The other factor is similar.

We see that the overall cost is $\mathrm{O}(N\cdot K^2)$.

Much better than the naive computation where we had $\mathrm{O}(K^N)$!

Message passing interpretation

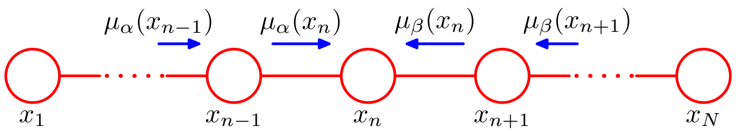


Define $\mu_\alpha(x_n)$ as a message passed from node x_{n-1} to node x_n.

Define $\mu_b(x_n)$ as a message passed from node x_{n+1} to node x_n.

Passing messages will correspond to the computation of taking input messages, and computing output messages.

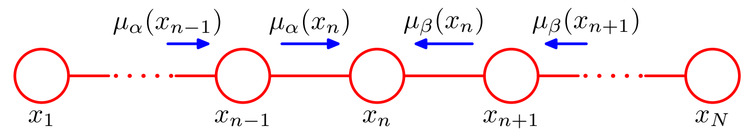
Message passing interpretation



$$\sum_{x_{n-1}}\sum_{x_{n-2}}\cdots\sum_{x_1}\prod_{i=1}^{n-1}\psi_{n-i,n-i+1}(x_{n-i},x_{n-i+1}) =$$

$$\underbrace{\left\{\sum_{x_{n-1}}\psi_{n-1,n}(x_{n-1},x_n)\cdots\underbrace{\left\{\sum_{x_3}\psi_{3,4}(x_3,x_4)\underbrace{\left\{\sum_{x_2}\psi_{2,3}(x_2,x_3)\underbrace{\left\{\sum_{x_1}\psi_{1,2}(x_1,x_2)\right\}}_{\mu_\alpha(x_2)}\right\}}_{\mu_\alpha(x_3)}\right\}}_{\mu_\alpha(x_4)}\cdots\right\}}_{\mu_\alpha(x_n)}$$

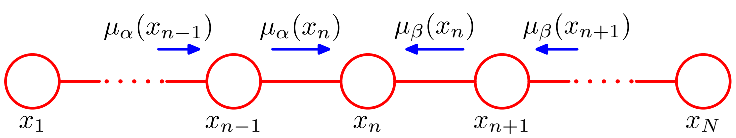
Message passing interpretation



$$\sum_{x_{n+1}}\cdots\sum_{x_{N-1}}\sum_{x_N}\prod_{i=n}^{N-1}\psi_{i,i+1}(x_i,x_{i+1}) =$$

$$\underbrace{\left\{\sum_{x_{n+1}}\psi_{n,n+1}(x_n,x_{n+1})\cdots\underbrace{\left\{\sum_{x_{N-1}}\psi_{N-2,N-1}(x_{N-2},x_{N-1})\underbrace{\left\{\sum_{x_N}\psi_{N-1,N}(x_{N-1},x_N)\right\}}_{\mu_b(x_{N-1})}\right\}}_{\mu_b(x_{N-2})}\cdots\right\}}_{\mu_b(x_n)}$$

Message passing interpretation

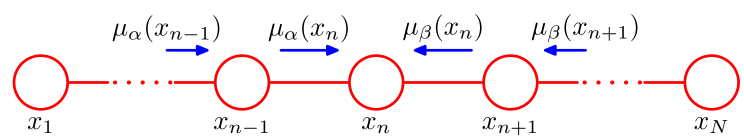


$$p(x_n) = \frac{1}{Z}\mu_a(x_n)\mu_b(x_n)$$

Algorithm

Send a message from x_1 to x_n.
Send a message from x_N to x_n.
Element wise multiply messages.
Normalize by sum over x_n (Z).

Computing all marginals



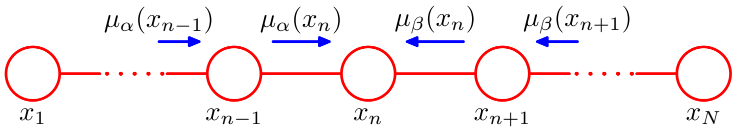
To compute all marginals, send a message from left to right, and right to left, storing the result.

Now compute any marginal as before.

This way, computing all marginals is only twice as expensive as computing one of them.

Normalization constant is easily computed using any node.

What if a node is observed?



If a node is observed, then we do the obvious. Specifically, we clamp the values of variables to the particular case.

This means that messages flowing into it, do not affect messages flowing out, which are set to the “clamped” value.