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

\[ p(x_1, x_2, x_3, \ldots, x_N) = \frac{1}{Z} \prod_{C} \psi_C(x_C) \]

**Q**: What restrictions should we place on the potentials \( \psi_C \)?

**A**: \( \psi_C \geq 0 \) (or \( \psi_C > 0 \))
Terminology: Potential Functions

\[ p(x_1, x_2, x_3, ..., x_N) = \frac{1}{Z} \prod_C \psi_C(x_C) \]

energy function (for clique C)

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

\[ \psi_C(x_C) = \exp -E(x_C) \]

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 \( p(\mathbf{X}_1, ..., \mathbf{X}_N) \)

Factor graph of \( p \): Bipartite graph of evidence nodes \( \mathbf{X} \), factor nodes \( \mathbf{F} \), and edges \( T \)

**Evidence nodes \( \mathbf{X} \)** are the random variables

**Factor nodes \( \mathbf{F} \)** 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

\[ Z_n(v) = \sum_{x:x_n=v} \prod_C \psi_C(x_c) \]

Example: 3 variables, fix the 2\textsuperscript{nd} 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

Finding the normalizer

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

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

Q: Why are these difficult?

A: Many different combinations

Computing the marginals

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

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

\[
p(x_i = v) = \sum_{x : x_i = v} p(x_1, x_2, \ldots, x_i, \ldots, x_N)
\]
The factor nodes can act as filters.
Sum-Product

**definition of marginal**

\[ p(x_i = v) = \sum_{x : x_i = v} p(x_1, x_2, ..., x_i, ..., x_N) \]

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

\[ r_{m \rightarrow n} \] is a **message** from factor node m to evidence node n
Sum-Product

\[
p(x_i = v) = \frac{\prod_f r_{f \rightarrow x_i}(x_i = v)}{\sum_w \prod_f r_{f \rightarrow x_i}(x_i = w)} \propto \prod_f r_{f \rightarrow x_i}(x_i)
\]

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

\( r_{m \rightarrow n} \) is a **message** from factor node \( m \) to evidence node \( n \)
IF FACTORS CAN SEND MESSAGES TO NODES

CAN NODES SEND MESSAGES TO FACTORS?
**Sum-Product**

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

$q_{n \rightarrow m_1}$

$r_{m_1 \rightarrow n}$

$q_{n \rightarrow m_2}$

$r_{m_2 \rightarrow n}$

$q_{n \rightarrow m_3}$

$r_{m_3 \rightarrow n}$

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

From **variables** to **factors**

\[ q_{n \rightarrow m}(x_n) = n \text{ aggregates information from the rest of its graph via its neighbors} \]
From variables to factors

\[ q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n) \]

set of factors in which variable \( n \) participates

default value of 1 if empty product
Sum-Product

**From variables to factors**

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

- Set of factors in which variable $n$ participates.

**From factors to variables**

$$r_{m \rightarrow n}(x_n) = m \text{ aggregates information from the rest of its graph via its neighbors}$$

- Default value of 1 if empty product.
Sum-Product

From variables to factors

\[ q_{n \rightarrow m}(x_n) = \prod_{m' \in \mathcal{M}(n) \setminus m} r_{m' \rightarrow n}(x_n) \]

set of factors in which variable \( n \) participates

From factors to variables

\[ r_{m \rightarrow n}(x_n) = m \text{ aggregates information from the rest of its graph via its neighbors} \]

But these neighbors are R.V.s that take on different values
**Sum-Product**

**From variables to factors**

\[ q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n) \]

*set of factors in which variable \( n \) participates*

**From factors to variables**

\[ r_{m \rightarrow n}(x_n) = \]

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

*default value of 1 if empty product*
Sum-Product

From variables to factors

\[ q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n) \]

set of factors in which variable \( n \) participates

From factors to variables

\[ r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'}) \]

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

\[ q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n) \]

set of factors in which variable n participates

From factors to variables

\[ r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'}) \]

sum over configuration of variables for the \( m^{th} \) factor, with variable \( n \) fixed

set of variables that the \( m^{th} \) factor depends on

default value of 1 if empty product
Meaning of the Computed Values

From variables to factors

\[ q_{n\rightarrow m}(x_n) = \prod_{m' \in M(n) \backslash m} r_{m' \rightarrow n}(x_n) \]

From factors to variables

\[ r_{m \rightarrow n}(x_n) = \sum_{w_m \backslash n} f_m(w_m) \prod_{n' \in N(m) \backslash n} q_{n' \rightarrow m}(x_{n'}) \]

\( x_n \) telling factor \( m \) the “goodness” for the rest of the graph if \( x_n \) has a particular value.

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

$r_{m_1 \rightarrow n}(x_n)$ tells $x_n$ the “goodness” from $m_1$’s perspective if $x_n$ has a particular value.

$r_{m_2 \rightarrow n}(x_n)$ tells $x_n$ the “goodness” from $m_2$’s perspective if $x_n$ has a particular value.
From Messages to Variable \textit{Beliefs}

\( r_{m_1 \rightarrow n}(x_n) \) tells \( x_n \) the “goodness” from \( m_1 \)’s perspective if \( x_n \) has a particular value.

\( r_{m_2 \rightarrow n}(x_n) \) 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*

\[ r_{m_1 \rightarrow n}(x_n) \text{ tells } x_n \text{ the “goodness” from } m_1 \text{’s perspective if } x_n \text{ has a particular value} \]

\[ r_{m_2 \rightarrow n}(x_n) \text{ tells } x_n \text{ the “goodness” from } m_2 \text{’s perspective if } x_n \text{ has a particular value} \]

Together, they describe the cover the entire graph!

\[ p(x_n = v) \propto r_{m_1 \rightarrow n}(x_n = v) \, r_{m_2 \rightarrow n}(x_n = v) \]
From Messages to **Variable Beliefs**: General Formula

\[ p(x_n = v) \propto \prod_{m \in N(x_n)} r_{m \rightarrow n}(x_n = v) \]

- \( r_{m_1 \rightarrow n}(x_n) \) tells \( x_n \) the “goodness” from \( m_1 \)’s perspective if \( x_n \) has a particular value.
- \( r_{m_2 \rightarrow n}(x_n) \) tells \( x_n \) the “goodness” from \( m_2 \)’s perspective if \( x_n \) has a particular value.
From Messages to **Factor Beliefs**: General Formula

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

$$p(x_{\{m\}} = \nu) \propto m(x_{\{m\}} = \nu) \prod_{x_{n_i} \in N(m)} q_{n_i \rightarrow m}(x_{n_i} = \nu_i)$$
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 $p(z_1, ..., z_N|w_{1:N})$
  • 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?
Q: What are the variables?
A: $x_1, x_2, x_3, x_4$

Q: What are the factors?
A: $f_a, f_b, f_c$
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?
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

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

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$
$$r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$
Example

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

\[ q_{x_1 \rightarrow f_a}(x_1) = 1 \]
\[ q_{x_4 \rightarrow f_c}(x_4) = 1 \]
\[ r_{f_a \rightarrow x_2}(x_2) = ??? \]

\[ q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n) \quad r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'}) \]
Example

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

\[ q_{x_1 \rightarrow f_a}(x_1) = 1 \]
\[ q_{x_4 \rightarrow f_c}(x_4) = 1 \]

\[ r_{f_a \rightarrow x_2}(x_2) = \sum_{k} f_a(x_1 = k, x_2) \]
\[ r_{f_c \rightarrow x_2}(x_2) = \sum_{k} f_a(x_2, x_4 = k) \]

\[ q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n) \]
\[ r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'}) \]
Example

1. Select the root, or pick one if a tree \((x_3)\)

1. Send messages from leaves to root

\[ q_{x_1 \rightarrow f_a}(x_1) = 1 \]
\[ q_{x_4 \rightarrow f_c}(x_4) = 1 \]
\[ r_{f_a \rightarrow x_2}(x_2) = \sum_k f_a(x_1 = k, x_2) \]
\[ r_{f_c \rightarrow x_2}(x_2) = \sum_k f_a(x_2, x_4 = k) \]
\[ q_{x_2 \rightarrow f_b}(x_2) = ?? \]

\[
q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n) \quad r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})
\]
Example

1. Select the root, or pick one if a tree \( x_3 \)

1. Send messages from leaves to root

\[
q_{x_1 \rightarrow f_a}(x_1) = 1 \\
q_{x_4 \rightarrow f_c}(x_4) = 1 \\
\]

\[
rf_{a \rightarrow x_2}(x_2) = \sum_{k} f_a(x_1 = k, x_2) \\
rf_{c \rightarrow x_2}(x_2) = \sum_{k} f_a(x_2, x_4 = k) \\
q_{x_2 \rightarrow f_b}(x_2) = rf_{a \rightarrow x_2}(x_2)rf_{c \rightarrow x_2}(x_2) \\
\]

\[
qn_{\rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n) \\
r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'}) \\
\]
Example

1. Select the root, or pick one if a tree ($x_3$)

1. Send messages from leaves to root

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

- $r_{f_a \rightarrow x_2}(x_2) = \sum_k f_a(x_1 = k, x_2)$
- $r_{f_c \rightarrow x_2}(x_2) = \sum_k f_a(x_2, x_4 = k)$
- $q_{x_2 \rightarrow f_b}(x_2) = r_{f_a \rightarrow x_2}(x_2) r_{f_c \rightarrow x_2}(x_2)$
- $r_{f_b \rightarrow x_3}(x_3) = \sum_k f_b(x_2 = k, x_3)$

$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n) \\
r_{m \rightarrow n}(x_n) = \sum f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$
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

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

$$q_{n \rightarrow m} (x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n} (x_n) \quad r_{m \rightarrow n} (x_n) = \sum_{w_m \setminus n} f_m (w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m} (x_{n'})$$
Example

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

\[
q_{x_3 \rightarrow f_b}(x_3) = 1
\]

\[
r_{f_b \rightarrow x_2}(x_2) = \sum_k f_b(x_2, x_3 = k)
\]

\[
q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)
\]

\[
r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})
\]
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

\[
q_{x_3 \rightarrow f_b}(x_3) = 1 \\
rf_{f_b \rightarrow x_2}(x_2) = \sum_{k} f_b(x_2, x_3 = k) \\
q_{x_2 \rightarrow f_a}(x_2) = ???
\]

\[
q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n) \\
r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})
\]
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

\[
q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n) \\
r_{m \to n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})
\]
Example

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

\[
q_{x_3 \rightarrow f_b}(x_3) = 1
\]
\[
r_{f_b \rightarrow x_2}(x_2) = \sum_{k} f_b(x_2, x_3 = k)
\]
\[
q_{x_2 \rightarrow f_a}(x_2) = r_{f_b \rightarrow x_2}(x_2)\ r_{f_c \rightarrow x_2}(x_2)
\]

Q: Where did we compute this?
A: In step 1 (leaves \(\rightarrow\) root)

\[
q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)
\]
\[
r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})
\]
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

$q_{x_3 \rightarrow f_b}(x_3) = 1$
$r_{f_b \rightarrow x_2}(x_2) = \sum_{k} f_b(x_2, x_3 = k)$
$q_{x_2 \rightarrow f_a}(x_2) = r_{f_b \rightarrow x_2}(x_2) r_{f_c \rightarrow x_2}(x_2)$
$q_{x_2 \rightarrow f_c}(x_2) = r_{f_a \rightarrow x_2}(x_2) r_{f_b \rightarrow x_2}(x_2)$

$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$
$r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$
Example

1. Select the root, or pick one if a tree \(x_3\)
2. Send messages from leaves to root
   \[q_{x_3 \rightarrow f_b}(x_3) = 1\]
3. Send messages from root to leaves
   \[r_{f_b \rightarrow x_2}(x_2) = \sum_k f_b(x_2, x_3 = k)\]
4. Compute messages for other nodes
   \[q_{x_2 \rightarrow f_a}(x_2) = r_{f_b \rightarrow x_2}(x_2) r_{f_c \rightarrow x_2}(x_2)\]
   \[q_{x_2 \rightarrow f_c}(x_2) = r_{f_a \rightarrow x_2}(x_2) r_{f_b \rightarrow x_2}(x_2)\]
   \[r_{f_c \rightarrow x_4}(x_4) = \sum_k f_c(x_2 = k, x_4)\]

\[q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)\]
\[r_{m \rightarrow n}(x_n) = \sum f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})\]
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

\[
\begin{align*}
q_{x_3 \rightarrow f_b}(x_3) &= 1 \\
rf_{b \rightarrow x_2}(x_2) &= \sum_k f_b(x_2, x_3 = k) \\
q_{x_2 \rightarrow f_a}(x_2) &= rf_{b \rightarrow x_2}(x_2)rf_{c \rightarrow x_2}(x_2) \\
q_{x_2 \rightarrow f_c}(x_2) &= rf_{a \rightarrow x_2}(x_2)rf_{b \rightarrow x_2}(x_2) \\
r_{f_c \rightarrow x_4}(x_4) &= \sum_k f_c(x_2 = k, x_4) \\
r_{f_a \rightarrow x_1}(x_1) &= \sum_k f_a(x_1, x_2 = k)
\end{align*}
\]

\[
q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n) \\
r_{m \rightarrow n}(x_n) = \sum f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})
\]
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

\[
p(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)
\]

\[
q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)
\]

\[
r_{n \rightarrow m}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})
\]
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

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

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

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

$$r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$
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

\[
p(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)
\]
\[
p(x_1) = r_{f_a \to x_1}(x_1)
\]
\[
p(x_2) = r_{f_a \to x_2}(x_2) r_{f_b \to x_2}(x_2) r_{f_c \to x_2}(x_2)
\]

\[
q_{n \to m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \to n}(x_n)
\]
\[
r_{m \to n}(x_n) = \sum_{m' \in M(n) \setminus m} f_m(w_m) \prod_{m' \in M(n) \setminus n} q_{m' \to n}(x_{m'})
\]
Example

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

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

\[
p(x_1) = r_{f_a \to x_1}(x_1)
\]

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

\[
p(x_3) = r_{f_b \to x_3}(x_3)
\]

\[
p(x_4) = r_{f_c \to x_4}(x_4)
\]

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

\[
r_{m \to n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \to m}(x_{n'})
\]
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. 

\[
q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n) \quad r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})
\]
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.

\[
q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n) \\
q_{n' \rightarrow m}(x_{n'}) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})
\]
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]

\[
q_{n\rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n) \quad r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})
\]
Max-Product (Max-Sum)

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

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

\[ r_{m \rightarrow n}(x_n) = \max_{w_m \setminus n} f_m(w_m) \prod_{n' \in \mathcal{N}(m) \setminus n} q_{n' \rightarrow m}(x_{n'}) \]

(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 \textit{anyway} and hope for the best

Requires a \textit{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)

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

**Directed** (e.g., maximum entropy Markov model [MEMM]; conditional)
Example: Linear Chain

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

- Given all words, generate each tag
- Globally normalized

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

Undirected (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

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

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

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

Q: What would the purple factors contain?
Example: Linear Chain

**Directed** (e.g., hidden Markov model [HMM]; generative)
- Given *all* words, generate each tag
- Globally normalized

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

Q: What would the purple factors contain?

A: Tag-to-tag potential scores

**Directed 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?
A: Tag-to-tag potential scores

Q: What would the green factors contain?

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

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

Q: What would the purple factors contain?
A: Tag-to-tag potential scores

Q: What would the green factors contain?
A: Sequence & tag potential scores
Example: Linear Chain Conditional Random Field

Widely used in applications like part-of-speech tagging

President Obama told Congress ...
Example: Linear Chain Conditional Random Field

Widely used in applications like part-of-speech tagging

*President Obama told Congress ...*

and named entity recognition

*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 $x_{1:N}$.
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 $x_{1:N}$. 
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: \( p(z_1, z_2, ..., z_N) = \frac{1}{Z} \prod_C \psi_C(z_c) \)
Linear Chain CRFs for Part of Speech Tagging

\[ p(z_1, z_2, ..., z_N) = \frac{1}{Z} \prod_c \psi_c(z_c) \]

\[ p(z_1, z_2, ..., z_N \mid x_{1:N}) \propto \text{product of exponentiated potential scores} \]
Linear Chain CRFs for Part of Speech Tagging

\[ p(z_1, z_2, \ldots, z_N | x_{1:N}) \propto \exp \left( -E_{g_1}(g_1) \right) \ldots \exp \left( -E_{g_N}(g_N) \right) \times \exp \left( -E_{f_1}(f_1) \right) \ldots \exp \left( -E_{f_N}(f_N) \right) \]
Linear Chain CRFs for Part of Speech Tagging

\[ p(z_1, z_2, \ldots, z_N | x_{1:N}) \propto \exp \left( -E_{g_1}(g_1) \right) \cdots \exp \left( -E_{g_N}(g_N) \right) * \exp \left( -E_{f_1}(f_1) \right) \cdots \exp \left( -E_{f_N}(f_N) \right) \]

- 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.
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

\[ p(z_1, z_2, ..., z_N | x_{1:N}) \propto \prod_{i=1}^{N} \exp \left( - \left( E_{g_i}(g_i) + E_{f_i}(f_i) \right) \right) \]
Linear Chain CRFs for Part of Speech Tagging

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

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
Linear Chain CRFs for Part of Speech Tagging

\[ p(z_1, z_2, \ldots, z_N \mid x_{1:N}) \propto \prod_{i=1}^{N} \exp(\langle \theta^{(f)}, f_i(z_i) \rangle + \langle \theta^{(g)}, g_i(z_i, z_{i+1}) \rangle) \]
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:

\[ g_{j,N \rightarrow V}(z_j, z_{j+1}) = 1 \text{ (if } z_j = N \& z_{j+1} = V) \text{ else 0} \]
\[ g_{j,told,N \rightarrow V}(z_j, z_{j+1}) = 1 \text{ (if } z_j = N \& z_{j+1} = V \& x_j = told) \text{ else 0} \]
(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 $\leftrightarrow$ (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

\[
\mathbb{E}_{y'} \left[ \sum_i h_i(y') \right] = \sum_i \sum_{y_{i-1}, y_i} p(y_{i-1}, y_i | x_{1:N}) h_i(y_{i-1}, y_i)
\]
• Run BP to compute beliefs (unnormalized, joint marginals)

\[ p(y_{i-1}, y_i | x_{1:N}) \propto g_{i-1}(y_{i-1}, y_i) \times q_{y_{i-1}\rightarrow g_{i-1}}(y_{i-1}) \times q_{y_i\rightarrow g_{i-1}}(y_{i}) \]
Outline

Message Passing: Graphical Model Inference

Example: Linear Chain CRF