Message Passing/Belief Propagation

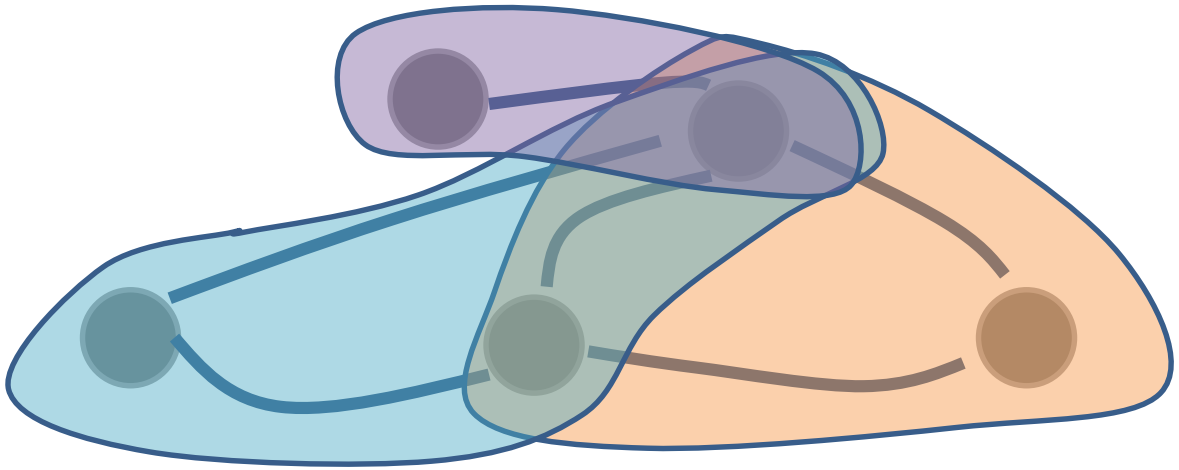
CMSC 691

UMBC

Markov Random Fields: Undirected Graphs

clique: subset of nodes, where nodes are pairwise connected

maximal clique: a clique that cannot add a node and remain a clique



$$p(x_1, x_2, x_3, \ldots, x_N) = \frac{1}{Z} \prod_C \psi_C(x_c)$$

variables part of the clique C

global normalization

maximal cliques

potential function (not necessarily a probability!)

Q: What restrictions should we place on the potentials ψ_C?

A: $\psi_C \geq 0$ (or $\psi_C > 0$)

Terminology: Potential Functions

$$p(x_1, x_2, x_3, \ldots, x_N) = \frac{1}{Z} \prod_C \psi_C(x_c)$$

energy function (for clique C)

(get the total energy of a configuration by summing the individual energy functions)

$$\psi_C(x_c) = \underbrace{\exp -E(x_C)}$$

Boltzmann distribution

MRFs as Factor Graphs

Undirected graphs: G=(V,E) that represents $p(X_1, \ldots, X_N)$

Factor graph of *p*: Bipartite graph of evidence nodes X, factor nodes F, and edges T

Evidence nodes X are the random variables

Factor nodes F take values associated with the *potential functions*

Edges show what variables are used in which factors

MRFs as Factor Graphs

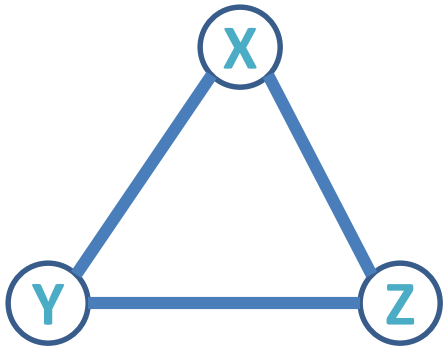
Undirected graphs: G=(V,E) that represents $p(X_1, \dots, X_N)$

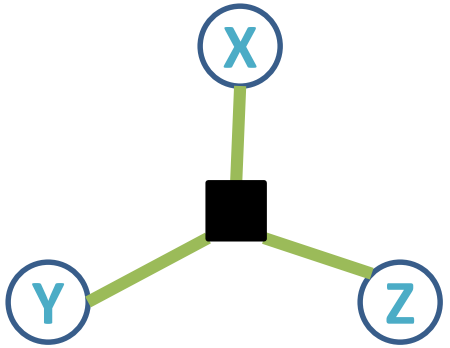
Factor graph of p: Bipartite graph of evidence nodes X, factor nodes F, and edges T

Evidence nodes X are the random variables

Factor nodes F take values associated with the *potential functions*

Edges show what variables are used in which factors





Outline

Message Passing: Graphical Model Inference

Example: Linear Chain CRF

Two Problems for Undirected Models

$$p(x_1, x_2, x_3, \ldots, x_N) = \frac{1}{Z} \prod_C \psi_C(x_c)$$

Finding the normalizer

$$Z = \sum_x \prod_c \psi_c(x_c)$$

Computing the marginals

Sum over all variable combinations, with the x_n coordinate fixed

$$Z_n(v) = \sum_{x:x_n=v} \prod_c \psi_c(x_c)$$

Example: 3 variables, fix the 2nd dimension

$$Z_2(v) = \sum_{x_1} \sum_{x_3} \prod_c \psi_c(x = (x_1, v, x_3))$$

Two Problems for Undirected Models

$$p(x_1, x_2, x_3, \ldots, x_N) = \frac{1}{Z} \prod_C \psi_C(x_c)$$

Finding the normalizer

$$Z = \sum_x \prod_c \psi_c(x_c)$$

Q: Why are these difficult?

A: Many different combinations

Computing the marginals

Sum over all variable combinations, with the x_n coordinate fixed

$$Z_n(v) = \sum_{x:x_n=v} \prod_c \psi_c(x_c)$$

Example: 3 variables, fix the 2nd dimension

$$Z_2(v) = \sum_{x_1} \sum_{x_3} \prod_c \psi_c(x = (x_1, v, x_3))$$

Message Passing: Count the Soldiers

If you are the front soldier in the line, say the number ‘one’ to the soldier behind you.

If you are the rearmost soldier in the line, say the number ‘one’ to the soldier in front of you.

If a soldier ahead of or behind you says a number to you, add one to it, and say the new number to the soldier on the other side

Message Passing: Count the Soldiers

If you are the front soldier in the line, say the number ‘one’ to the soldier behind you.

If you are the rearmost soldier in the line, say the number ‘one’ to the soldier in front of you.

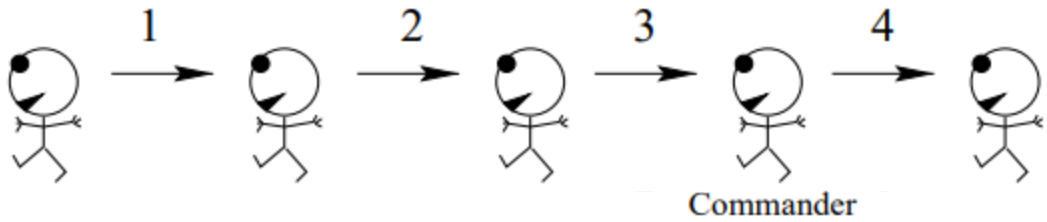
If a soldier ahead of or behind you says a number to you, add one to it, and say the new number to the soldier on the other side

Message Passing: Count the Soldiers

If you are the front soldier in the line, say the number ‘one’ to the soldier behind you.

If you are the rearmost soldier in the line, say the number ‘one’ to the soldier in front of you.

If a soldier ahead of or behind you says a number to you, add one to it, and say the new number to the soldier on the other side

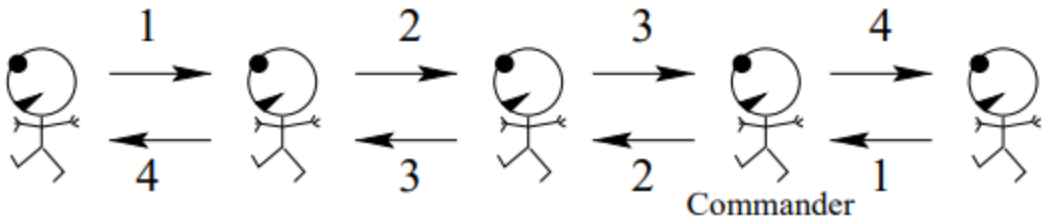


Message Passing: Count the Soldiers

If you are the front soldier in the line, say the number ‘one’ to the soldier behind you.

If you are the rearmost soldier in the line, say the number ‘one’ to the soldier in front of you.

If a soldier ahead of or behind you says a number to you, add one to it, and say the new number to the soldier on the other side



Sum-Product Algorithm

Main idea: message passing

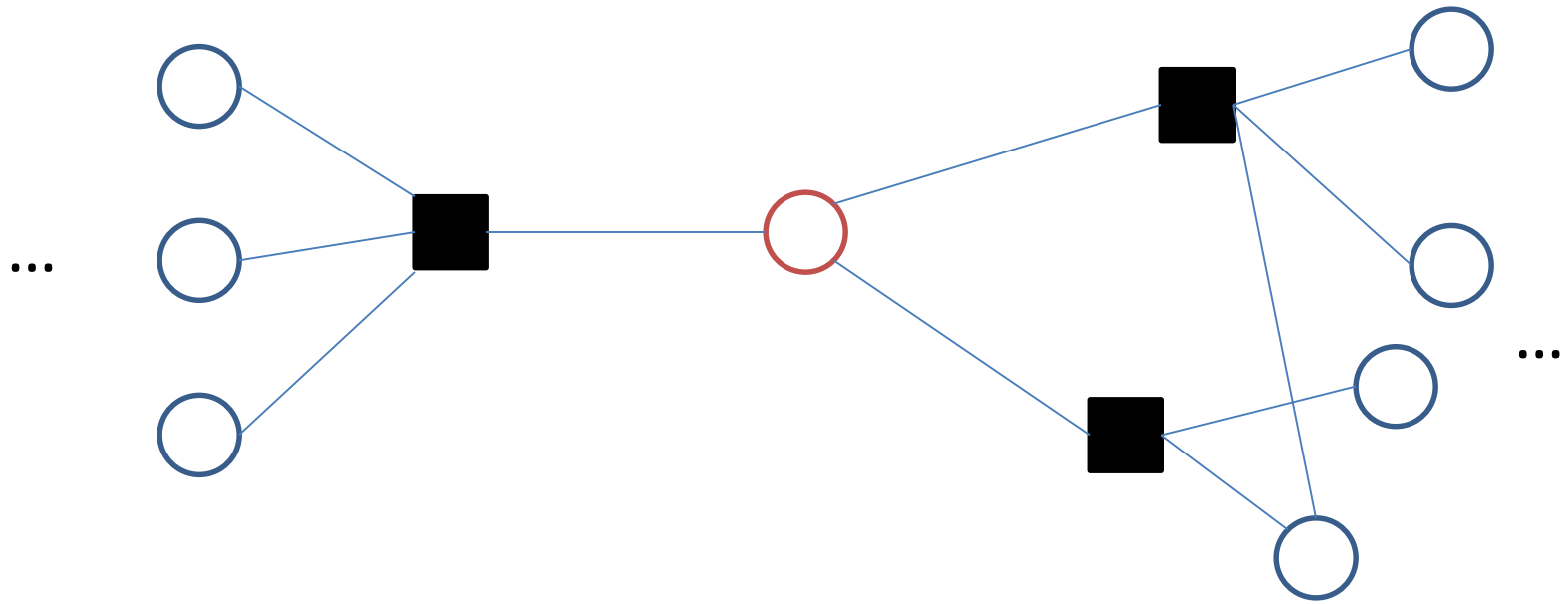
An exact inference algorithm for tree-like graphs

Belief propagation (forward-backward for HMMs) is a special case

Sum-Product

definition of marginal

$$p(x_i = v) = \sum_{x : x_i = v} p(x_1, x_2, \ldots, x_i, \ldots, x_N)$$

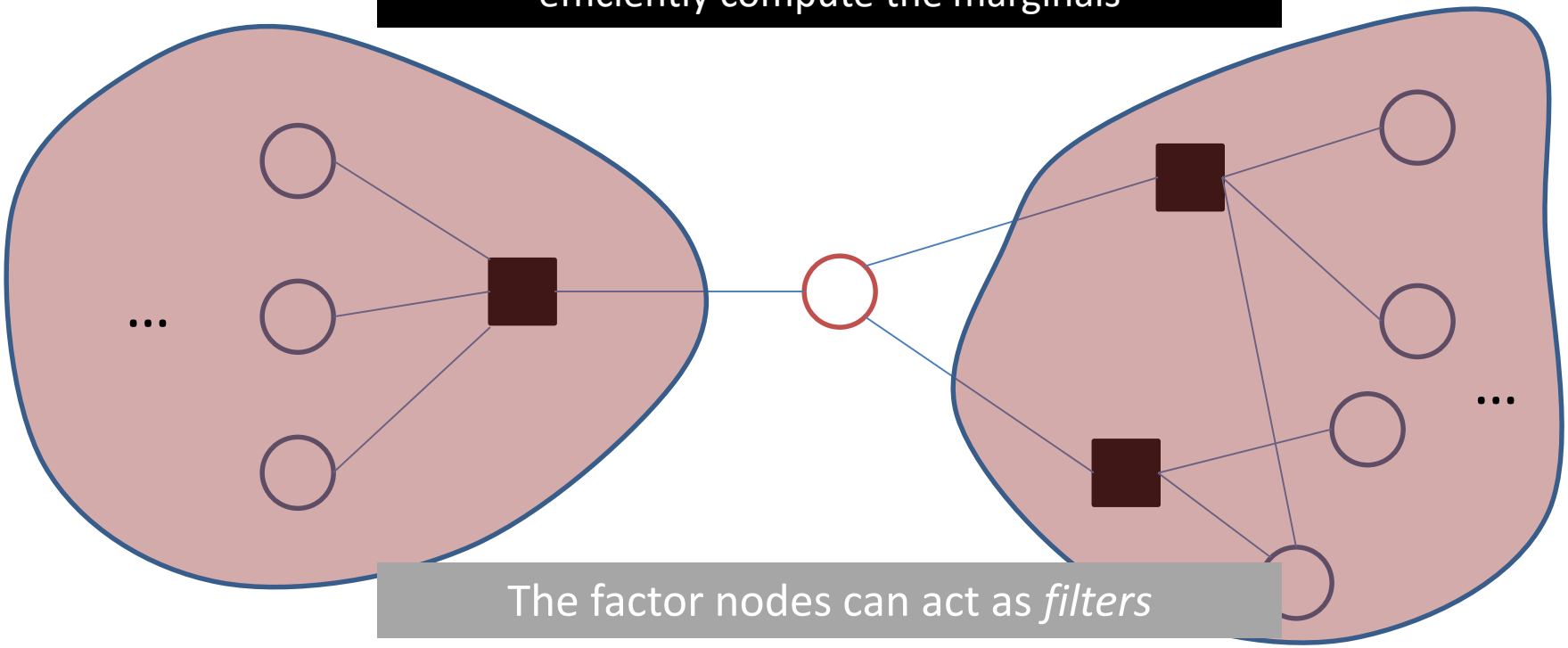


Sum-Product

definition of marginal

$$p(x_i = v) = \sum_{x : x_i = v} p(x_1, x_2, \ldots, x_i, \ldots, x_N)$$

main idea: use bipartite nature of graph to efficiently compute the marginals



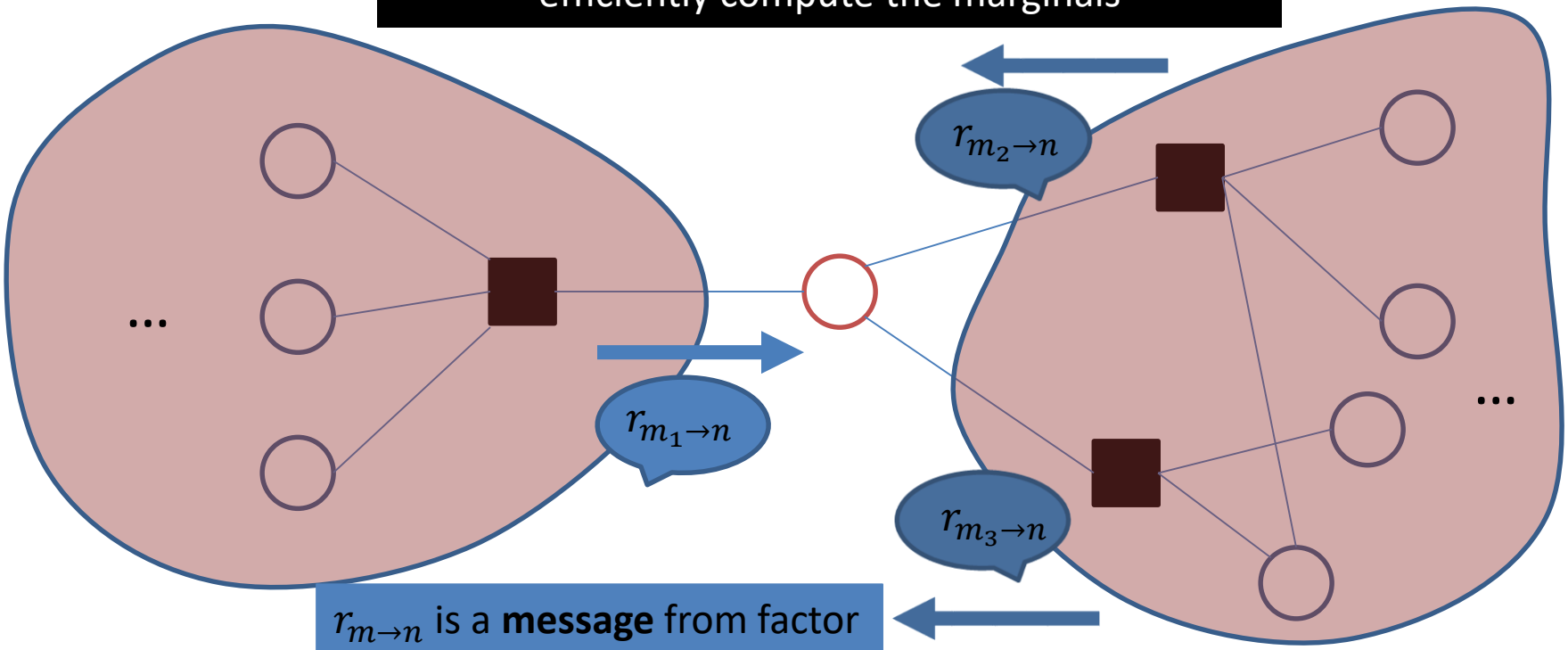
The factor nodes can act as *filters*

Sum-Product

definition of marginal

$$p(x_i = v) = \sum_{x:x_i=v} p(x_1, x_2, \ldots, x_i, \ldots, x_N)$$

main idea: use bipartite nature of graph to efficiently compute the marginals



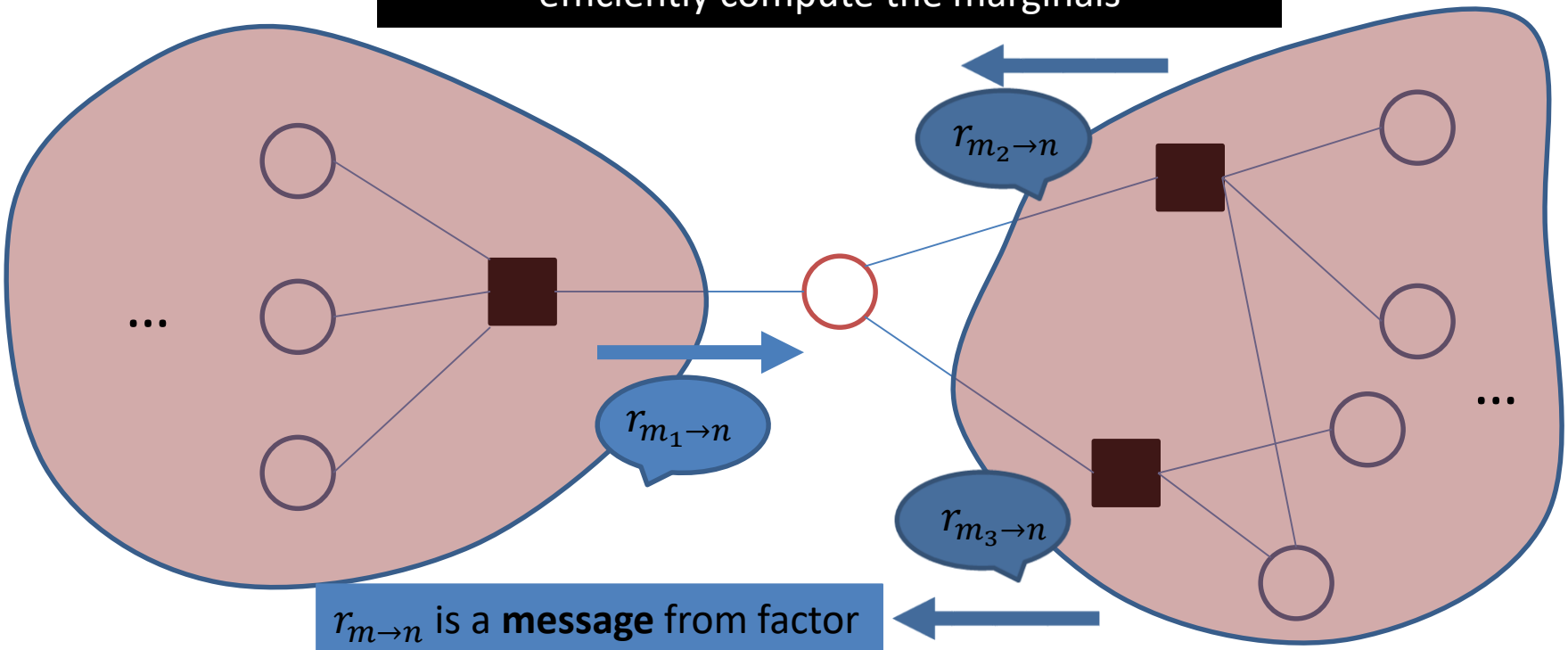
$r_{m \to n}$ is a **message** from factor node m to evidence node n

Sum-Product

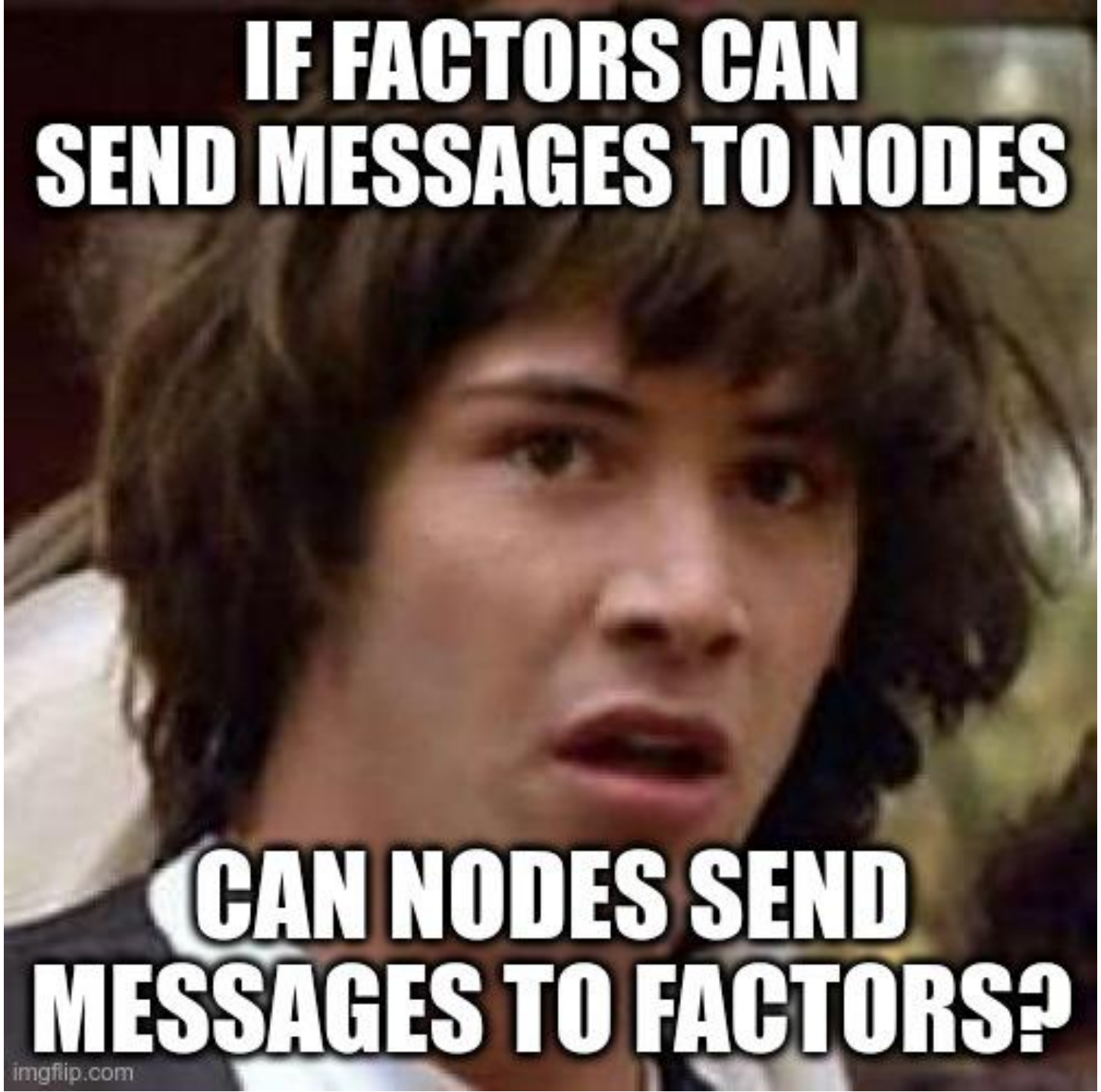
alternative marginal computation

$$p(x_i = v) = \frac{\prod_f r_{f \to x_i}(x_i = v)}{\sum_w \prod_f r_{f \to x_i}(x_i = w)} \propto \prod_f r_{f \to x_i}(x_i)$$

main idea: use bipartite nature of graph to efficiently compute the marginals

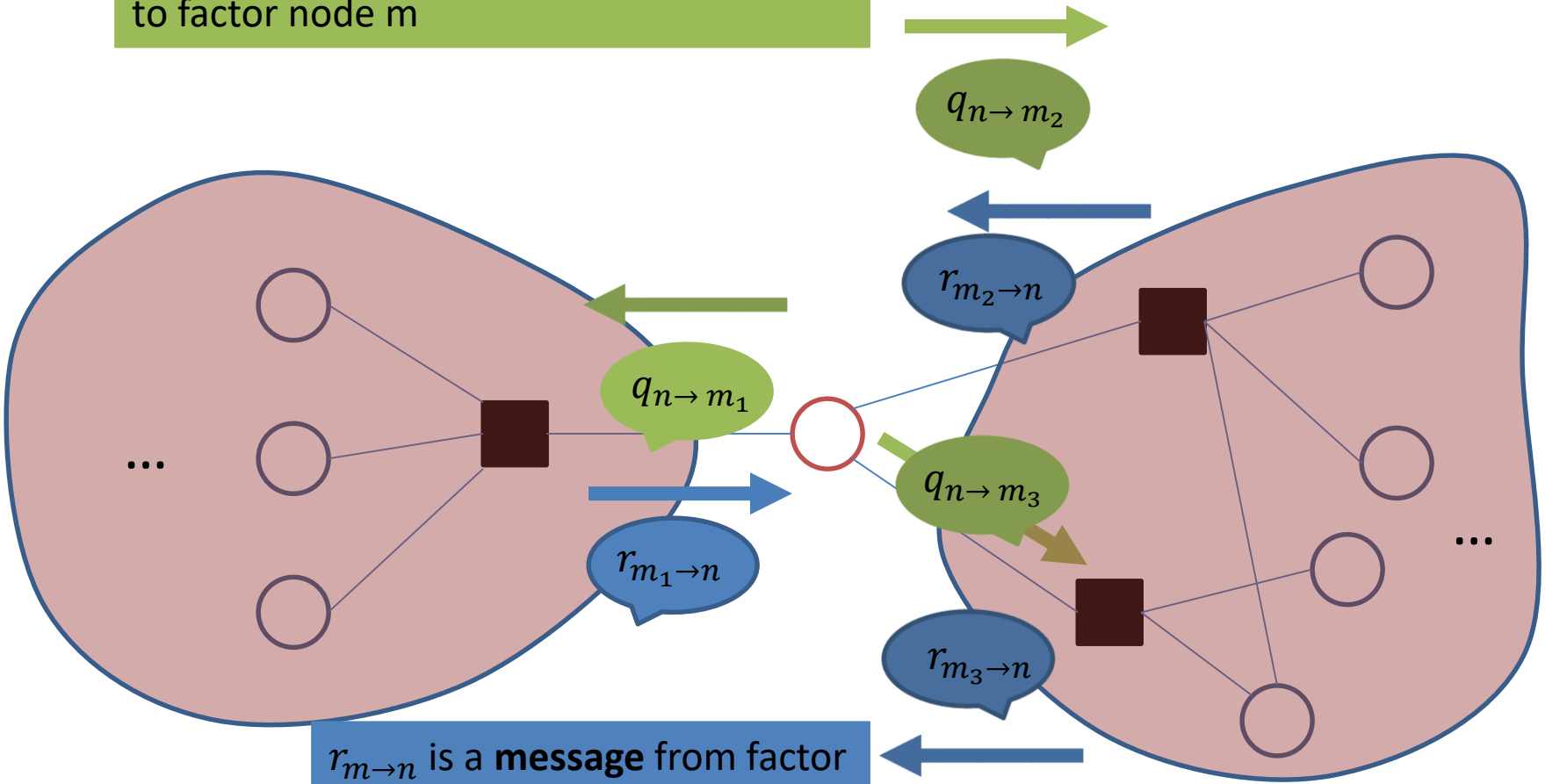


$r_{m \to n}$ is a **message** from factor node m to evidence node n


IF FACTORS CAN
SEND MESSAGES TO NODES
CAN NODES SEND
MESSAGES TO FACTORS?
imgflip.com

Sum-Product

$q_{n \to n}$ is a **message** from evidence node n to factor node m



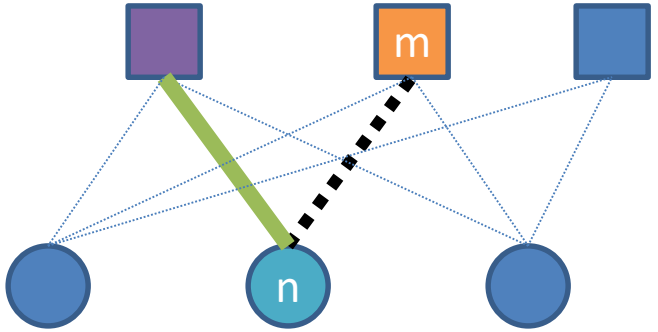
$r_{m \to n}$ is a **message** from factor node m to evidence node n

Sum-Product

From variables to factors

$q_{n\to m}(x_n) =$ *n aggregates information from the rest of its graph via its neighbors*

.....



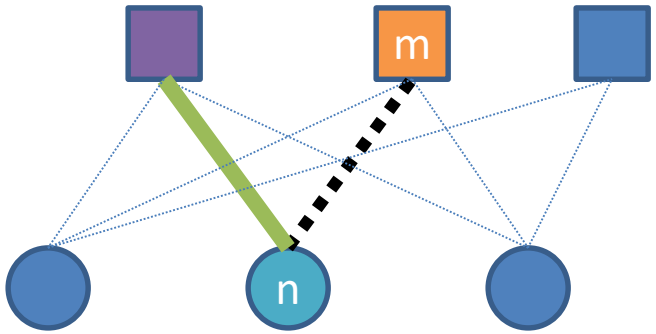
Sum-Product

From variables to factors

$$q_{n\to m}(x_n) = \prod_{m' \in M(n)\setminus m} r_{m'\to n}(x_n)$$

set of factors in which variable n *participates*

default value of 1 if empty product



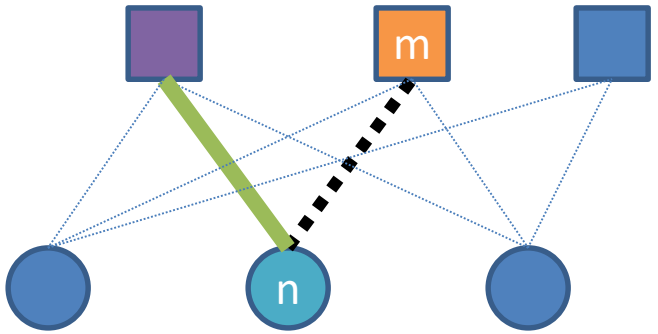
Sum-Product

From variables to factors

$$q_{n\to m}(x_n) = \prod_{m' \in M(n)\setminus m} r_{m'\to n}(x_n)$$

set of factors in which variable n *participates*

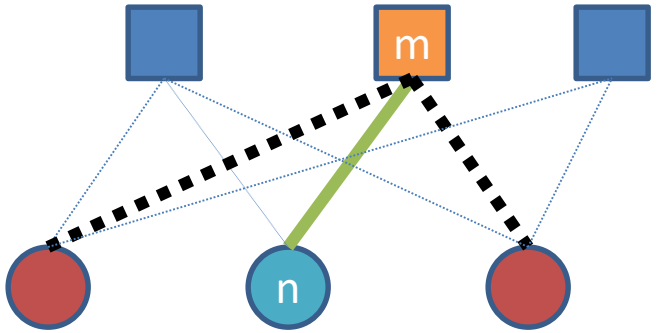
default value of 1 if empty product



From factors to variables

$$r_{m\to n}(x_n) =$$

m aggregates information from the rest of its graph via its neighbors



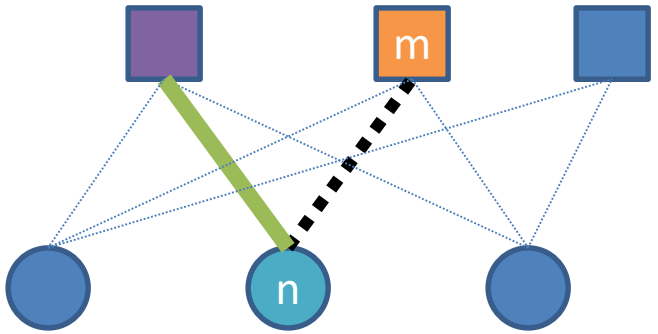
Sum-Product

From variables to factors

$$q_{n\to m}(x_n) = \prod_{m' \in M(n)\setminus m} r_{m'\to n}(x_n)$$

set of factors in which variable n *participates*

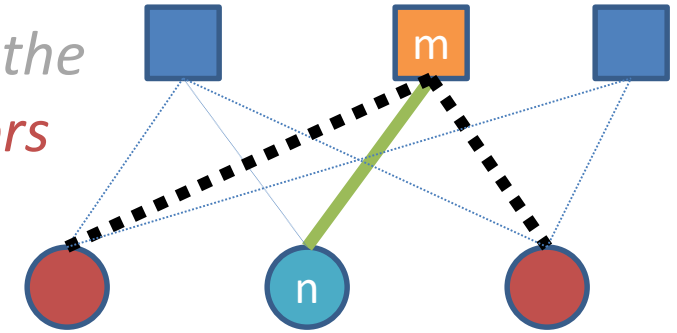
default value of 1 if empty product



From factors to variables

$$r_{m\to n}(x_n) =$$

m aggregates information from the rest of its graph via its neighbors



But these neighbors are R.V.s that take on different values

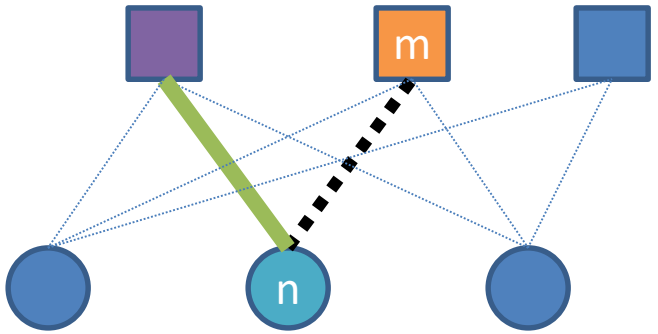
Sum-Product

From variables to factors

$$q_{n\to m}(x_n) = \prod_{m' \in M(n)\setminus m} r_{m'\to n}(x_n)$$

set of factors in which variable n participates

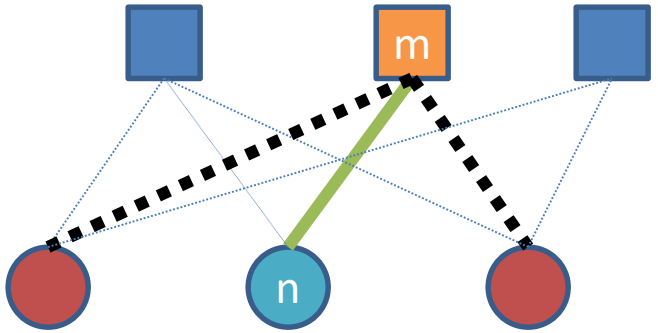
default value of 1 if empty product



From factors to variables

$$r_{m\to n}(x_n) =$$

1. *sum over configuration of variables for the m^{th} factor, with variable n fixed*
2. *aggregate info those other variables provide about the rest of the graph*



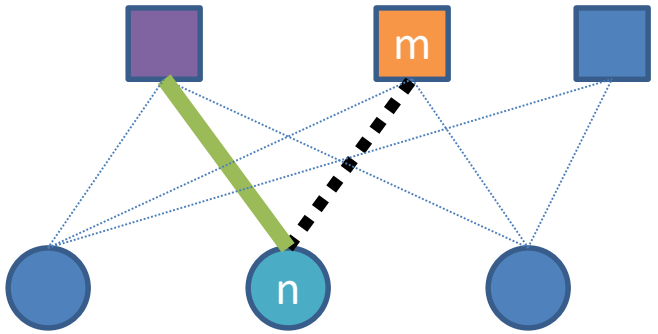
Sum-Product

From variables to factors

$$q_{n\to m}(x_n) = \prod_{m' \in M(n)\setminus m} r_{m'\to n}(x_n)$$

set of factors in which variable n participates

default value of 1 if empty product

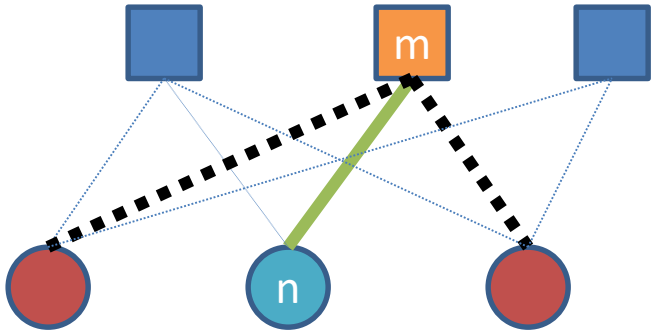


From factors to variables

$$r_{m\to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m)\setminus n} q_{n'\to m}(x_{n'})$$

1. sum over configuration of variables for the m^{th} factor, with variable n fixed

2. aggregate info those other variables provide about the rest of the graph

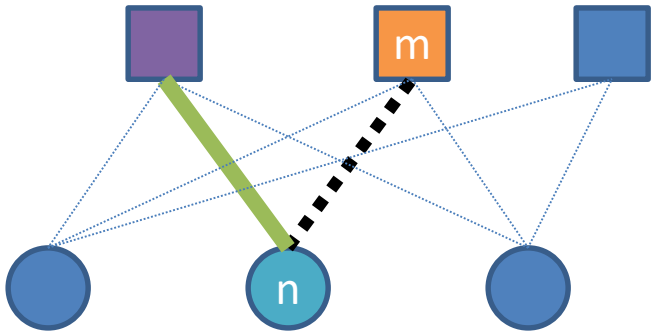


Sum-Product

From variables to factors

$$q_{n\to m}(x_n) = \prod_{m' \in M(n)\setminus m} r_{m'\to n}(x_n)$$

set of factors in which variable n participates



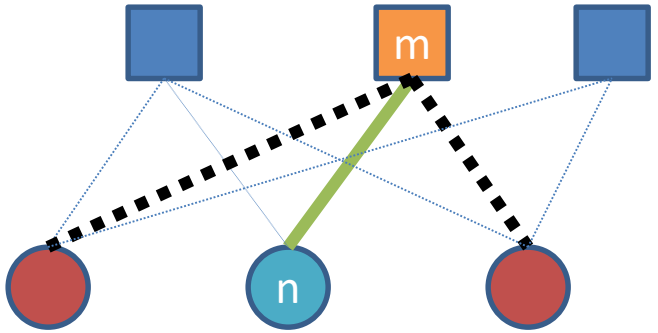
default value of 1 if empty product

From factors to variables

$$r_{m\to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m)\setminus n} q_{n'\to m}(x_{n'})$$

sum over configuration of variables for the m[th] factor, with variable n fixed

set of variables that the m[th] factor depends on



Meaning of the Computed Values

From variables to factors

$$q_{n\to m}(x_n) = \prod_{m' \in M(n)\setminus m} r_{m'\to n}(x_n)$$

x_n telling factor m the "goodness" for the rest of the graph if x_n has a particular value

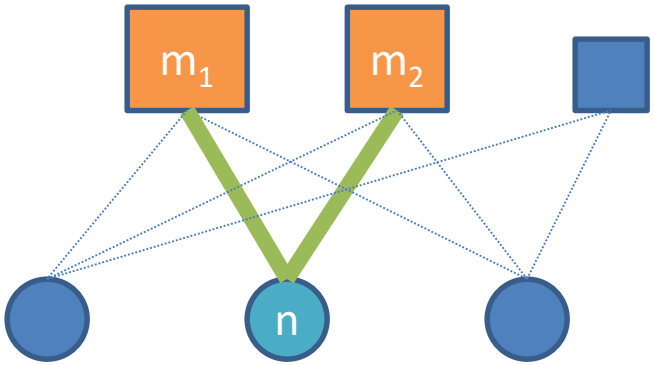
From factors to variables

$$r_{m\to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m)\setminus n} q_{n'\to m}(x_{n'})$$

factor m telling x_n the "goodness" for the rest of the graph if x_n has a particular value

From Messages to Variable *Beliefs*

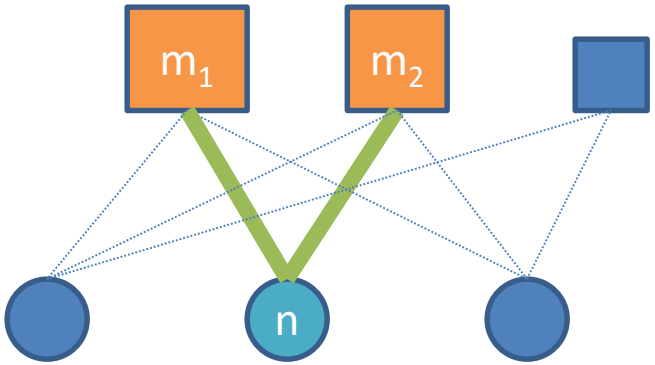
$r_{m_1 \to n}(x_n)$ tells x_n the "goodness" from m_1's perspective if x_n has a particular value



$r_{m_2 \to n}(x_n)$ tells x_n the "goodness" from m_2's perspective if x_n has a particular value

From Messages to Variable *Beliefs*

$r_{m_1 \to n}(x_n)$ tells x_n the "goodness" from m_1's perspective if x_n has a particular value

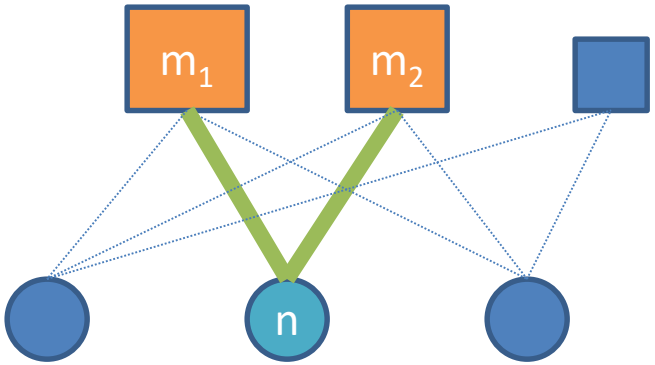


$r_{m_2 \to n}(x_n)$ tells x_n the "goodness" from m_2's perspective if x_n has a particular value

Together, they describe the cover the entire graph!

From Messages to Variable *Beliefs*

$r_{m_1 \to n}(x_n)$ tells x_n the "goodness" from m_1's perspective if x_n has a particular value



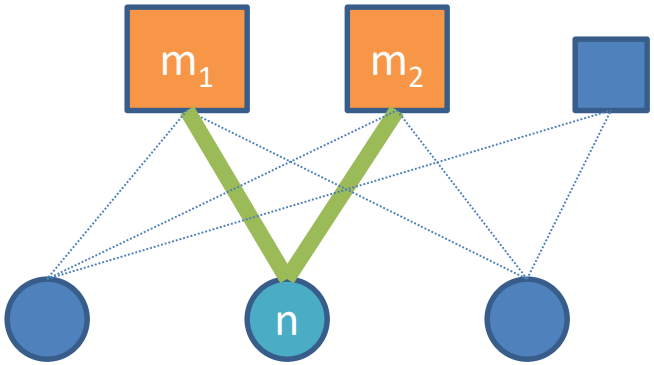
$r_{m_2 \to n}(x_n)$ tells x_n the "goodness" from m_2's perspective if x_n has a particular value

Together, they describe the cover the entire graph!

$$p(x_n = v) \propto r_{m_1 \to n}(x_n = v)\, r_{m_2 \to n}(x_n = v)$$

From Messages to Variable *Beliefs*: General Formula

$r_{m_1 \to n}(x_n)$ tells x_n the "goodness" from m_1's perspective if x_n has a particular value

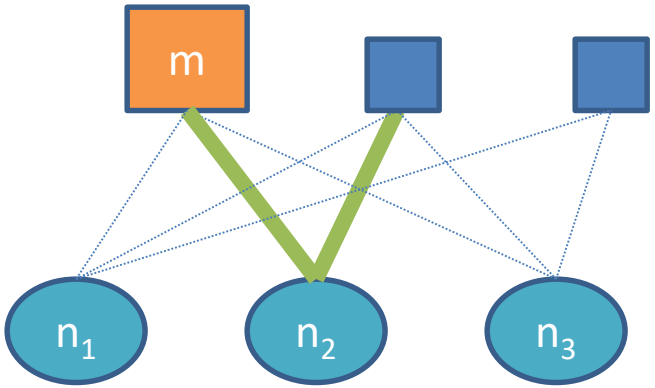


$r_{m_2 \to n}(x_n)$ tells x_n the "goodness" from m_2's perspective if x_n has a particular value

$$p(x_n = v) \propto \prod_{m \in N(x_n)} r_{m \to n}(x_n = v)$$

From Messages to Factor *Beliefs*: General Formula

$q_{n_i \to m}$ tells m the "goodness" from x_{n_i}'s perspective if it has a particular value



$$p(x_{\{m\}} = \boldsymbol{v}) \propto m(x_{\{m\}} = \boldsymbol{v}) \prod_{x_{n_i} \in N(m)} q_{n_i \to m}(x_{n_i} = v_i)$$

How to Use these Messages

1. Select the root, or pick one if a tree
   a) Send messages from leaves to root
   b) Send messages from root to leaves
   c) Use messages to compute (unnormalized) marginal probabilities
2. Are we done?
   a) If a tree structure, we've converged
   b) If not:
      i. Either accept the partially converged result, or…
      ii. Go back to (1) and repeat

How to Use these Messages

Compute Marginals/Normalizer

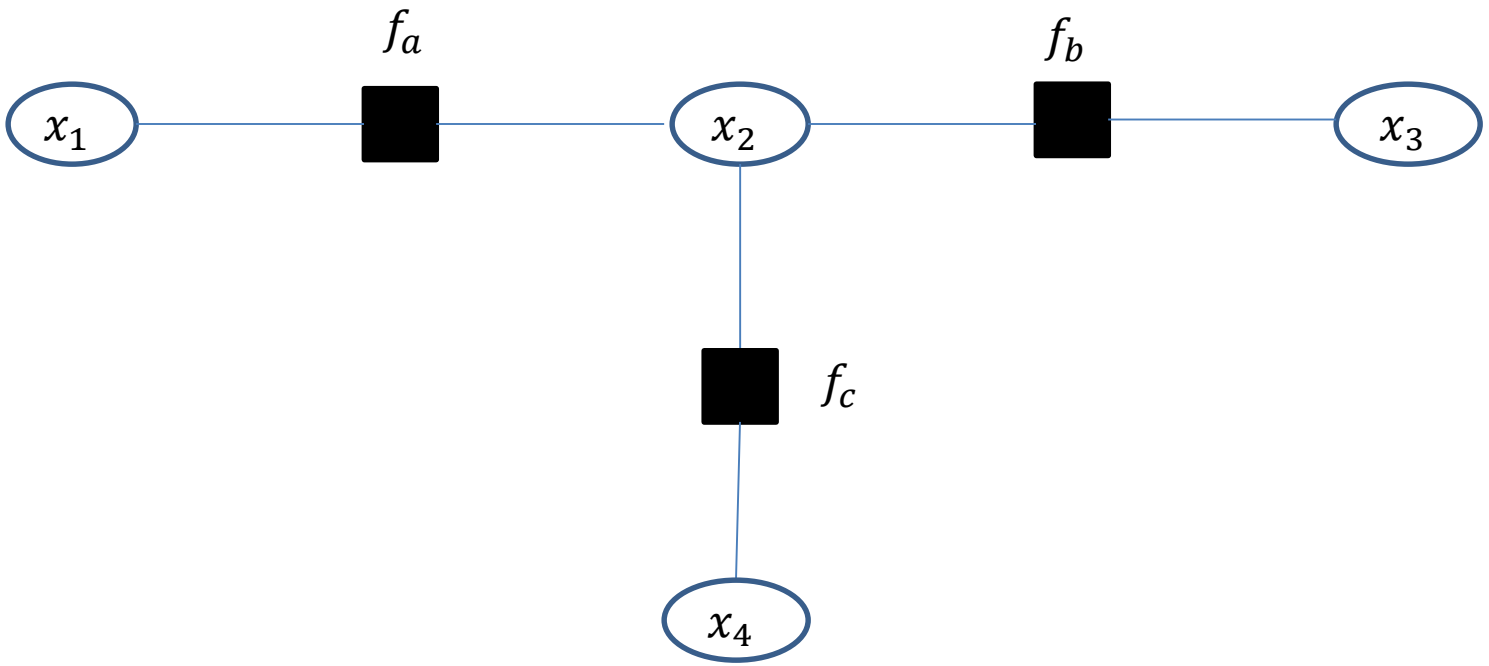
1. Select the root, or pick one if a tree
   a) Send messages from leaves to root
   b) Send messages from root to leaves
   c) Use messages to compute (unnormalized) marginal probabilities
2. Are we done?
   a) If a tree structure, we've converged
   b) If not:
      i. Either accept the partially converged result, or…
      ii. Go back to (1) and repeat

For Learning/Inference

Whenever you need to compute a likelihood, marginal probability, or a model-specific expectation, run this algorithm to compute the necessary probabilities

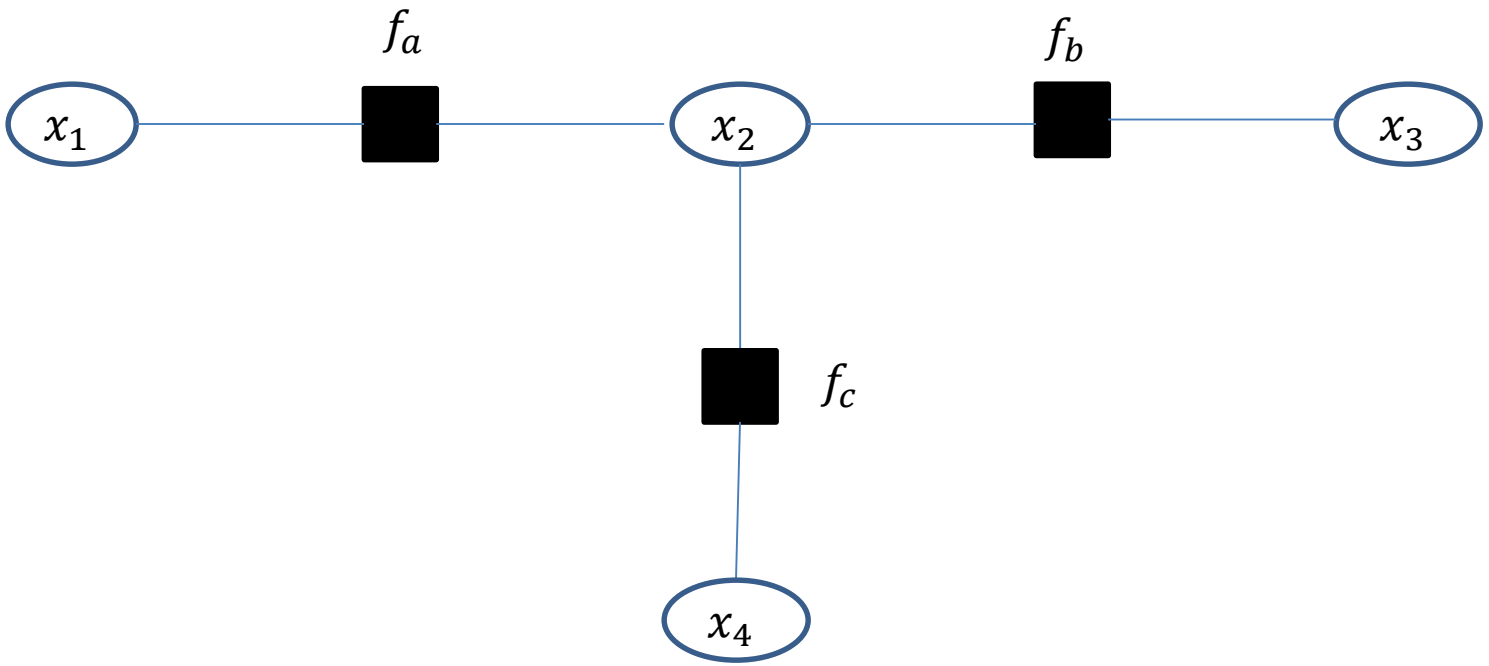
- Prediction:
  - Of a sequence $p(z_1, \dots, z_N | w_{1:N})$
  - Of an individual tag $p(z_i | w_{1:N})$
- Marginal (if appropriate)
  - $p(w_{1:N})$
- Learning model parameters
  - EM
  - Variational inference
  - …

Example



Q: What are the variables?

Example

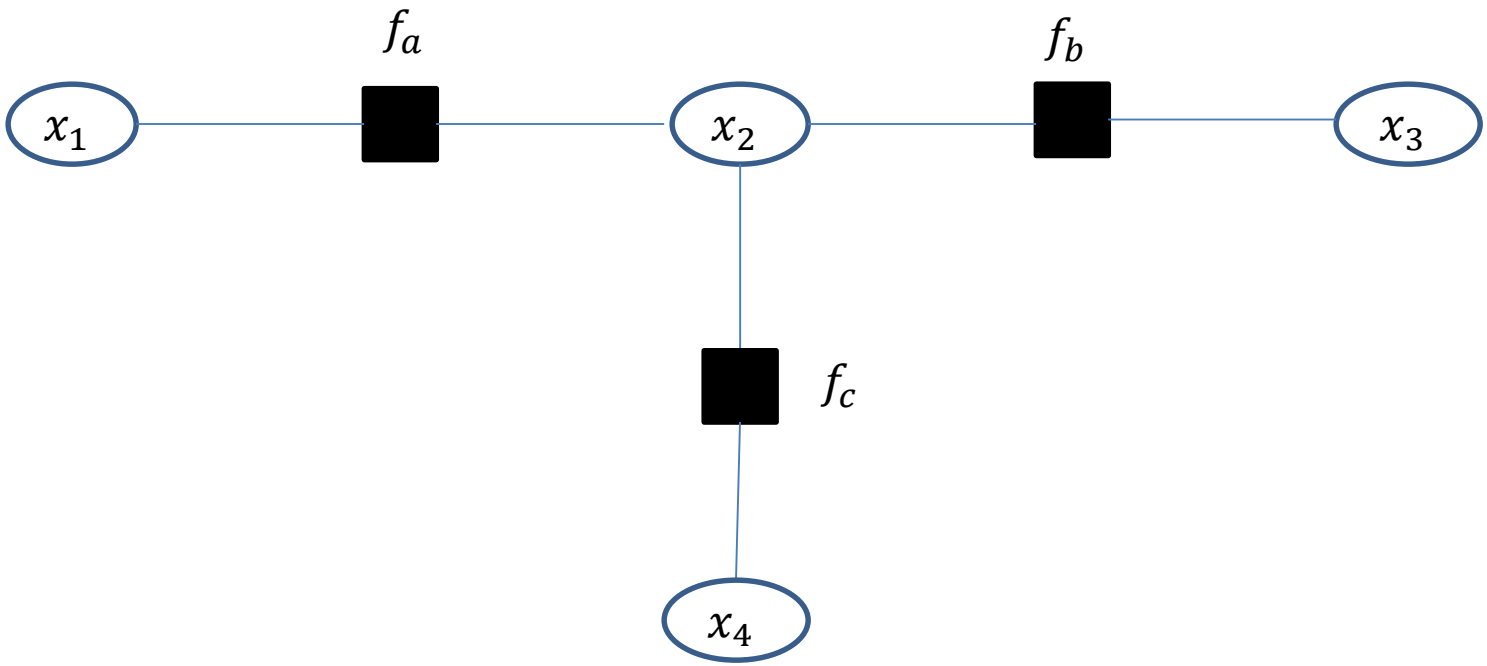


Q: What are the variables?

A: x_1, x_2, x_3, x_4

Q: What are the factors?

Example



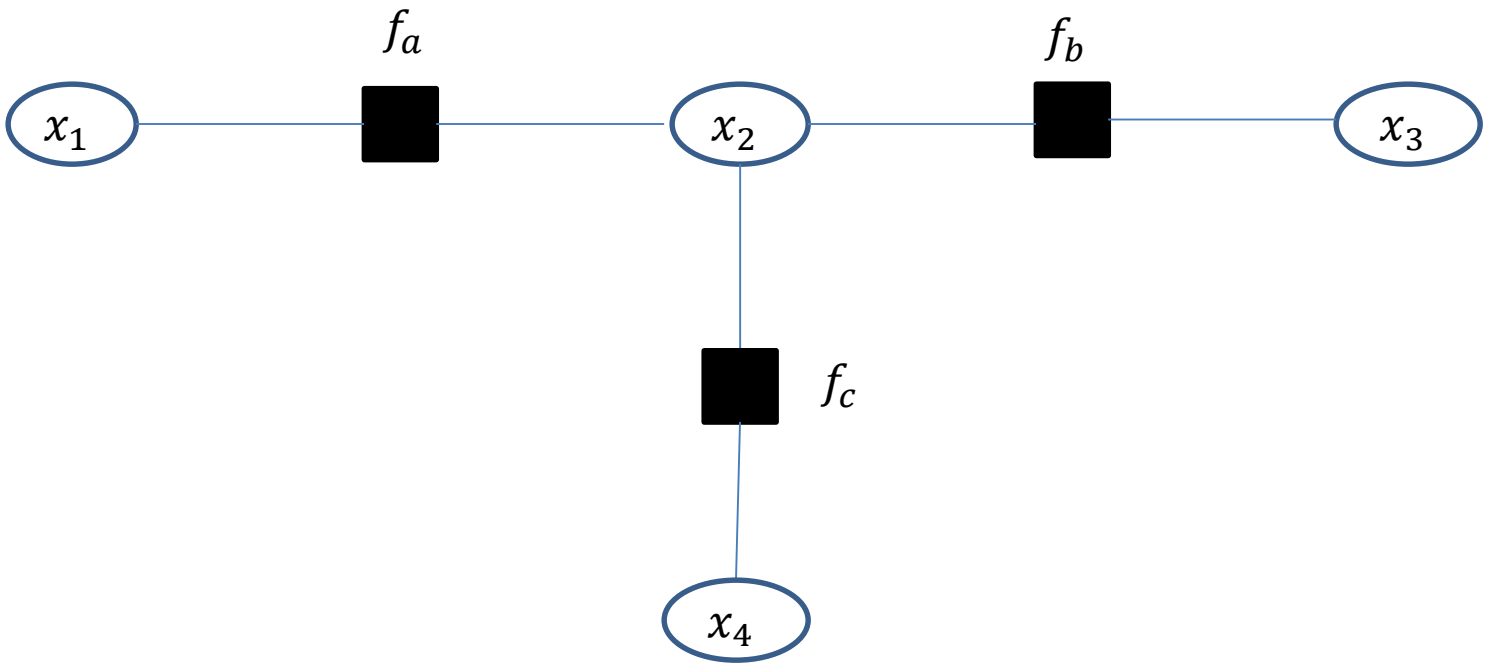
Q: What are the variables?

A: x_1, x_2, x_3, x_4

Q: What are the factors?

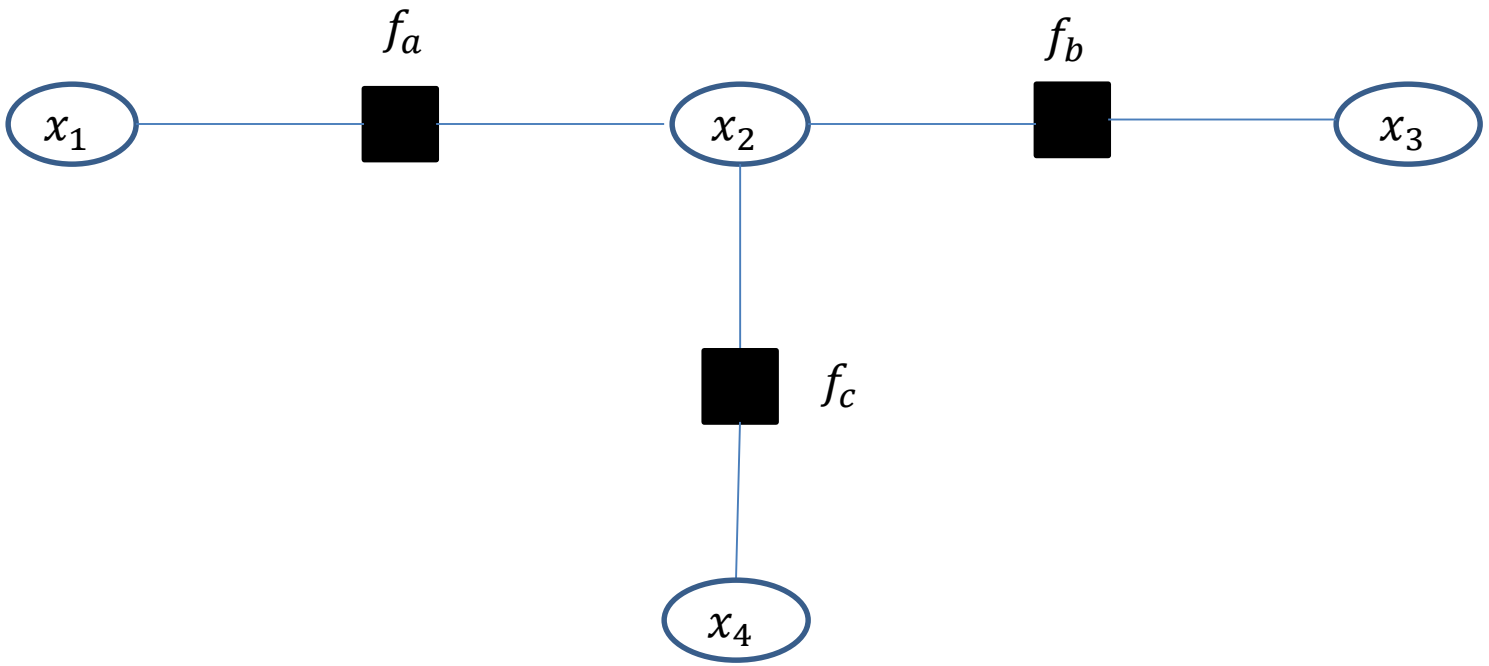
A: $f_a(x_1, x_2)$, $f_b(x_2, x_3)$, $f_c(x_2, x_4)$

Example



Q: What is the distribution we're modeling?

Example

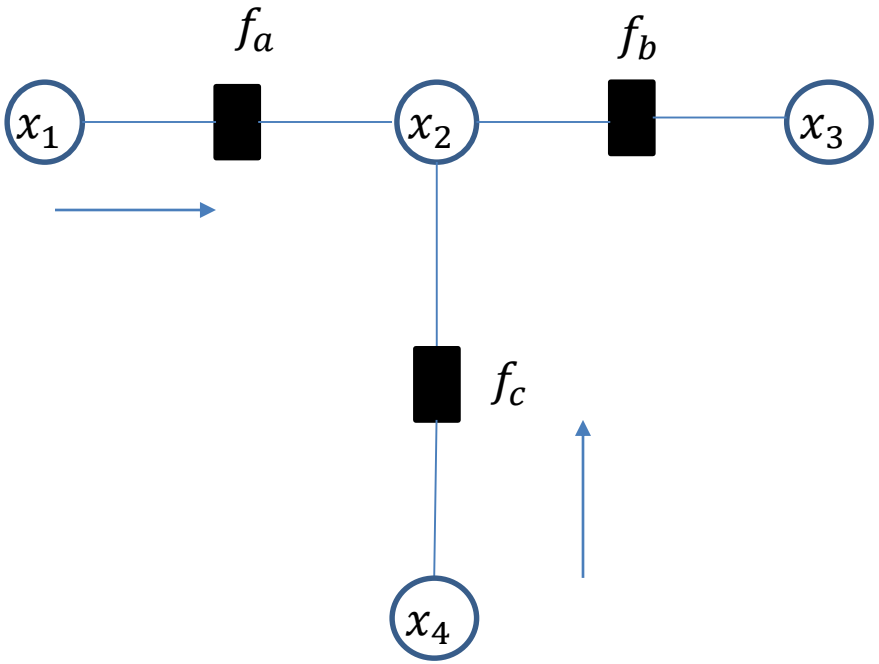


Q: What is the distribution we're modeling?

A:

$$p(x_1, x_2, x_3, x_4) = f_a(x_1, x_2)\, f_b(x_2, x_3)\, f_c(x_2, x_4)$$

Example



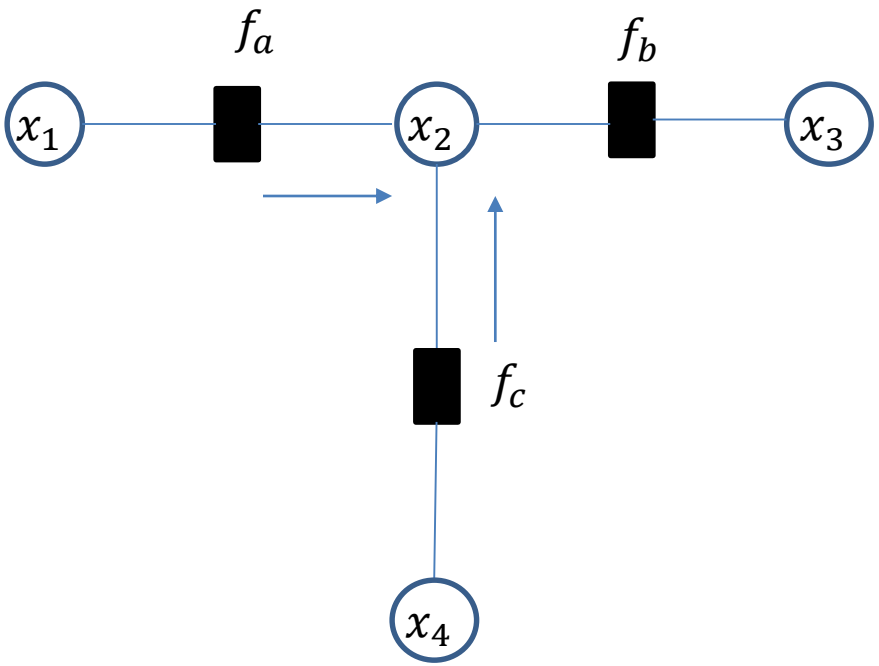
1. Select the root, or pick one if a tree (x_3)
    1. Send messages from leaves to root

$$q_{x_1 \to f_a}(x_1) = 1$$
$$q_{x_4 \to f_c}(x_4) = 1$$

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example

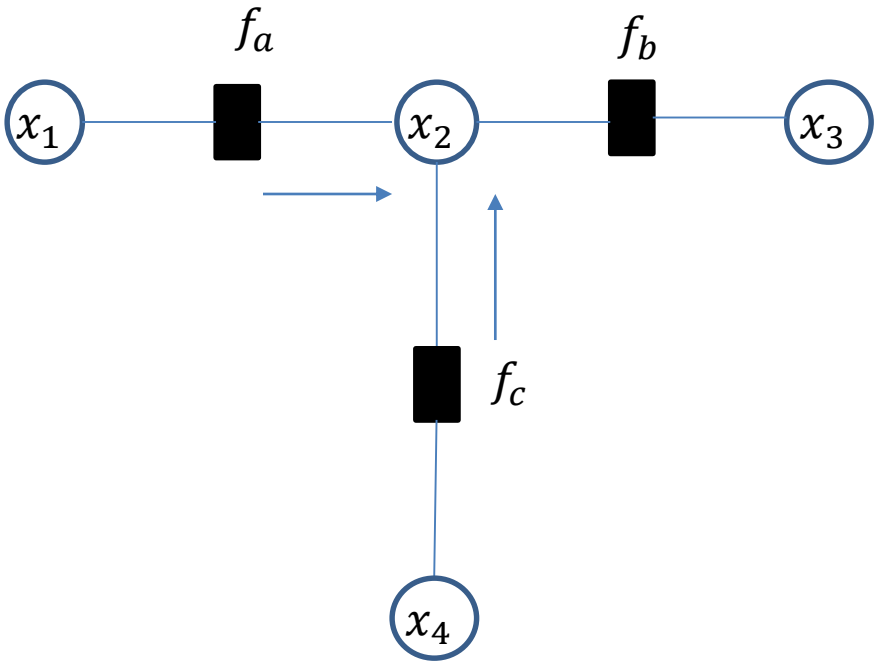


1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root

$$q_{x_1 \to f_a}(x_1) = 1$$
$$q_{x_4 \to f_c}(x_4) = 1$$
$$r_{f_a \to x_2}(x_2) = ???$$

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n) \qquad r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root

$$q_{x_1 \to f_a}(x_1) = 1$$

$$q_{x_4 \to f_c}(x_4) = 1$$

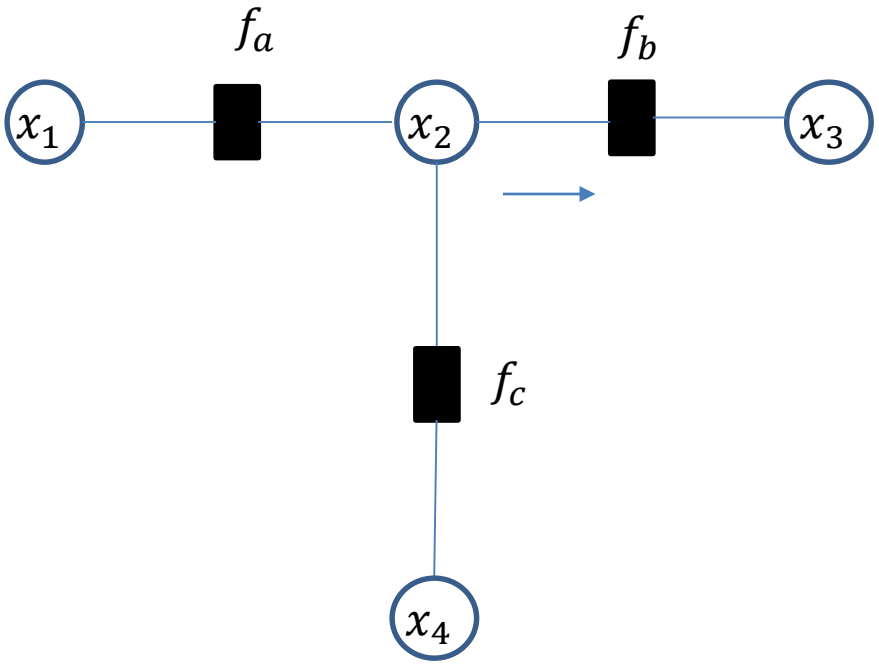
$$r_{f_a \to x_2}(x_2) = \sum_k f_a(x_1 = k, x_2)$$

$$r_{f_c \to x_2}(x_2) = \sum_k f_a(x_2, x_4 = k)$$

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root

$$q_{x_1 \to f_a}(x_1) = 1$$

$$q_{x_4 \to f_c}(x_4) = 1$$

$$r_{f_a \to x_2}(x_2) = \sum_k f_a(x_1 = k, x_2)$$

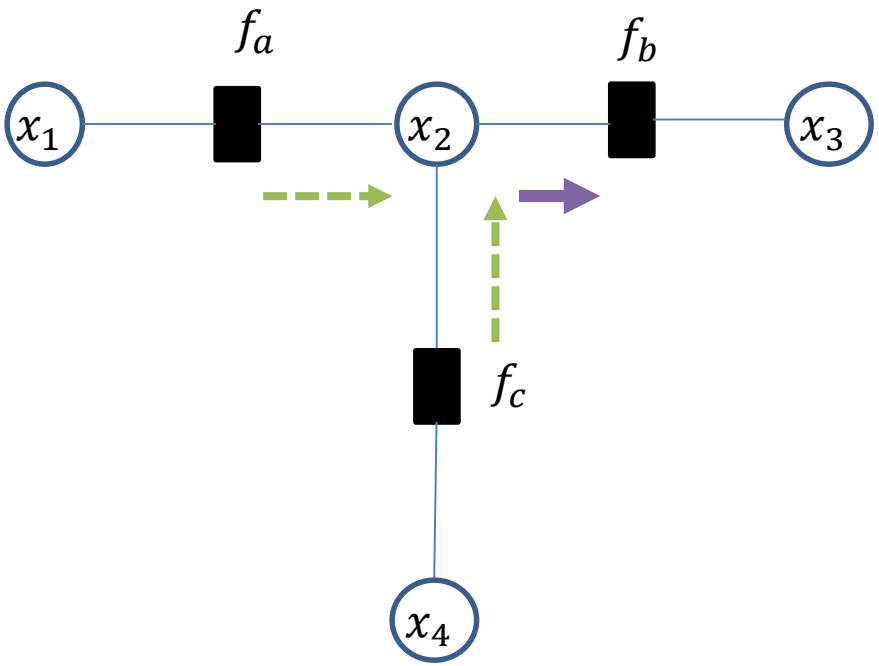
$$r_{f_c \to x_2}(x_2) = \sum_k f_a(x_2, x_4 = k)$$

$$q_{x_2 \to f_b}(x_2) = ???$$

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root

$$q_{x_1 \to f_a}(x_1) = 1$$

$$q_{x_4 \to f_c}(x_4) = 1$$

$$r_{f_a \to x_2}(x_2) = \sum_k f_a(x_1 = k, x_2)$$

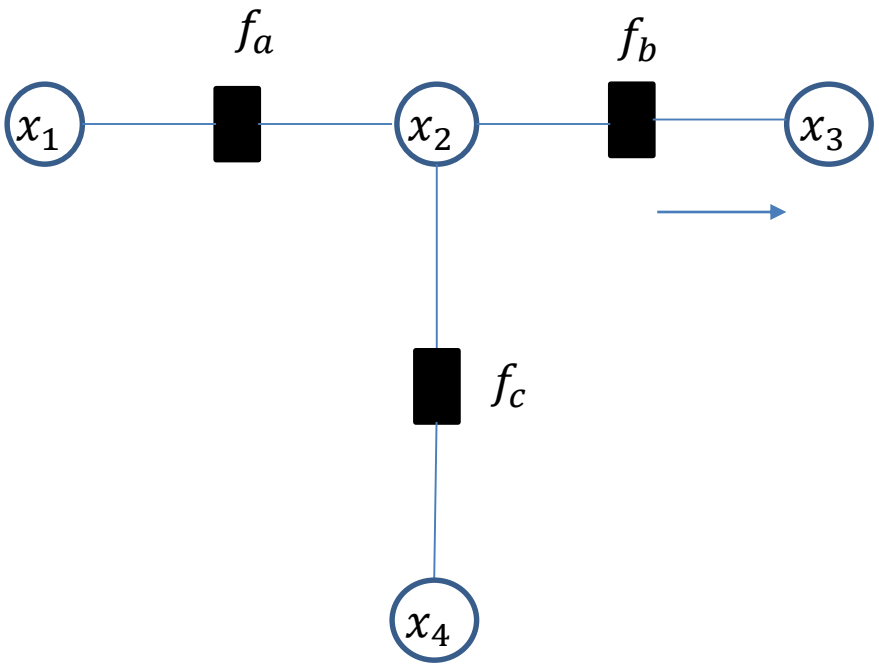
$$r_{f_c \to x_2}(x_2) = \sum_k f_a(x_2, x_4 = k)$$

$$q_{x_2 \to f_b}(x_2) = r_{f_a \to x_2}(x_2) r_{f_c \to x_2}(x_2)$$

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root

$$q_{x_1 \to f_a}(x_1) = 1$$

$$q_{x_4 \to f_c}(x_4) = 1$$

$$r_{f_a \to x_2}(x_2) = \sum_k f_a(x_1 = k, x_2)$$

$$r_{f_c \to x_2}(x_2) = \sum_k f_a(x_2, x_4 = k)$$

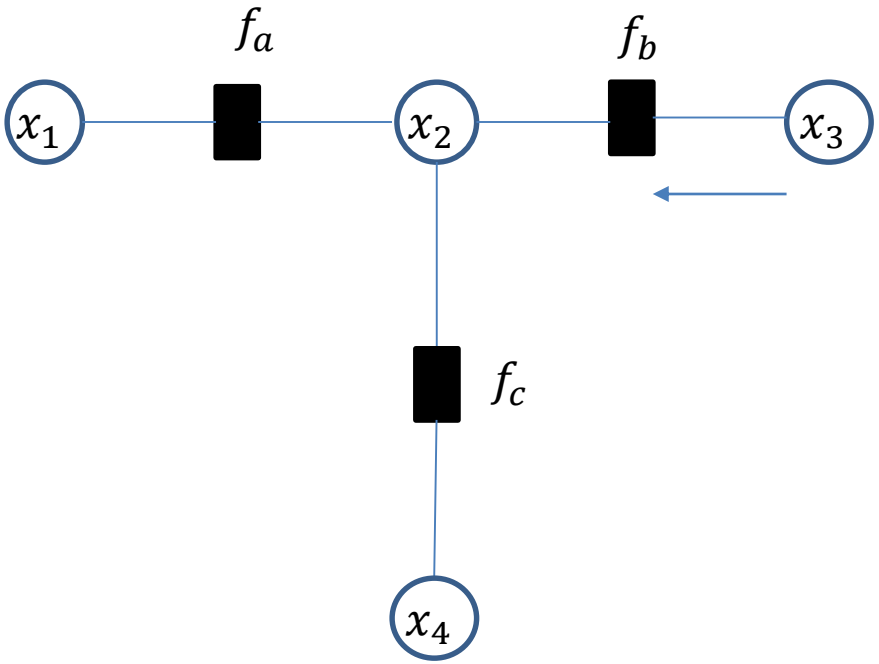
$$q_{x_2 \to f_b}(x_2) = r_{f_a \to x_2}(x_2) r_{f_c \to x_2}(x_2)$$

$$r_{f_b \to x_3}(x_3) = \sum_k f_b(x_2 = k, x_3)$$

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example



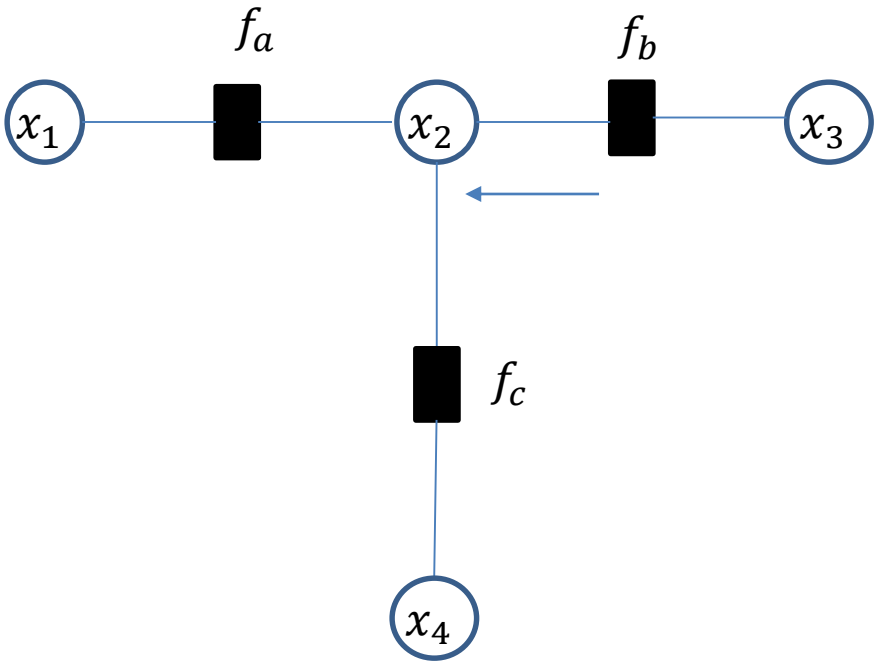
1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root
   2. Send messages from root to leaves

$$q_{x_3 \to f_b}(x_3) = 1$$

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root
   2. Send messages from root to leaves

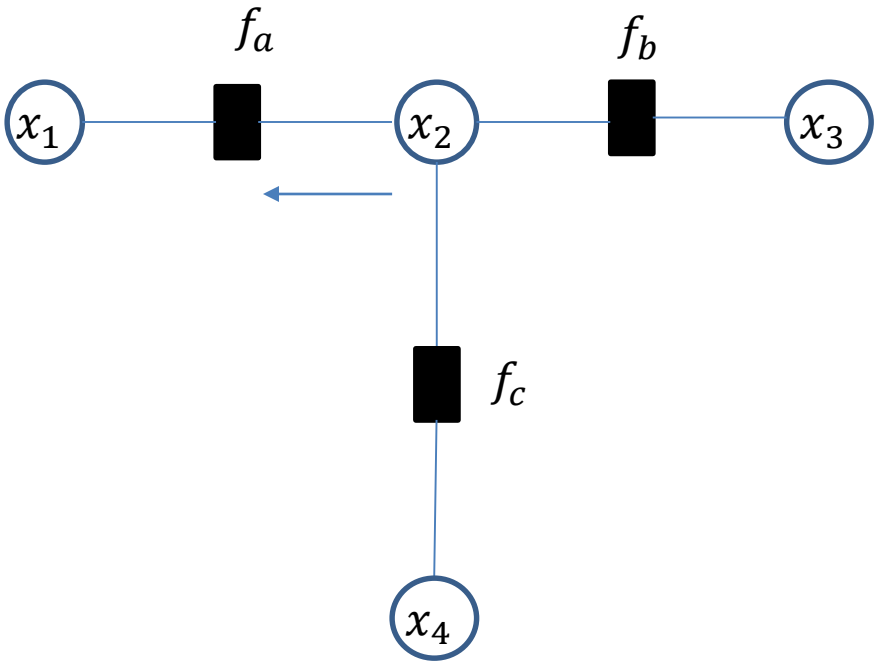
$$q_{x_3 \to f_b}(x_3) = 1$$

$$r_{f_b \to x_2}(x_2) = \sum_k f_b(x_2, x_3 = k)$$

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root
   2. Send messages from root to leaves

$$q_{x_3 \to f_b}(x_3) = 1$$

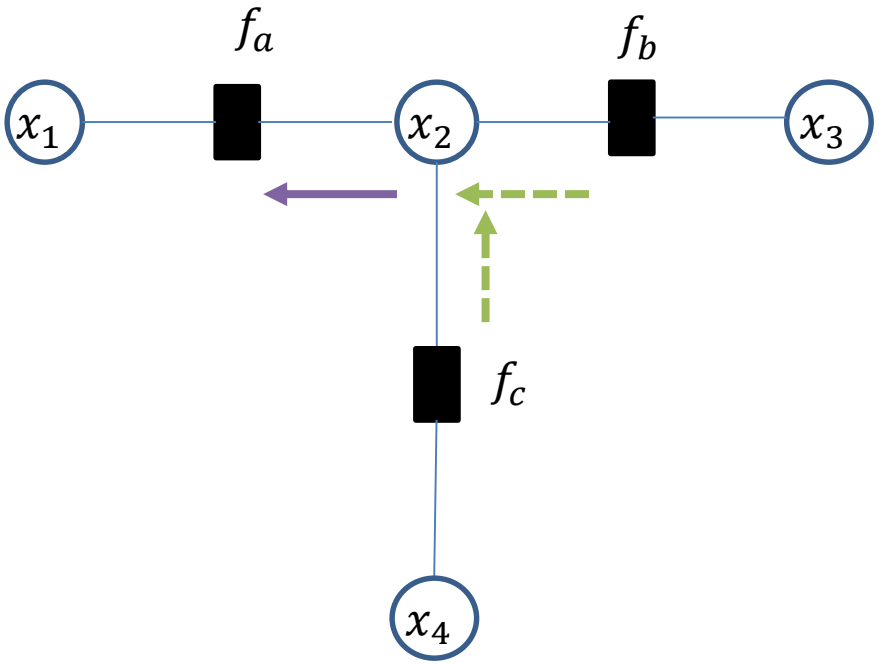
$$r_{f_b \to x_2}(x_2) = \sum_k f_b(x_2, x_3 = k)$$

$$q_{x_2 \to f_a}(x_2) = \; ???$$

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root
   2. Send messages from root to leaves

$$q_{x_3 \to f_b}(x_3) = 1$$

$$r_{f_b \to x_2}(x_2) = \sum_k f_b(x_2, x_3 = k)$$

$$q_{x_2 \to f_a}(x_2) = r_{f_b \to x_2}(x_2) r_{f_c \to x_2}(x_2)$$

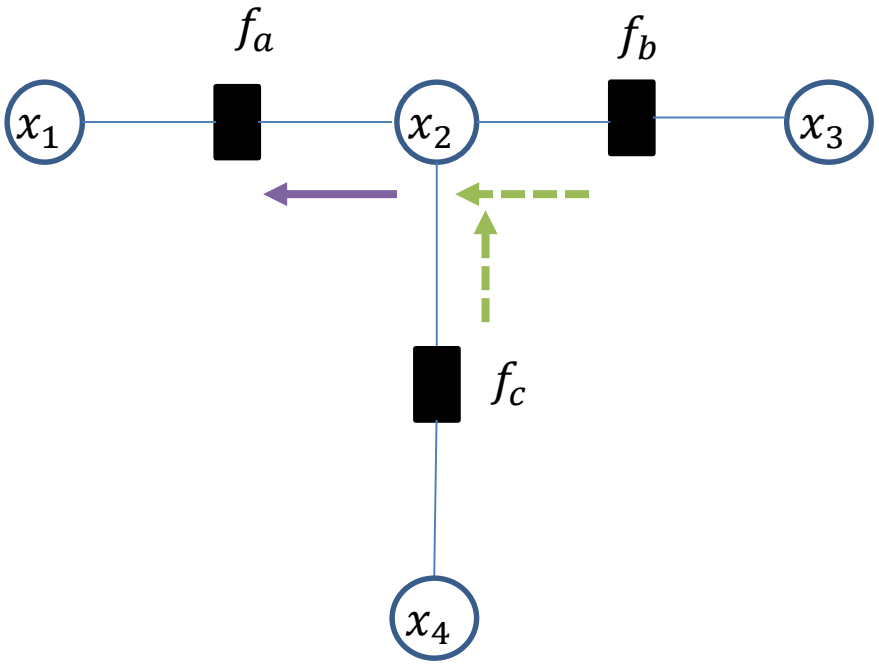
We just computed this

Q: Where did we compute this?

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root
   2. Send messages from root to leaves

$$q_{x_3 \to f_b}(x_3) = 1$$

$$r_{f_b \to x_2}(x_2) = \sum_k f_b(x_2, x_3 = k)$$

$$q_{x_2 \to f_a}(x_2) = r_{f_b \to x_2}(x_2) r_{f_c \to x_2}(x_2)$$

We just computed this

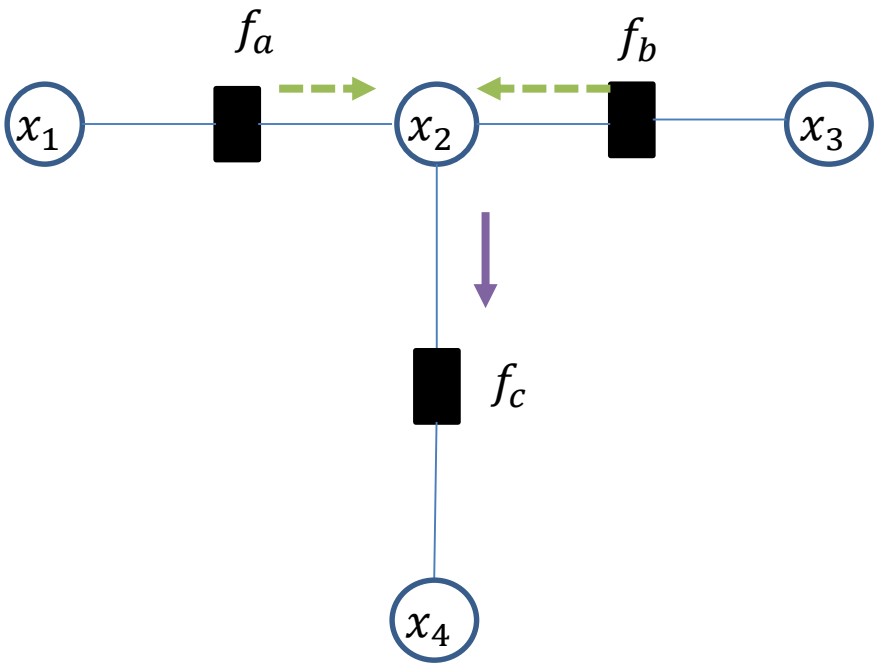
Q: Where did we compute this?

A: In step 1 (leaves → root)

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root
   2. Send messages from root to leaves

$$q_{x_3 \to f_b}(x_3) = 1$$

$$r_{f_b \to x_2}(x_2) = \sum_k f_b(x_2, x_3 = k)$$

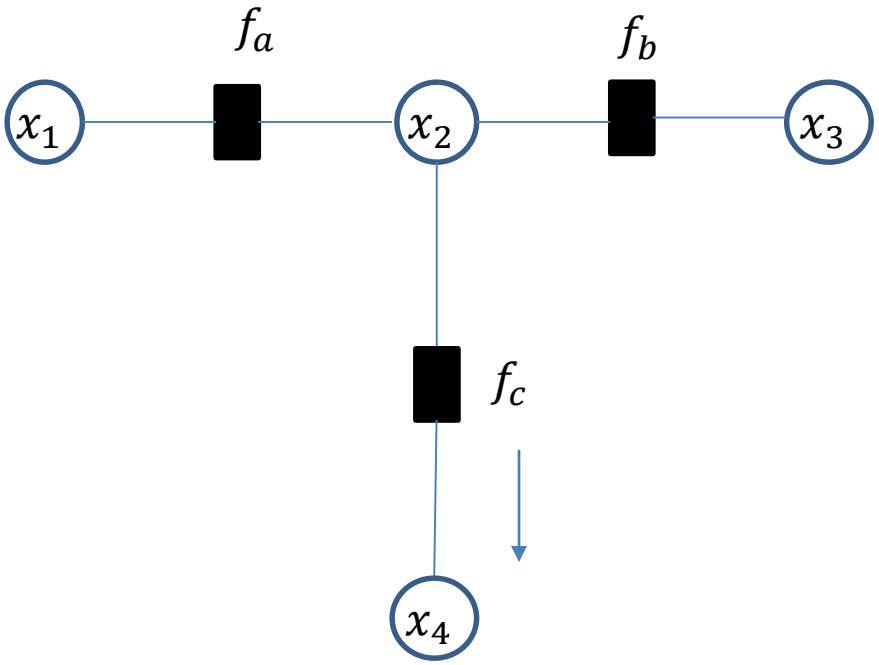
$$q_{x_2 \to f_a}(x_2) = r_{f_b \to x_2}(x_2) r_{f_c \to x_2}(x_2)$$

$$q_{x_2 \to f_c}(x_2) = r_{f_a \to x_2}(x_2) r_{f_b \to x_2}(x_2)$$

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root
   2. Send messages from root to leaves

$$q_{x_3 \to f_b}(x_3) = 1$$

$$r_{f_b \to x_2}(x_2) = \sum_k f_b(x_2, x_3 = k)$$

$$q_{x_2 \to f_a}(x_2) = r_{f_b \to x_2}(x_2) r_{f_c \to x_2}(x_2)$$

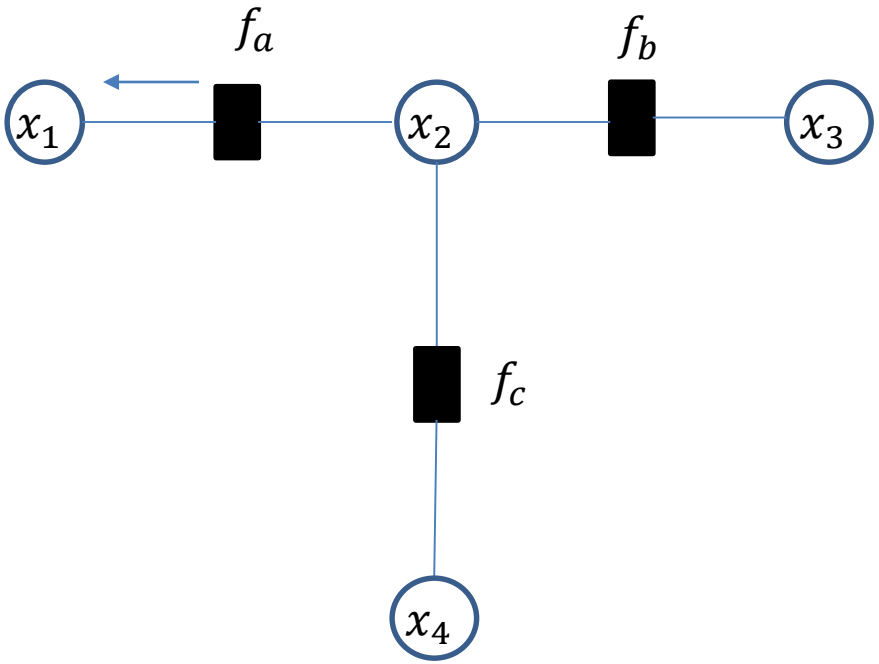
$$q_{x_2 \to f_c}(x_2) = r_{f_a \to x_2}(x_2) r_{f_b \to x_2}(x_2)$$

$$r_{f_c \to x_4}(x_4) = \sum_k f_c(x_2 = k, x_4)$$

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root
   2. Send messages from root to leaves

$$q_{x_3 \to f_b}(x_3) = 1$$

$$r_{f_b \to x_2}(x_2) = \sum_k f_b(x_2, x_3 = k)$$

$$q_{x_2 \to f_a}(x_2) = r_{f_b \to x_2}(x_2) r_{f_c \to x_2}(x_2)$$

$$q_{x_2 \to f_c}(x_2) = r_{f_a \to x_2}(x_2) r_{f_b \to x_2}(x_2)$$

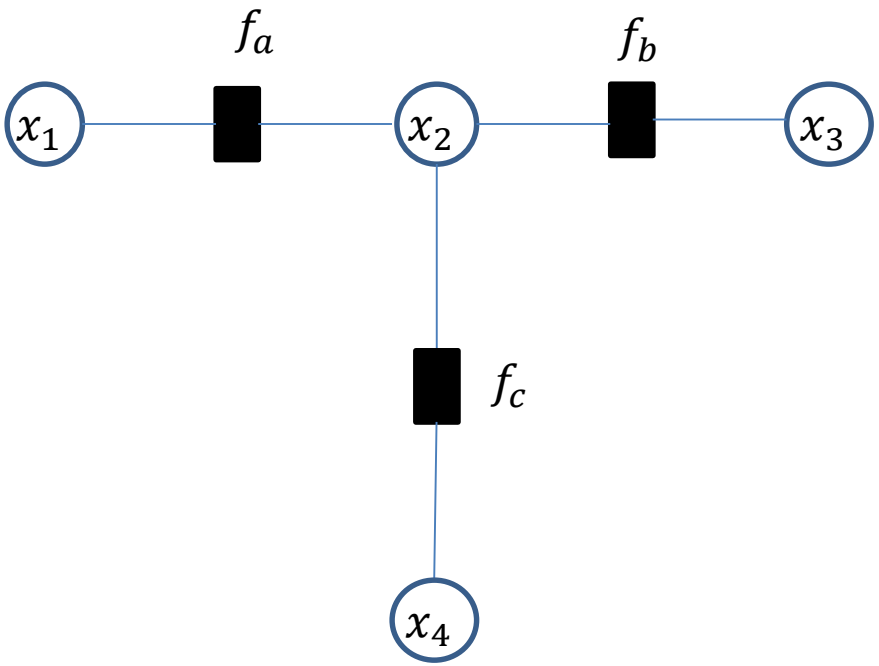
$$r_{f_c \to x_4}(x_4) = \sum_k f_c(x_2 = k, x_4)$$

$$r_{f_a \to x_1}(x_1) = \sum_k f_a(x_1, x_2 = k)$$

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example



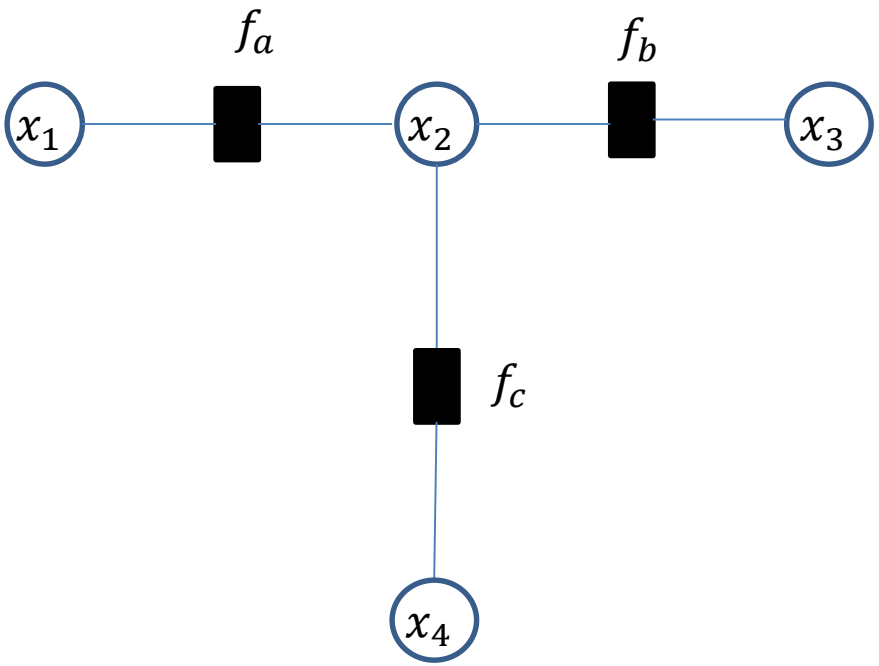
1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root
   2. Send messages from root to leaves
   3. Use messages to compute marginal probabilities

$$p(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root
   2. Send messages from root to leaves
   3. Use messages to compute marginal probabilities

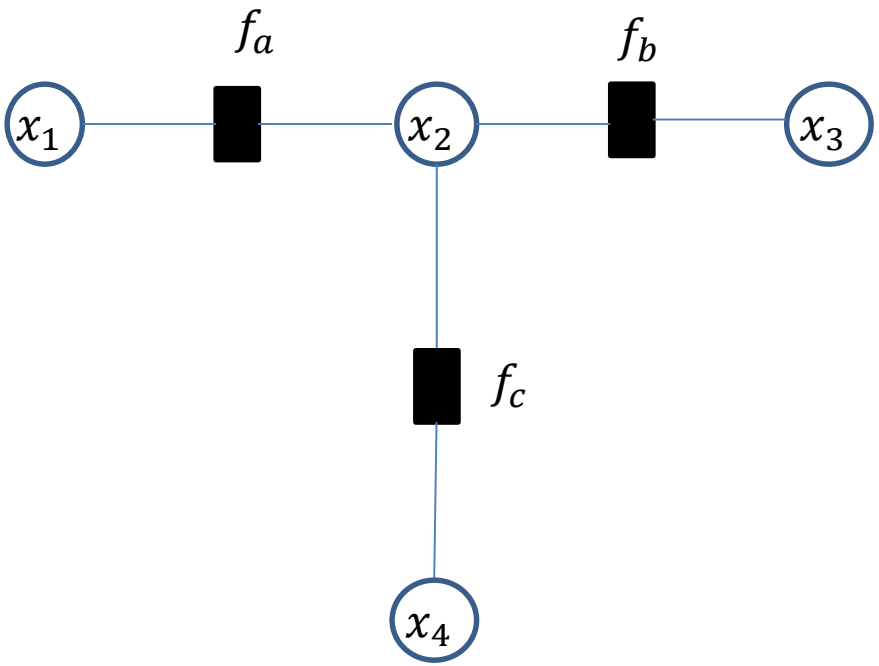
$$p(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$p(x_1) = r_{f_a \to x_1}(x_1)$$

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root
   2. Send messages from root to leaves
   3. Use messages to compute marginal probabilities

$$p(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

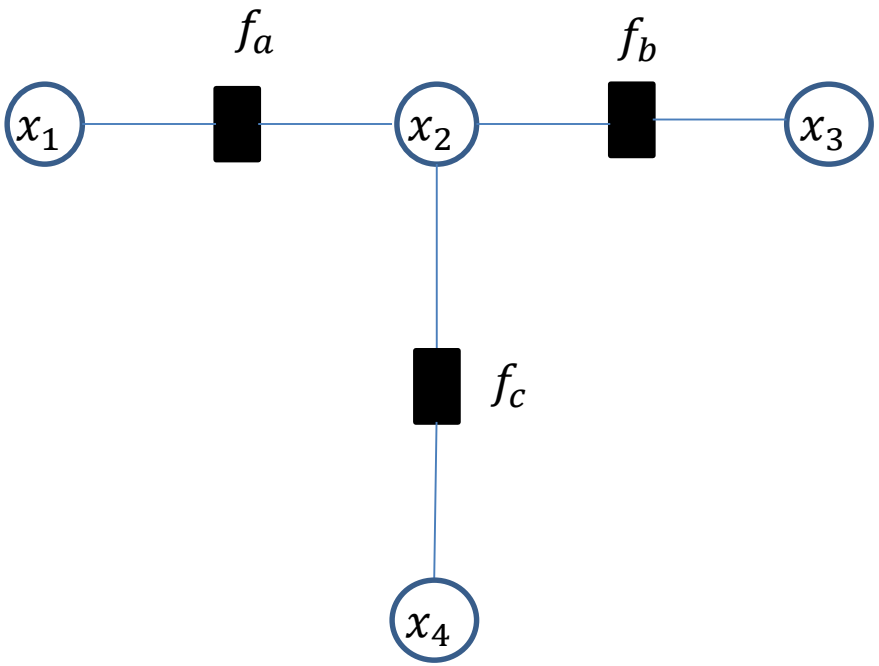
$$p(x_1) = r_{f_a \to x_1}(x_1)$$

$$p(x_2) = r_{f_a \to x_2}(x_2) r_{f_b \to x_2}(x_2) r_{f_c \to x_2}(x_2)$$

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
   1. Send messages from leaves to root
   2. Send messages from root to leaves
   3. Use messages to compute marginal probabilities

$$p(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$p(x_1) = r_{f_a \to x_1}(x_1)$$

$$p(x_2) = r_{f_a \to x_2}(x_2) r_{f_b \to x_2}(x_2) r_{f_c \to x_2}(x_2)$$

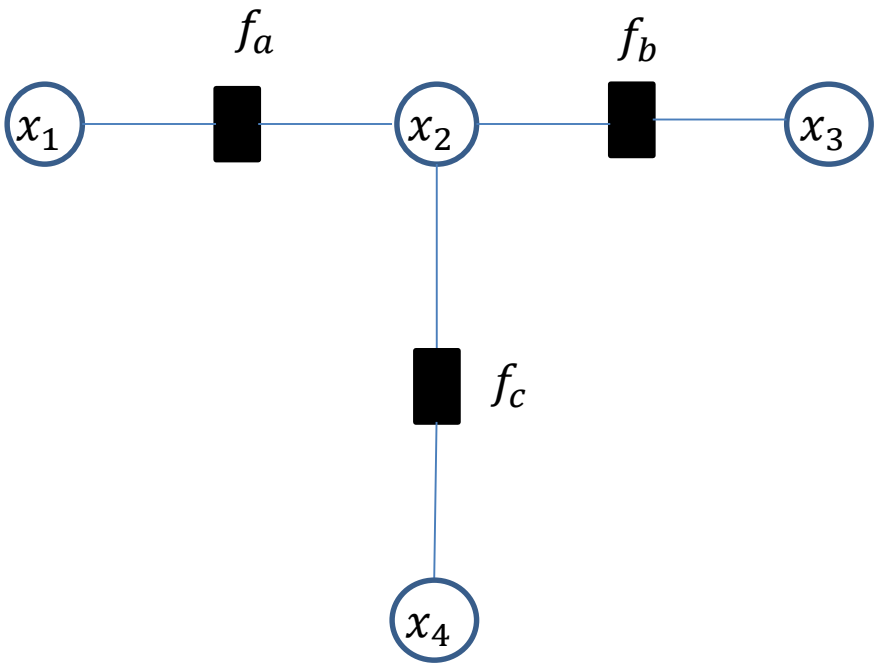
$$p(x_3) = r_{f_b \to x_3}(x_3)$$

$$p(x_4) = r_{f_c \to x_4}(x_4)$$

$$q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)$$

$$r_{m \to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})$$

Example

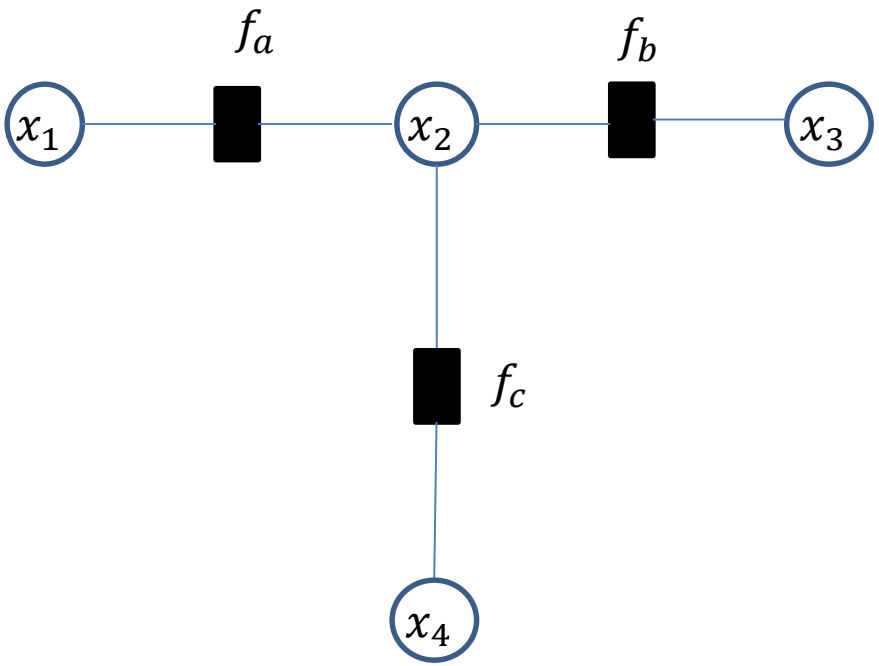


1. Select the root, or pick one if a tree (x_3)
    1. Send messages from leaves to root
    2. Send messages from root to leaves
    3. Use messages to compute marginal probabilities
2. Are we done?
    1. If a tree structure, we've converged
    2.

$$q_{n\to m}(x_n) = \prod_{m' \in M(n)\setminus m} r_{m'\to n}(x_n)$$

$$r_{m\to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m)\setminus n} q_{n'\to m}(x_{n'})$$

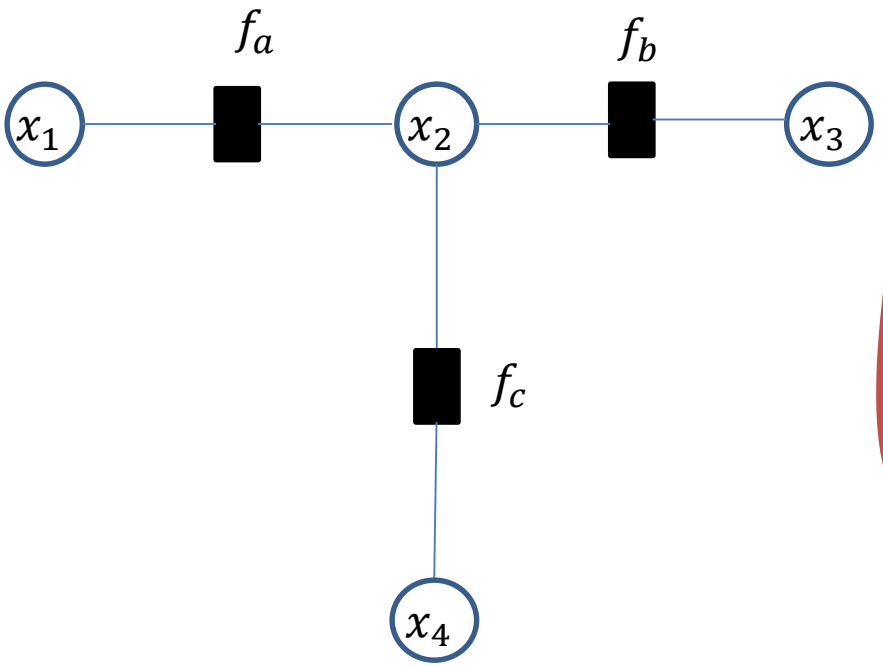
Example



1. Select the root, or pick one if a tree (x_3)
    1. Send messages from leaves to root
    2. Send messages from root to leaves
    3. Use messages to compute marginal probabilities
2. Are we done?
    1. If a tree structure, we've converged
    2. If not:
        1. Either accept the partially converged result, or…
        2.

$$q_{n\to m}(x_n) = \prod_{m' \in M(n)\backslash m} r_{m'\to n}(x_n) \qquad r_{m\to n}(x_n) = \sum_{\boldsymbol{w}_m \backslash n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m)\backslash n} q_{n'\to m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree
    1. Send messages from leaves to root
    2. Send messages from root to leaves
    3. Use messages to compute marginal probabilities
2. Are we done?
    1. If a tree structure, we've converged
    2. If not:
        1. Either accept the partially converged result, or…
        2. Go back to (1) and repeat

[Loopy BP]

$$q_{n\to m}(x_n) = \prod_{m' \in M(n)\setminus m} r_{m'\to n}(x_n)$$

$$r_{m\to n}(x_n) = \sum_{\boldsymbol{w}_m \setminus n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m)\setminus n} q_{n'\to m}(x_{n'})$$

Max-Product (Max-Sum)

Problem: how to find the most likely (best) setting of latent variables

Replace sum (+) with max in factor→variable computations

$$r_{m \to n}(x_n) = \max_{\boldsymbol{w}_m \backslash n} f_m(\boldsymbol{w}_m) \prod_{n' \in N(m) \backslash n} q_{n' \to m}(x_{n'})$$

(why max-*sum*? computationally, implement with logs)

Loopy Belief Propagation

Sum-product algorithm is not exact for general graphs

Loopy Belief Propagation (Loopy BP): run sum-product algorithm *anyway* and hope for the best

Requires a **message passing schedule**

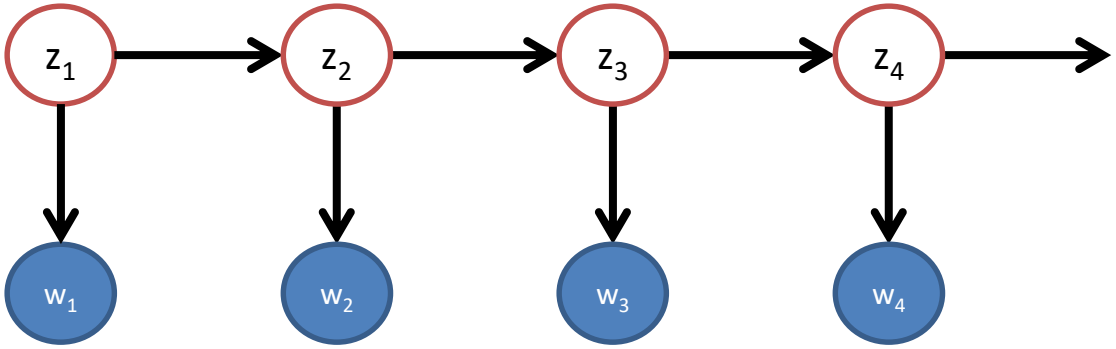
Outline

Message Passing: Graphical Model Inference

Example: Linear Chain CRF

Example: Linear Chain

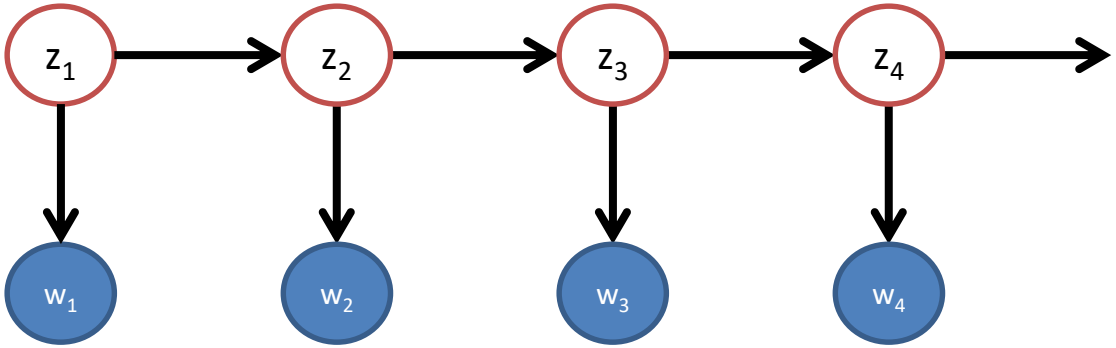
Directed (e.g., hidden Markov model [HMM]; generative)



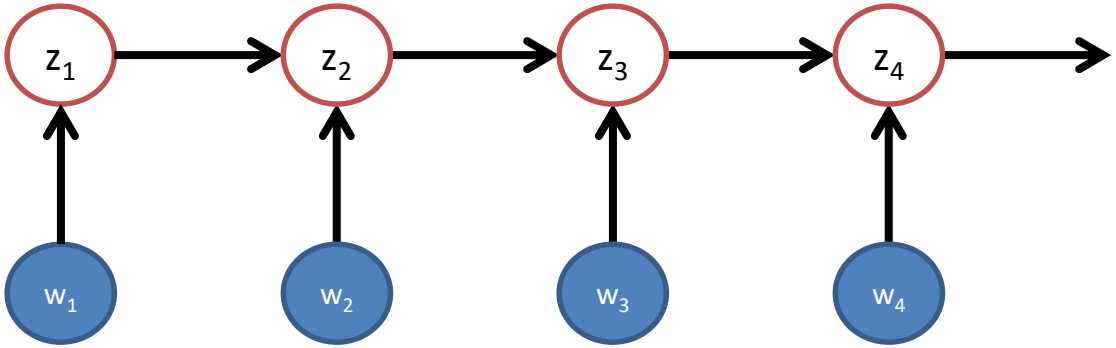
- Generate each tag, and generate each word from the tag
- Locally normalized

Example: Linear Chain

Directed (e.g., hidden Markov model [HMM]; generative)



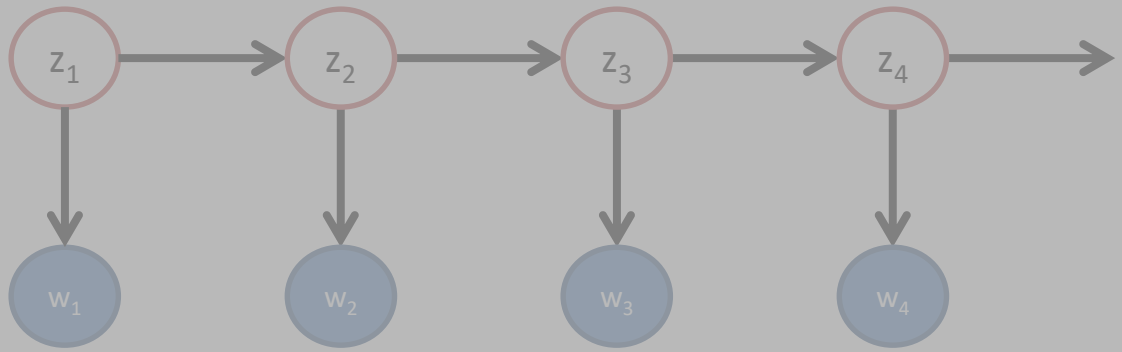
Directed (e.g.., maximum entropy Markov model [MEMM]; conditional)



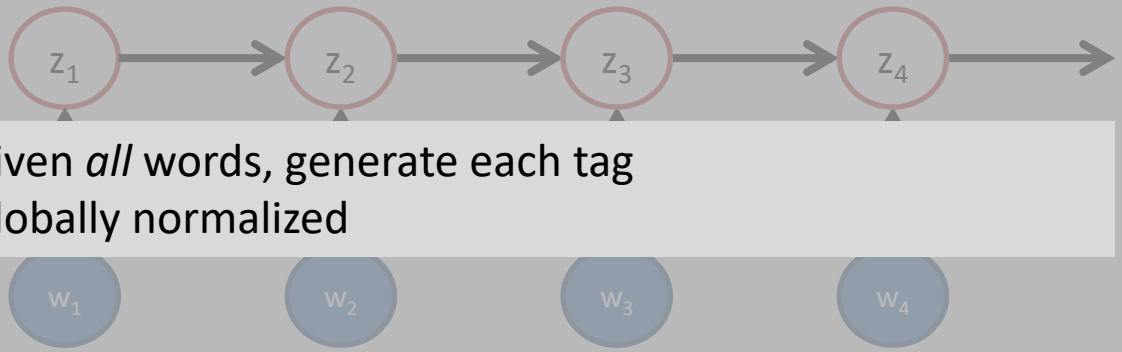
- Given each word, generate (predict) each tag
- Locally normalized

Example: Linear Chain

Directed (e.g., hidden Markov model [HMM]; generative)

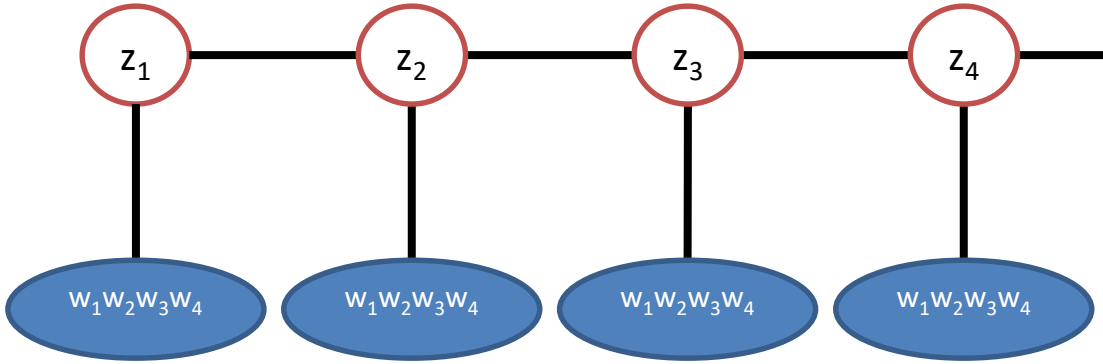


Directed (e.g.., maximum entropy Markov model [MEMM]; conditional)



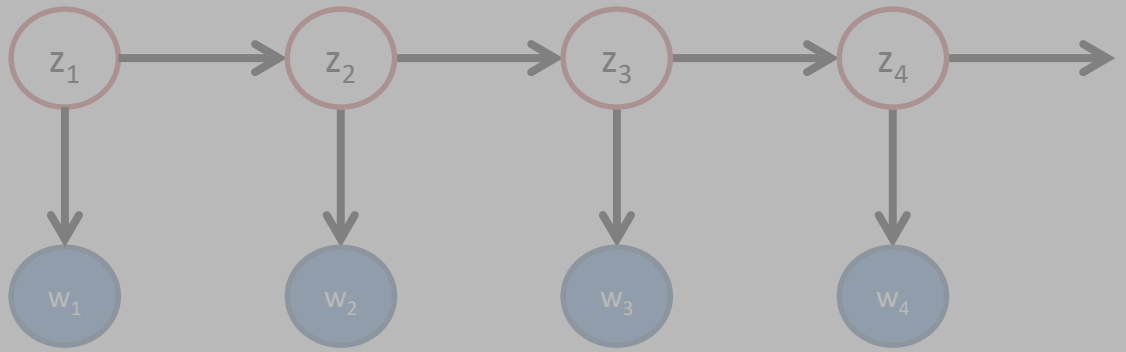
- Given *all* words, generate each tag
- Globally normalized

Undirected (e.g., conditional random field [CRF])

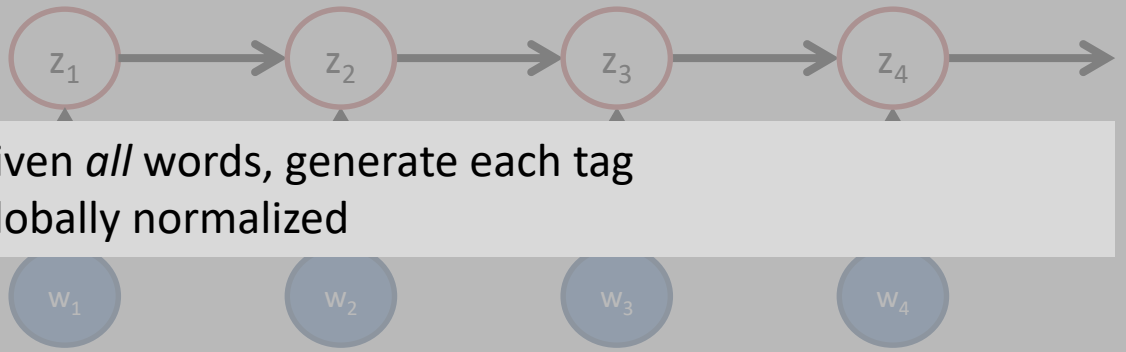


Example: Linear Chain

Directed (e.g., hidden Markov model [HMM]; generative)

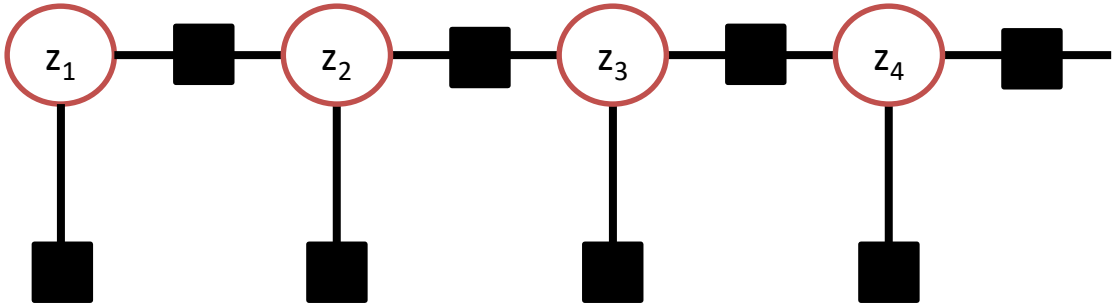


Directed (e.g.., maximum entropy Markov model [MEMM]; conditional)



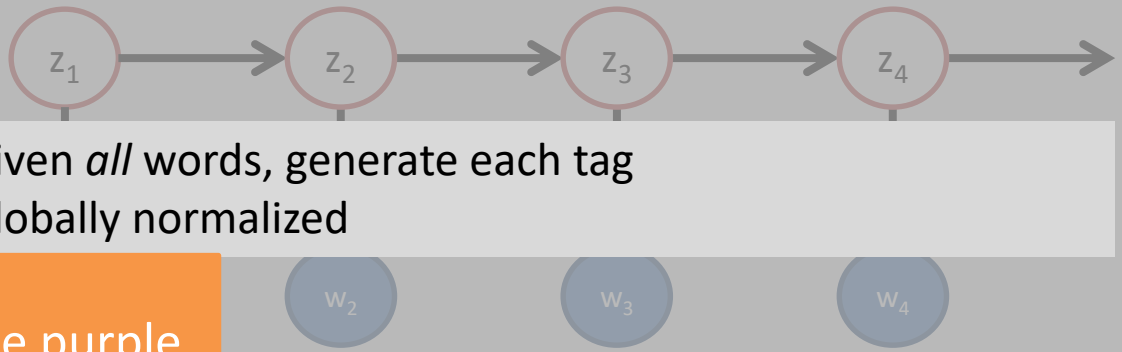
- Given *all* words, generate each tag
- Globally normalized

Undirected as factor graph
(e.g., conditional random field [CRF])



Example: Linear Chain

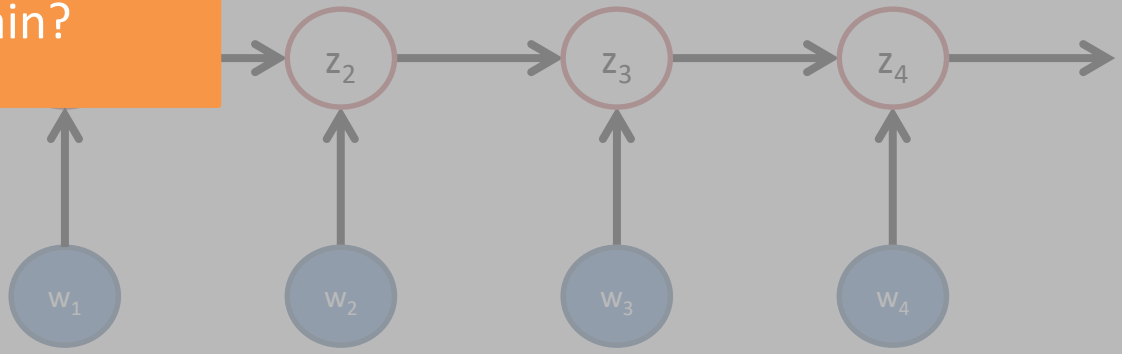
Directed (e.g., hidden Markov model [HMM]; generative)



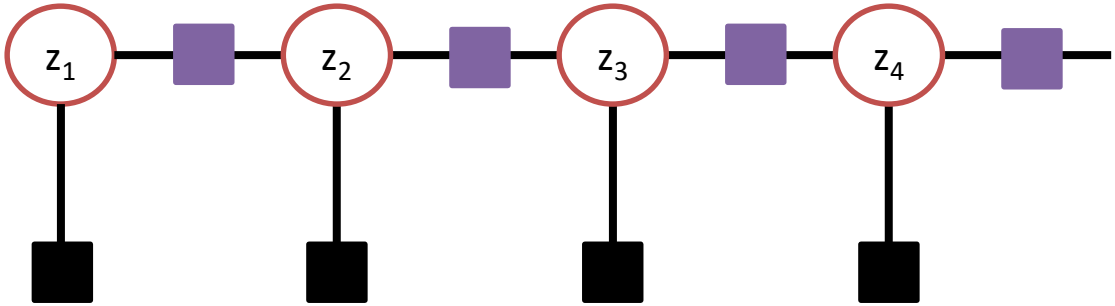
- Given *all* words, generate each tag
- Globally normalized

Q: What would the purple factors contain?

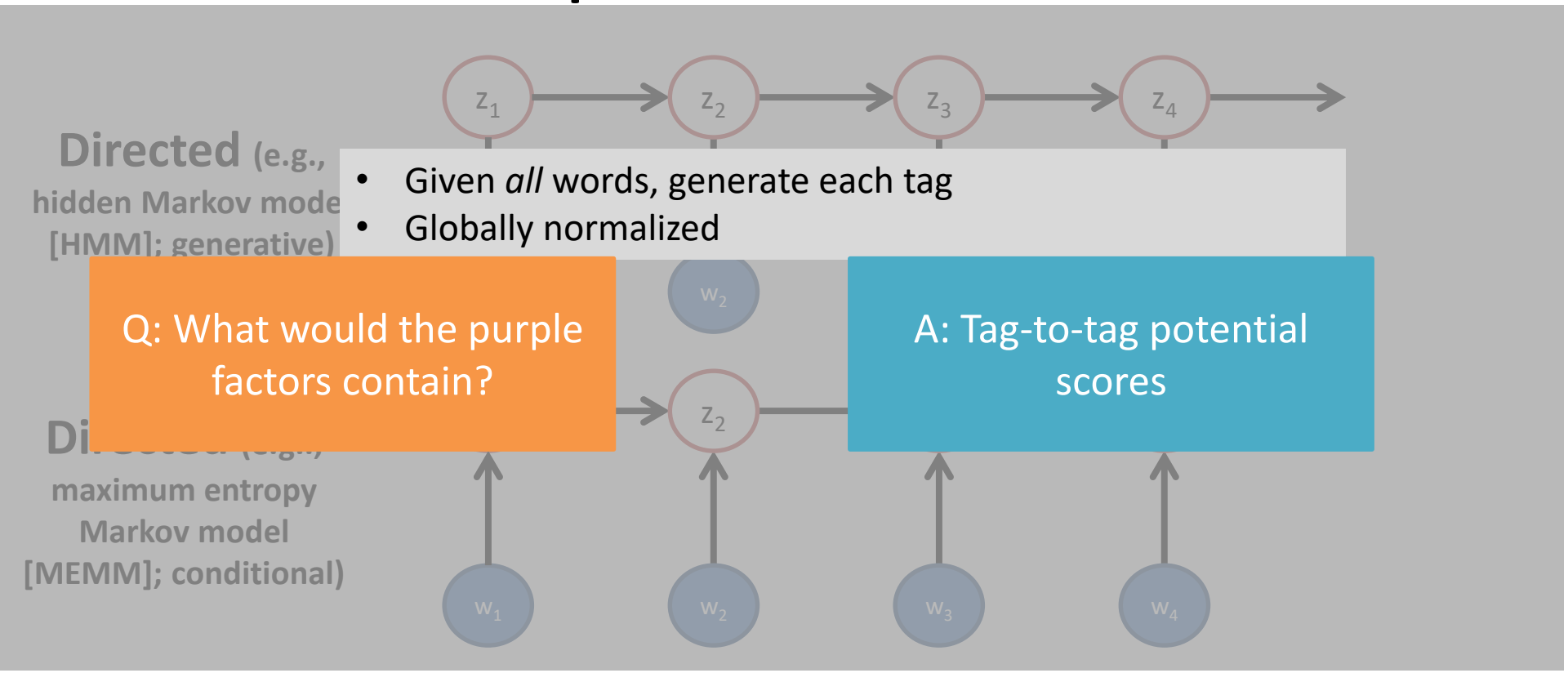
Di (e.g., maximum entropy Markov model [MEMM]; conditional)



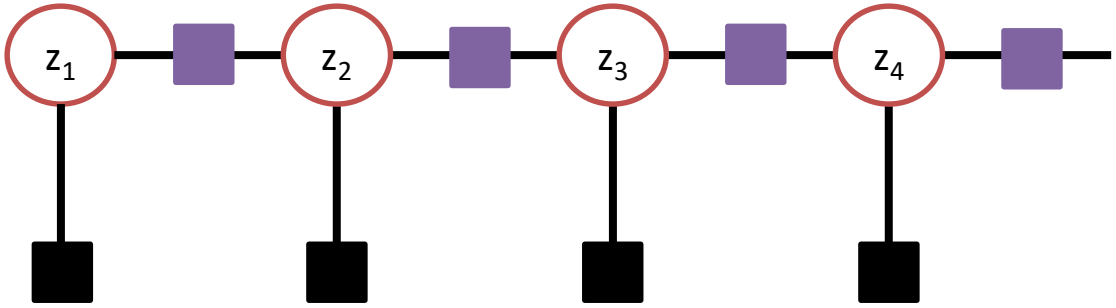
Undirected as factor graph
(e.g., conditional random field [CRF])



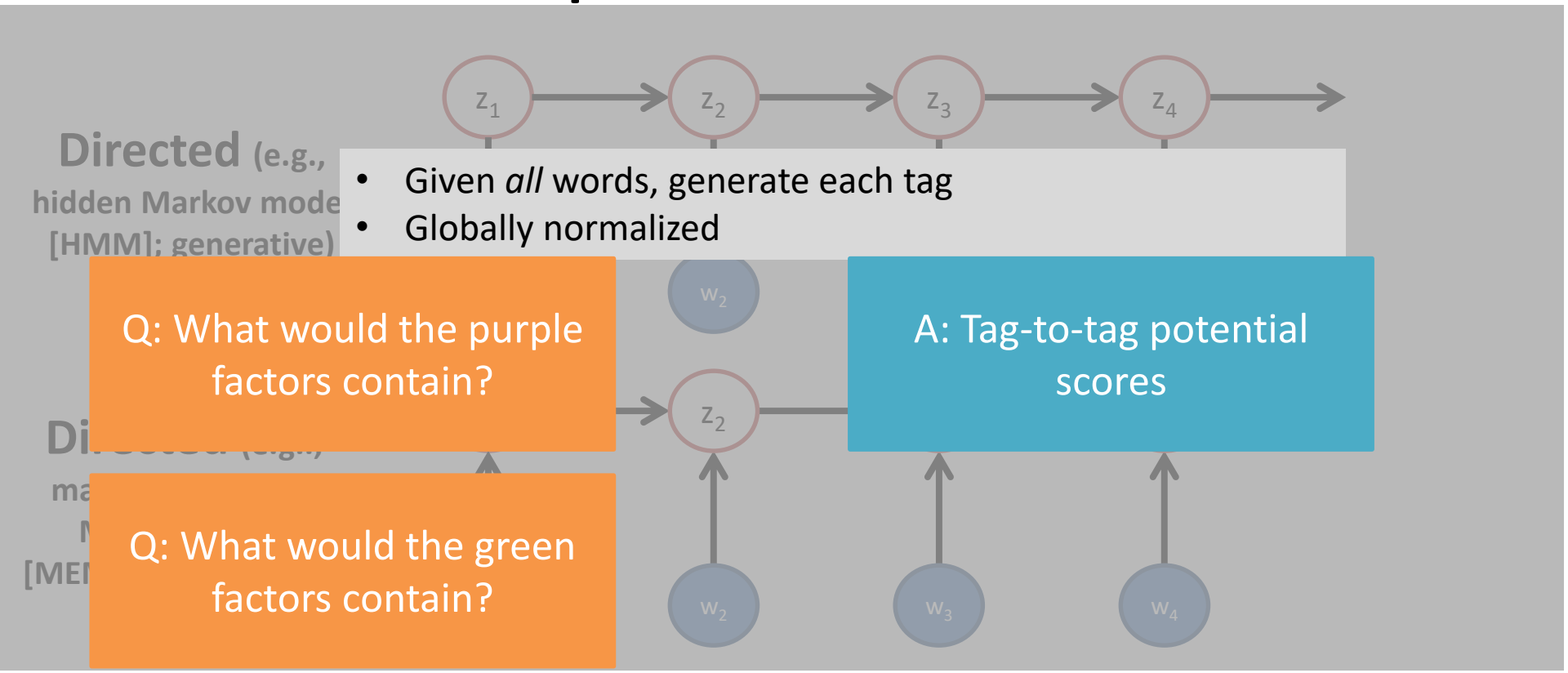
Example: Linear Chain



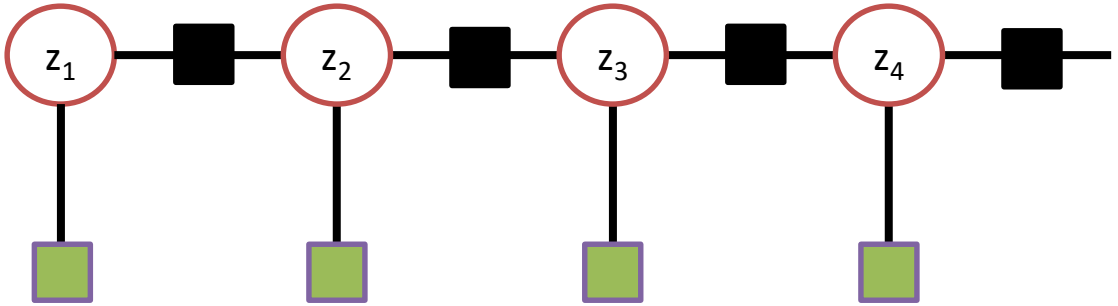
Undirected as factor graph
(e.g., conditional random field [CRF])



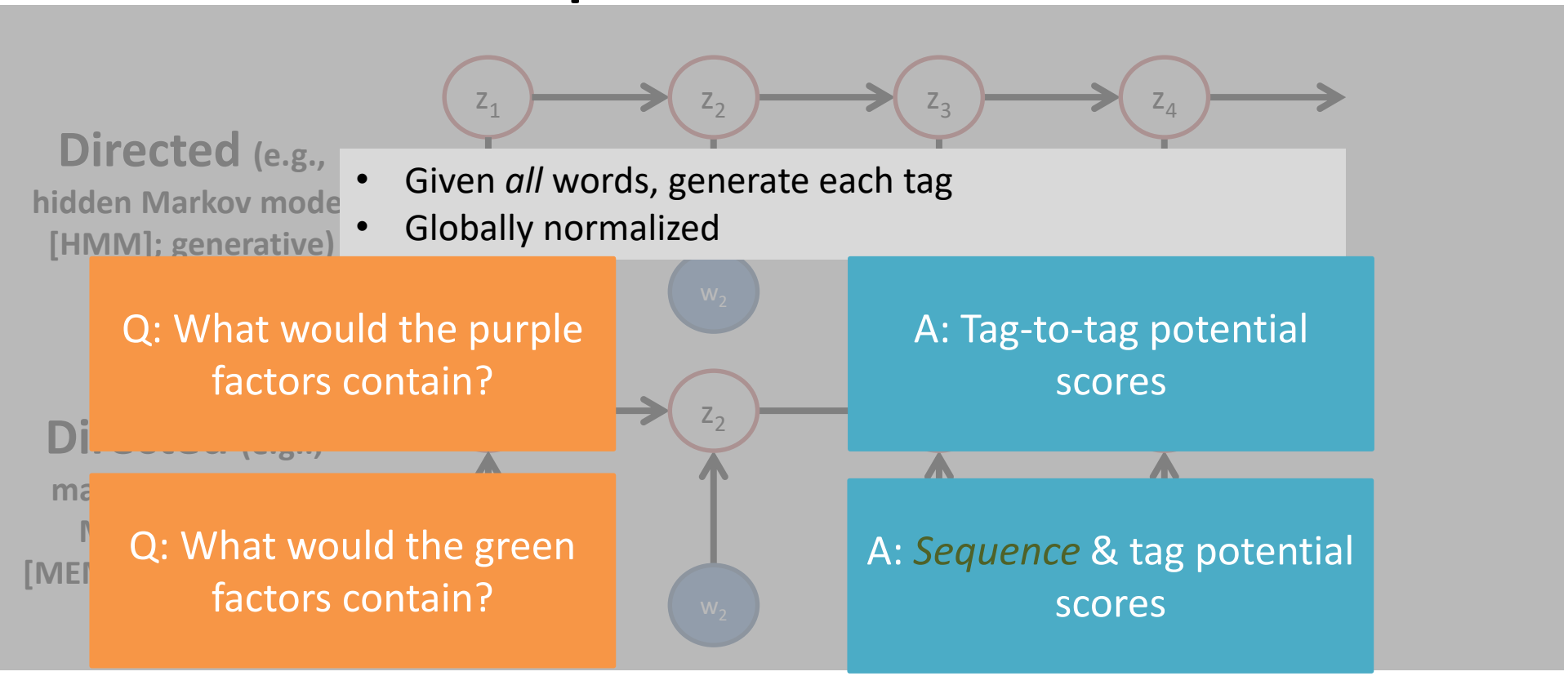
Example: Linear Chain



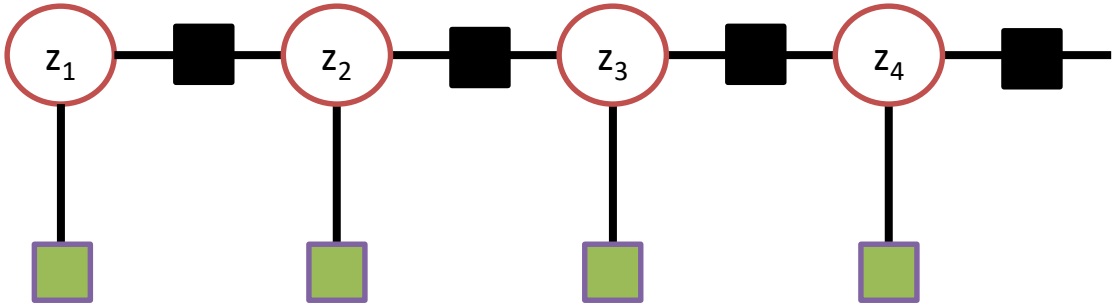
Undirected as factor graph
(e.g., conditional random field [CRF])



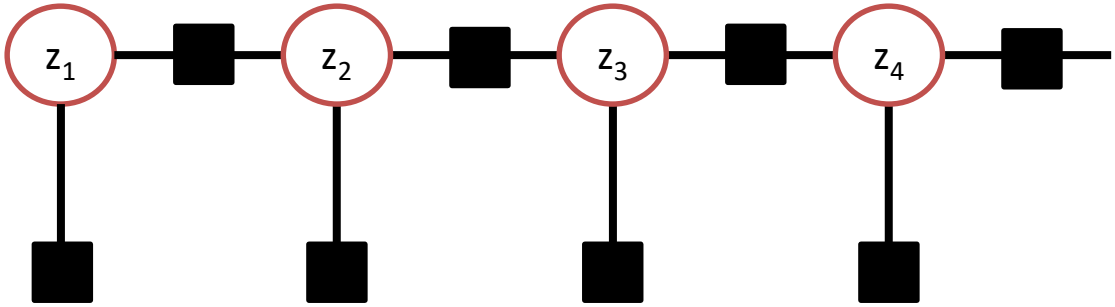
Example: Linear Chain



Undirected as factor graph
(e.g., conditional random field [CRF])



Example: Linear Chain Conditional Random Field

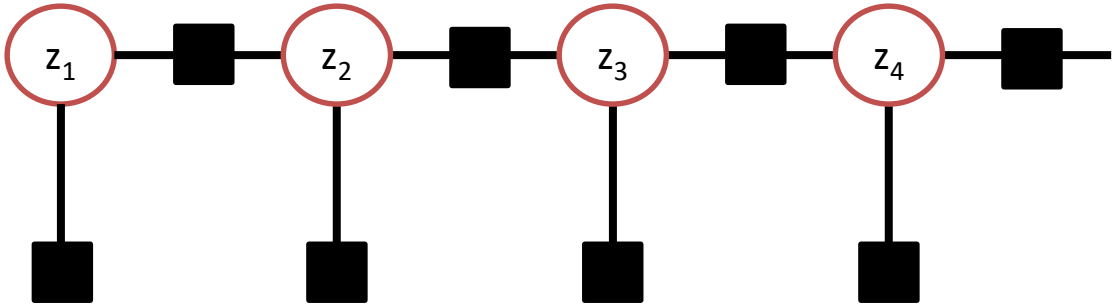


Widely used in applications like part-of-speech tagging

Noun-Mod *Noun* *Verb* *Noun*

President Obama told Congress …

Example: Linear Chain Conditional Random Field



Widely used in applications like

part-of-speech tagging

Noun-Mod *Noun* *Verb* *Noun*

President Obama told Congress …

and named entity recognition

Person *Person* *Other* *Org.*

President Obama told Congress …

Linear Chain CRFs for Part of Speech Tagging

A linear chain CRF is a conditional probabilistic model of the sequence of tags $z_1, z_2, \ldots, z_N$ conditioned on the *entire* input sequence $x_{1:N}$

Linear Chain CRFs for Part of Speech Tagging

$$p(\clubsuit|\diamondsuit)$$

A linear chain CRF is a conditional probabilistic model of the sequence of tags $z_1, z_2, \ldots, z_N$ conditioned on the *entire* input sequence $x_{1:N}$

Linear Chain CRFs for Part of Speech Tagging

$$p(z_1, z_2, \ldots, z_N | \Diamond)$$

A linear chain CRF is a conditional probabilistic model of the sequence of tags $z_1, z_2, \ldots, z_N$ conditioned on the *entire* input sequence $x_{1:N}$

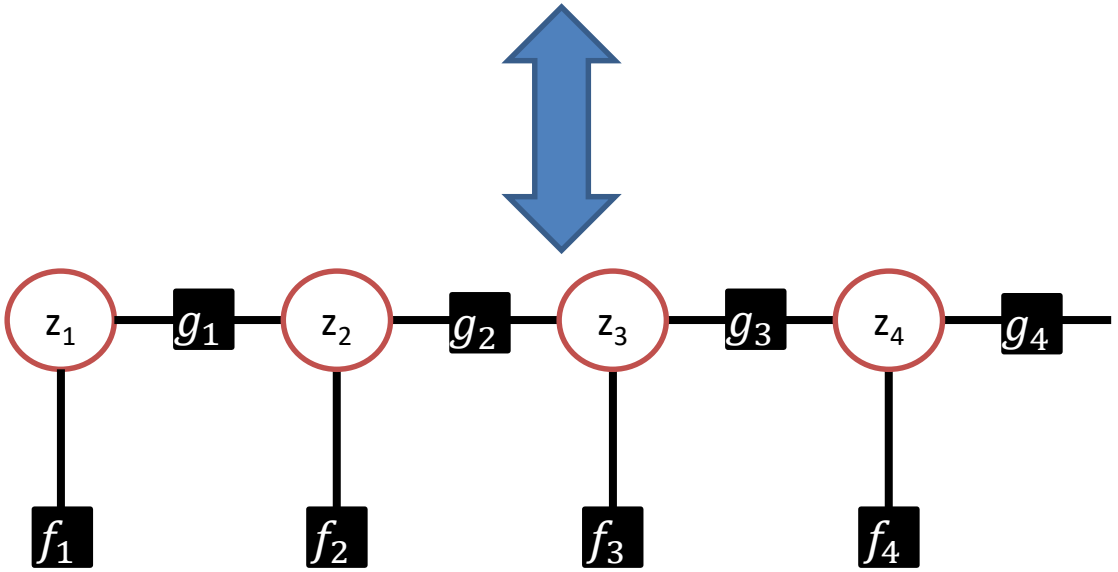
Linear Chain CRFs for Part of Speech Tagging

$$p(z_1, z_2, \ldots, z_N | x_{1:N})$$

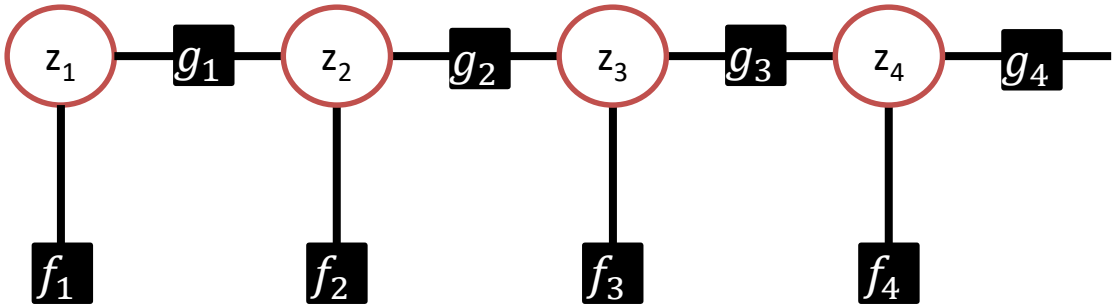
A linear chain CRF is a conditional probabilistic model of the sequence of tags $z_1, z_2, \ldots, z_N$ conditioned on the *entire* input sequence $x_{1:N}$

Linear Chain CRFs for Part of Speech Tagging

$$p(z_1, z_2, \ldots, z_N | x_{1:N})$$

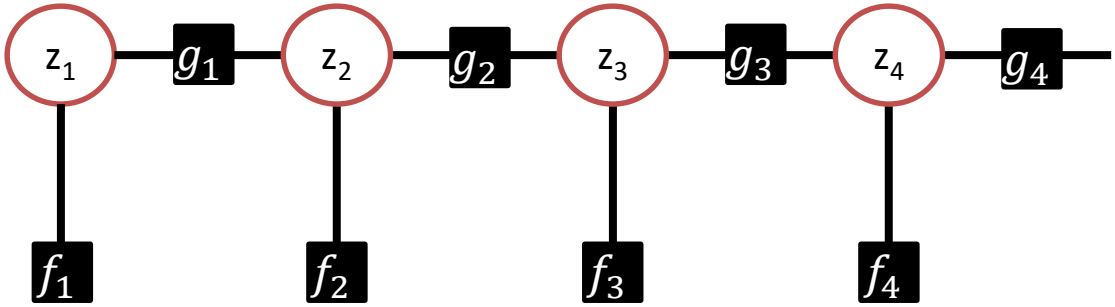


Linear Chain CRFs for Part of Speech Tagging



Q: What's the general formula for a factor graph/undirected PGM distribution?

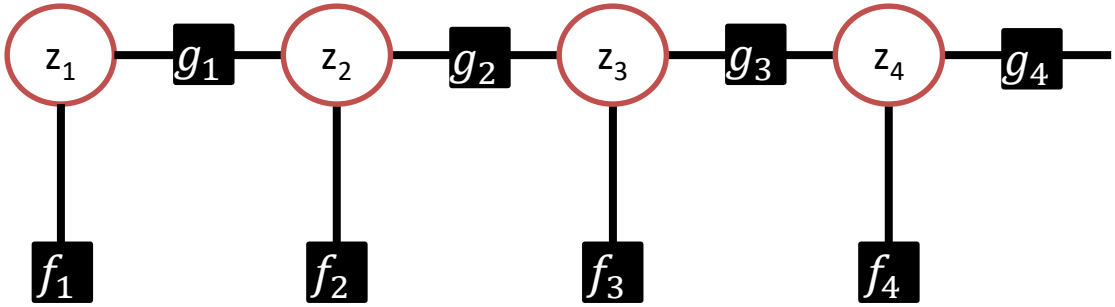
Linear Chain CRFs for Part of Speech Tagging



Q: What's the general formula for a factor graph/undirected PGM distribution?

A: $p(z_1, z_2, \ldots, z_N) = \frac{1}{Z} \prod_C \psi_C(z_c)$

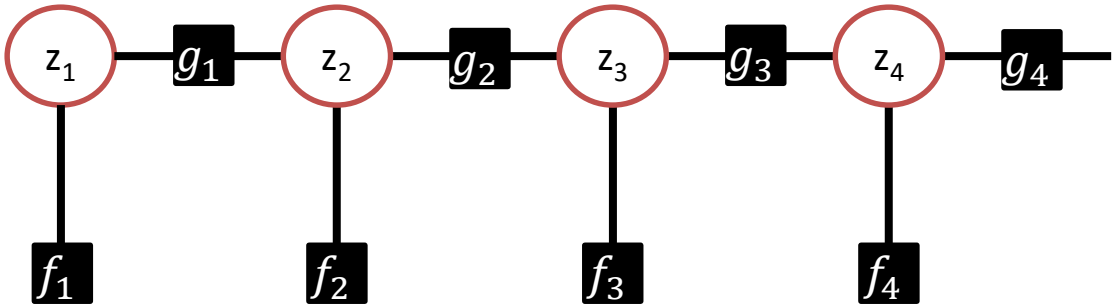
Linear Chain CRFs for Part of Speech Tagging



$$p(z_1, z_2, \ldots, z_N) = \frac{1}{Z} \prod_C \psi_C(z_c)$$

$p(z_1, z_2, \ldots, z_N | x_{1:N}) \propto$
product of exponentiated potential scores

Linear Chain CRFs for Part of Speech Tagging

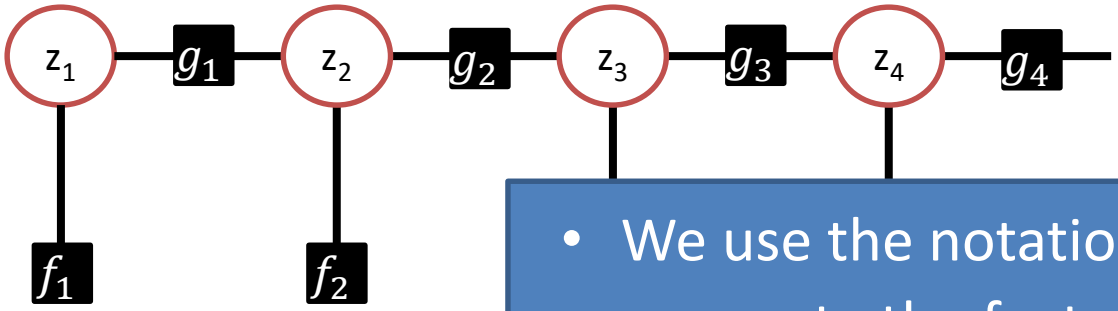


$$p(z_1, z_2, \ldots, z_N | x_{1:N}) \propto$$

$$\exp\left(-E_{g_1}(g_1)\right) \ldots \exp\left(-E_{g_N}(g_N)\right) *$$

$$\exp\left(-E_{f_1}(f_1)\right) \ldots \exp\left(-E_{f_N}(f_N)\right)$$

Linear Chain CRFs for Part of Speech Tagging



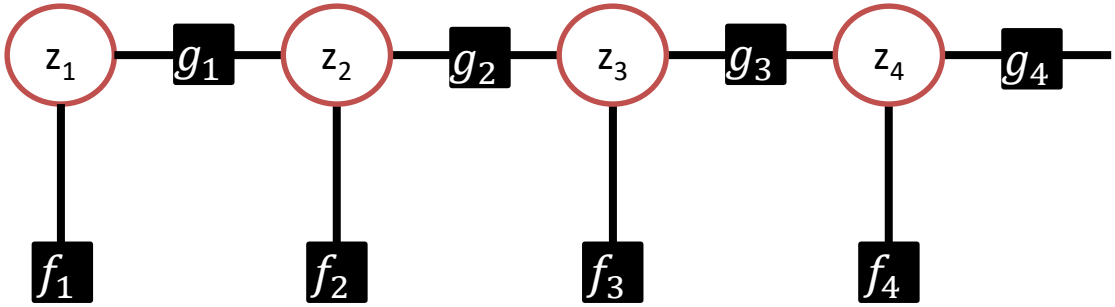
- We use the notation $E_{g_i}(g_i)$ to separate the features g_i from how we reweight them
- We use $-E_{g_i}$ to represent these as Boltzmann distributions

$$p(z_1, z_2, \ldots, z_N | x_{1:N}) \propto$$

$$\exp\left(-E_{g_1}(g_1)\right) \ldots \exp\left(-E_{g_N}(g_N)\right) *$$

$$\exp\left(-E_{f_1}(f_1)\right) \ldots \exp\left(-E_{f_N}(f_N)\right)$$

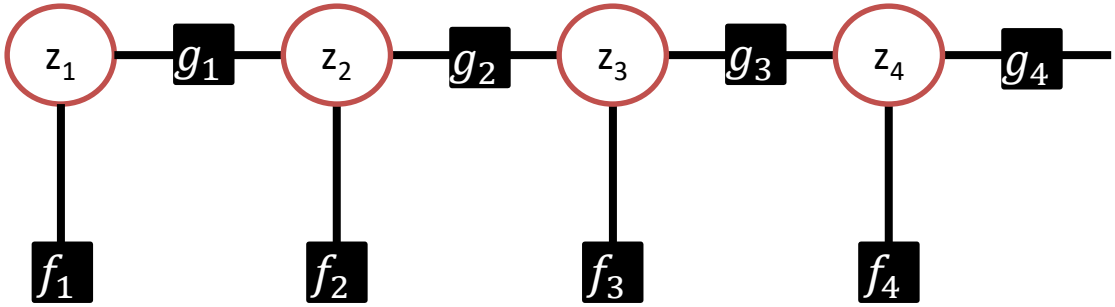
Linear Chain CRFs for Part of Speech Tagging



$$p(z_1, z_2, \ldots, z_N | x_{1:N}) \propto$$

$$\prod_{i=1}^{N} \exp\left(-E_{g_i}(g_i)\right) \exp\left(-E_{f_i}(f_i)\right)$$

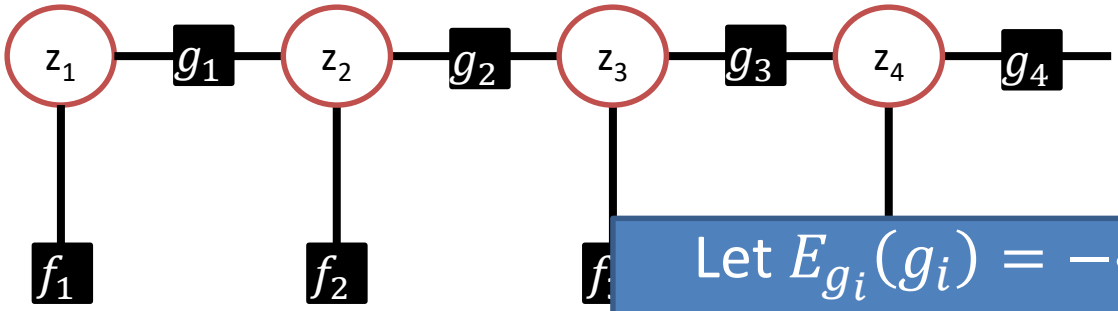
Linear Chain CRFs for Part of Speech Tagging



$$p(z_1, z_2, \ldots, z_N | x_{1:N}) \propto$$

$$\prod_{i=1}^{N} \exp\left(-\left(E_{g_i}(g_i) + E_{f_i}(f_i)\right)\right)$$

Linear Chain CRFs for Part of Speech Tagging



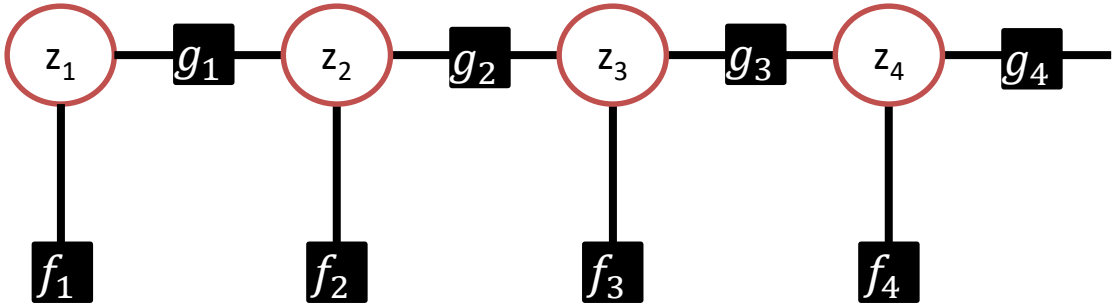
Let $E_{g_i}(g_i) = -\langle \theta^{(g)}, g_i \rangle$

Let $E_{f_i}(f_i) = -\langle \theta^{(f)}, f_i \rangle$

where $\theta^{(f)}, \theta^{(g)}$ are learnable parameters

$$p(z_1, z_2, \ldots, z_N | x_{1:N}) \propto \prod_{i=1}^{N} \exp\left(-\left(E_{g_i}(g_i) + E_{f_i}(f_i)\right)\right)$$

Linear Chain CRFs for Part of Speech Tagging

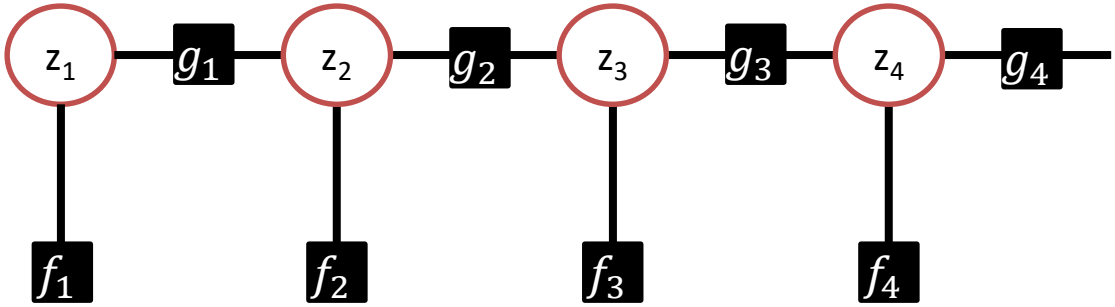


$$p(z_1, z_2, \ldots, z_N | x_{1:N}) \propto$$

$$\prod_{i=1}^{N} \exp(\langle \theta^{(f)}, f_i(z_i) \rangle + \langle \theta^{(g)}, g_i(z_i, z_{i+1}) \rangle)$$

Linear Chain CRFs for Part of Speech Tagging

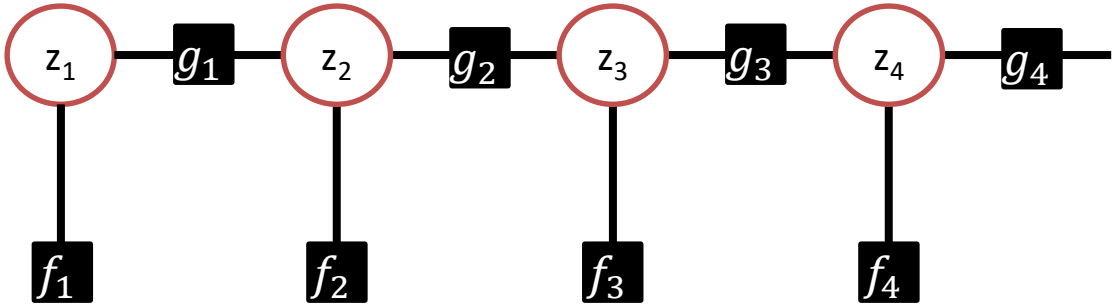
g_j: inter-tag features (can depend on any/all input words $x_{1:N}$)



Linear Chain CRFs for Part of Speech Tagging

g_j: inter-tag features (can depend on any/all input words $x_{1:N}$)

f_i: solo tag features (can depend on any/all input words $x_{1:N}$)



Linear Chain CRFs for Part of Speech Tagging

g_j: inter-tag features (can depend on any/all input words $x_{1:N}$)

f_i: solo tag features (can depend on any/all input words $x_{1:N}$)

Feature design, just like in maxent models!

Linear Chain CRFs for Part of Speech Tagging

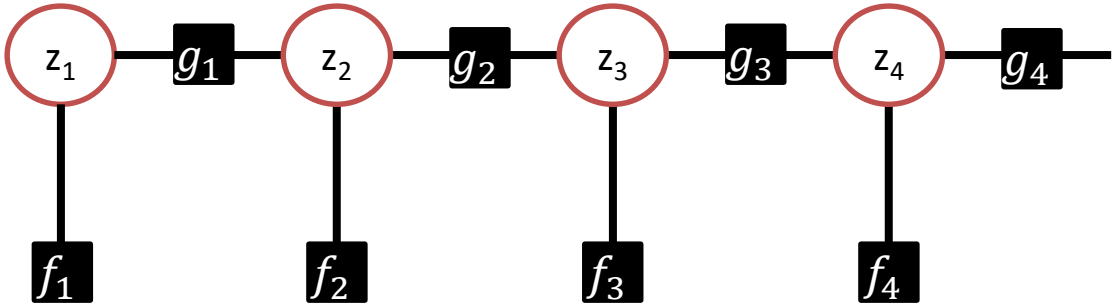
g_j: inter-tag features (can depend on any/all input words $x_{1:N}$)

f_i: solo tag features (can depend on any/all input words $x_{1:N}$)

Example:

$g_{j,N\to V}(\mathrm{z_j}, \mathrm{z_{j+1}}) = 1$ (if $\mathrm{z_j} == \mathrm{N}$ & $\mathrm{z_{j+1}} == \mathrm{V}$) else 0

$g_{j,\mathrm{told},N\to V}(\mathrm{z_j}, \mathrm{z_{j+1}}) = 1$ (if $\mathrm{z_j} == \mathrm{N}$ & $\mathrm{z_{j+1}} == \mathrm{V}$ & $\mathrm{x_j} == \mathrm{told}$) else 0



(For discussion/whiteboard)

- How would we learn a CRF?
- What objective would we optimize?
- How would we use BP?

Key Insights (1)

- Minimize (structured) cross-entropy loss ↔ (structured) maximum likelihood

- Gradient has very familiar form of

  "observed feature counts – expected feature counts"

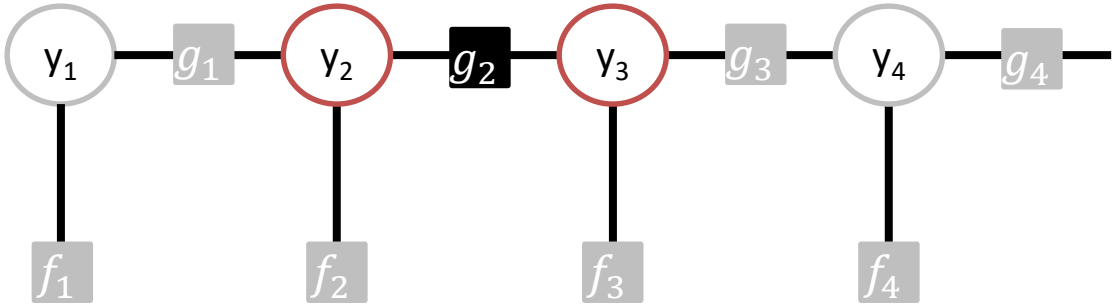
Key Insights (2)

- Rely on adjacency connections/independence assumptions to compute

$$\mathbb{E}_{y'}\left[\sum_i h_i(y')\right] = \sum_i \sum_{y_{i-1}, y_i} p(y_{i-1}, y_i | x_{1:N}) h_i(y_{i-1}, y_i)$$

Key Insights (3)

$$p(x_{\{m\}} = \boldsymbol{v}) \propto m(x_{\{m\}} = \boldsymbol{v}) \prod_{x_{n_i} \in N(m)} q_{n_i \to m}(x_{n_i} = v_i)$$



- Run BP to compute beliefs (unnormalized, joint marginals)

$$p(y_{i-1}, y_i | x_{1:N}) \propto$$
$$g_{i-1}(y_{i-1}, y_i) *$$
$$q_{y_{i-1} \to g_{i-1}}(y_{i-1}) *$$
$$q_{y_i \to g_{i-1}}(y_i)$$

Outline

Message Passing: Graphical Model Inference

Example: Linear Chain CRF