Variational Inference for Structured NLP Models

Berkeley NLP

ACL, August 4, 2013

David Burkett and Dan Klein

Berkeley NLP

Tutorial Outline

1. Structured Models and Factor Graphs
2. Mean Field
3. Structured Mean Field
4. Belief Propagation
5. Structured Belief Propagation
6. Wrap-Up

Part 1: Structured Models and Factor Graphs

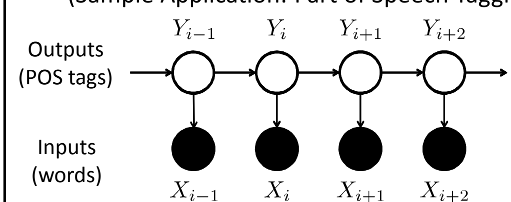
Berkeley NLP

Berkeley NLP

Structured NLP Models

Example: Hidden Markov Model

(Sample Application: Part of Speech Tagging)

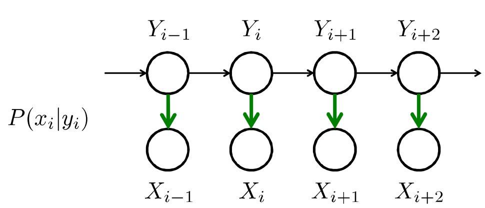


Goal: Queries from posterior $P(Y = y|X = x)$ ($P(y|x)$)

Berkeley NLP

Structured NLP Models

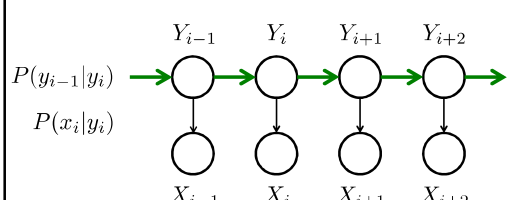
Example: Hidden Markov Model



Berkeley NLP

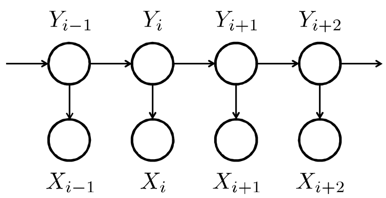
Structured NLP Models

Example: Hidden Markov Model



Structured NLP Models

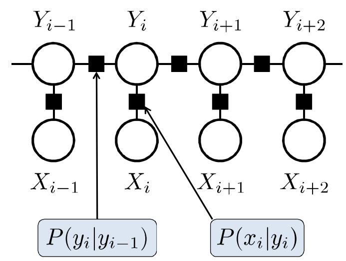
Example: Hidden Markov Model



$$P(y|x) \propto \prod_i P(y_i|y_{i-1})P(x_i|y_i)$$

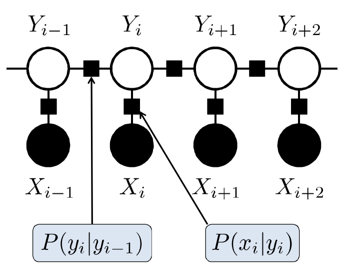
Structured NLP Models

Example: Hidden Markov Model



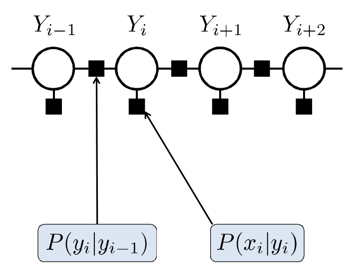
Structured NLP Models

Example: Hidden Markov Model



Structured NLP Models

Example: Hidden Markov Model

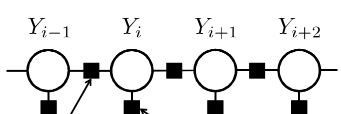


Factor Graph Notation

Variables Y_i

Factors ϕ

Cliques c



Binary Factor

$\phi_c(y_{i-1}, y_i) = P(y_i|y_{i-1})$

$c = \{i-1, i\}$

Unary Factor

$\phi_c(y_i) = P(x_i|y_i)$

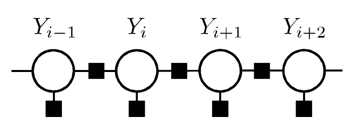
$c = \{i\}$

Factor Graph Notation

Variables Y_i

Factors ϕ

Cliques c



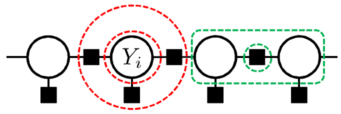
$$P(y|x) \propto \prod_c \phi_c(y_c) = \prod_i P(y_i|y_{i-1})P(x_i|y_i)$$

Factor Graph Notation

Variables Y_i

Factors ϕ

Cliques c



Variables have factor (clique) neighbors:

$\mathcal{N}(i) = \{c : i \in c\}$

Factors have variable neighbors:

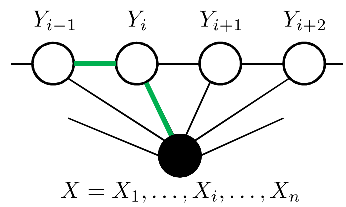
$\mathcal{N}(\phi_c) = c$

Structured NLP Models

(Lafferty et al., 2001)

Example: Conditional Random Field

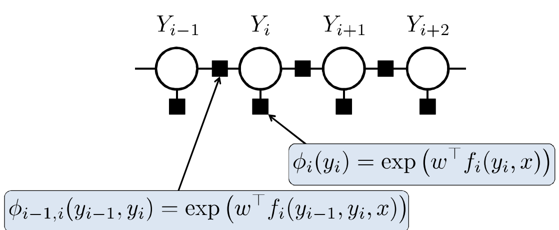
(Sample Application: Named Entity Recognition)



$$P(y|x) \propto \exp\left(\sum_i w^\top f_i(y_i, x) + w^\top f_i(y_{i-1}, y_i, x)\right)$$

Structured NLP Models

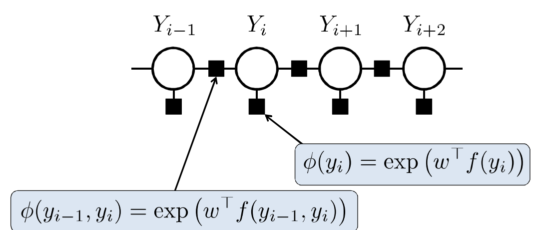
Example: Conditional Random Field



$$P(y|x) \propto \exp\left(\sum_i w^\top f_i(y_i, x) + w^\top f_i(y_{i-1}, y_i, x)\right)$$

Structured NLP Models

Example: Conditional Random Field

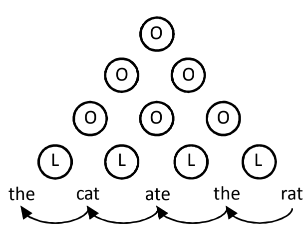


$$P(y|x) \propto \exp\left(\sum_i w^\top f(y_i) + w^\top f(y_{i-1}, y_i)\right)$$

Structured NLP Models

Example: Edge-Factored Dependency Parsing

$y_{ij} \in \{\text{left, right, off}\}$

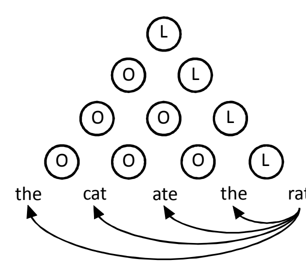


(McDonald et al., 2005)

Structured NLP Models

Example: Edge-Factored Dependency Parsing

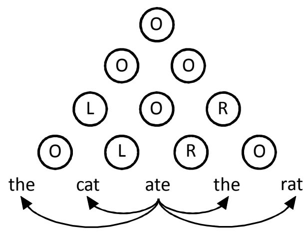
$y_{ij} \in \{\text{left, right, off}\}$



Structured NLP Models

Example: Edge-Factored Dependency Parsing

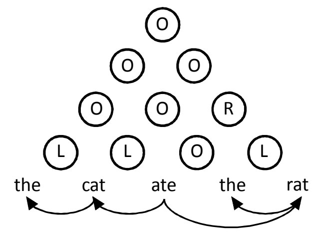
$y_{ij} \in \{\text{left}, \text{right}, \text{off}\}$



Structured NLP Models

Example: Edge-Factored Dependency Parsing

$y_{ij} \in \{\text{left}, \text{right}, \text{off}\}$

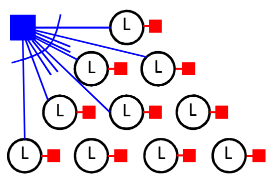


Structured NLP Models

Example: Edge-Factored Dependency Parsing

$y_{ij} \in \{\text{left}, \text{right}, \text{off}\}$

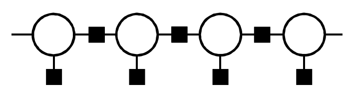
$$\phi(y) = \begin{cases} 1 & y \text{ forms a tree} \\ 0 & \text{otherwise} \end{cases}$$



$$\phi(y_{ij}) = \begin{cases} \exp(w^\top f(i,j)) & y_{ij} = \text{left} \\ \exp(w^\top f(j,i)) & y_{ij} = \text{right} \\ 1 & y_{ij} = \text{off} \end{cases}$$

Inference

- Input: Factor Graph



- Output: Marginals $P(y_i|x)$

Inference

- Typical NLP Approach: Dynamic Programs!
- Examples:
  - Sequence Models (Forward/Backward)
  - Phrase Structure Parsing (CKY, Inside/Outside)
  - Dependency Parsing (Eisner algorithm)
  - ITG Parsing (Bitext Inside/Outside)

Complex Structured Models

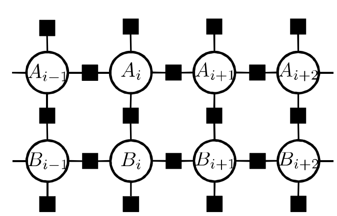
POS Tagging

↑

Joint

↓

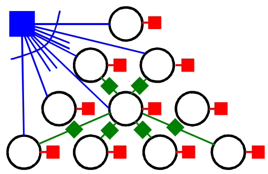
Named Entity Recognition



(Sutton et al., 2004)

Complex Structured Models

Dependency Parsing
with Second Order Features

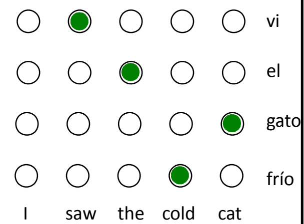


(McDonald & Pereira, 2006)
(Carreras, 2007)

Complex Structured Models

Word Alignment

$y_{ij} \in \{\text{on}, \text{off}\}$



(Taskar et al., 2005)

Complex Structured Models

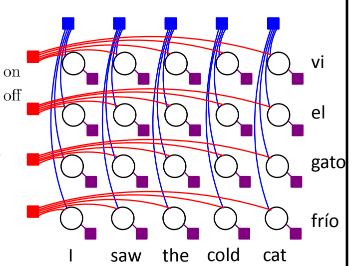
Word Alignment

$y_{ij} \in \{\text{on}, \text{off}\}$

$$\phi(y_{ij}) = \begin{cases} \exp(w^\top f(i,j)) & y_{ij} = \text{on} \\ 1 & y_{ij} = \text{off} \end{cases}$$

$$\phi(y_{i*}) = \begin{cases} 1 & |\{j : y_{ij} = \text{on}\}| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\phi(y_{*j}) = \begin{cases} 1 & |\{i : y_{ij} = \text{on}\}| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

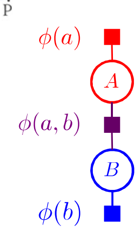


Variational Inference

- Approximate inference techniques that can be applied to any graphical model
- This tutorial:
  - Mean Field: Approximate the joint distribution with a product of marginals
  - Belief Propagation: Apply tree inference algorithms even if your graph isn't a tree
  - Structure: What changes when your factor graph has tractable substructures

Part 2: Mean Field

Mean Field Warmup



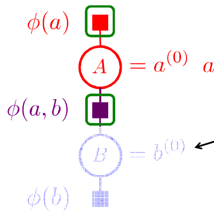
Wanted: $\operatorname*{argmax}_{a,b} P(a,b|x)$

Idea: coordinate ascent

Key object: assignments

Iterated Conditional Modes (Besag, 1986)

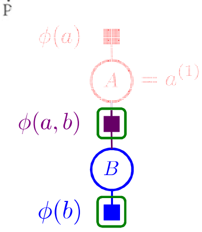
Mean Field Warmup



$a^{(1)} = \arg\max_a \phi(a)\phi(a,b)$

Wanted: $\arg\max_{a,b} P(a,b|x)$

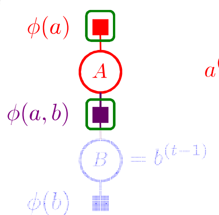
Mean Field Warmup



$b^{(1)} = \arg\max_b \phi(b)\phi(a,b)$

Wanted: $\arg\max_{a,b} P(a,b|x)$

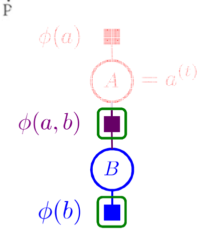
Mean Field Warmup



$a^{(t)} = \arg\max_a \phi(a)\phi(a,b)$

Wanted: $\arg\max_{a,b} P(a,b|x)$

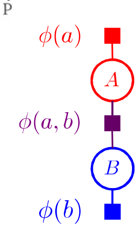
Mean Field Warmup



$b^{(t)} = \arg\max_b \phi(b)\phi(a,b)$

Wanted: $\arg\max_{a,b} P(a,b|x)$

Mean Field Warmup



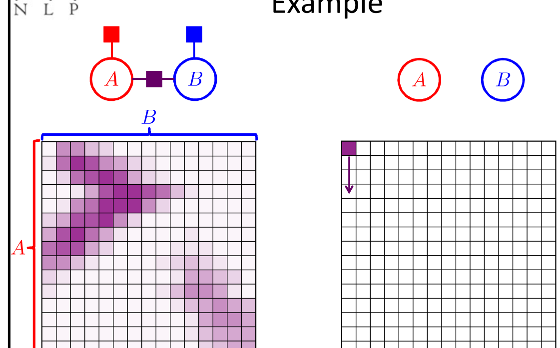
$a^{(t)} = a^{(t-1)}$

$b^{(t)} = b^{(t-1)}$

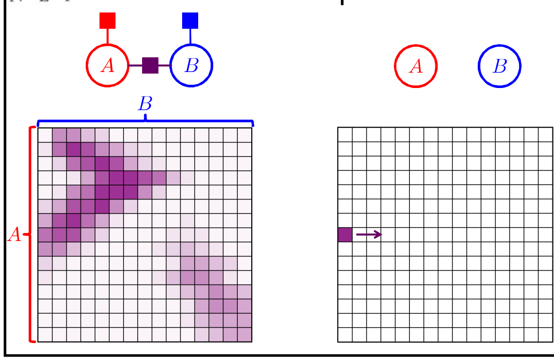
Wanted: $\arg\max_{a,b} P(a,b|x)$

Approximate Result: $(a^{(t)}, b^{(t)})$

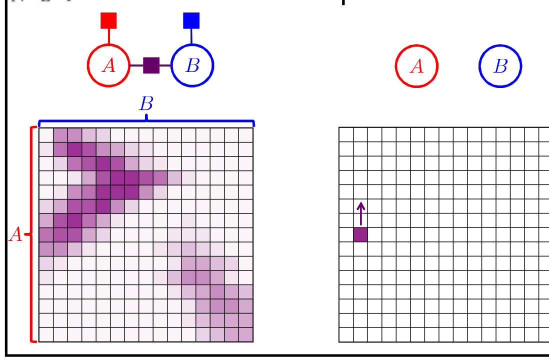
Iterated Conditional Modes Example



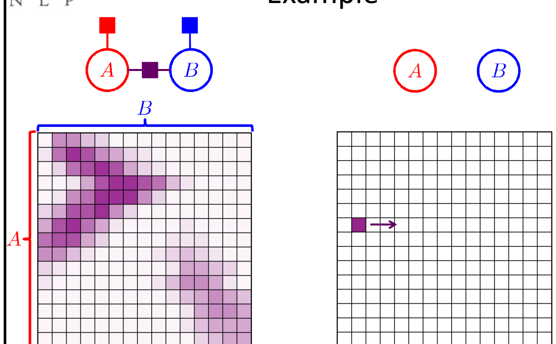
Iterated Conditional Modes Example



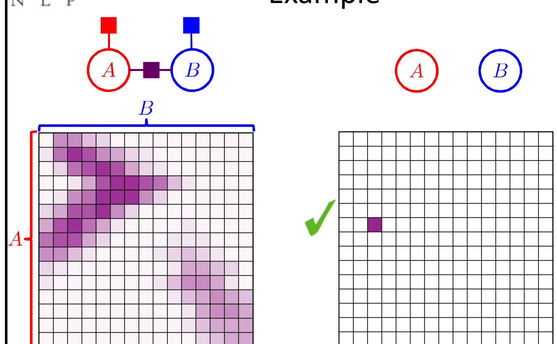
Iterated Conditional Modes Example



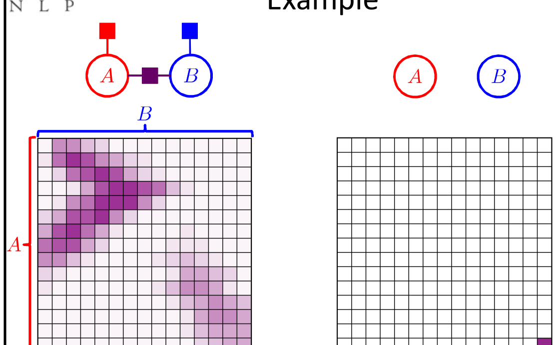
Iterated Conditional Modes Example



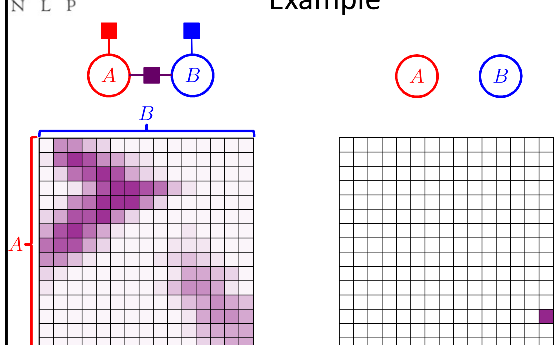
Iterated Conditional Modes Example



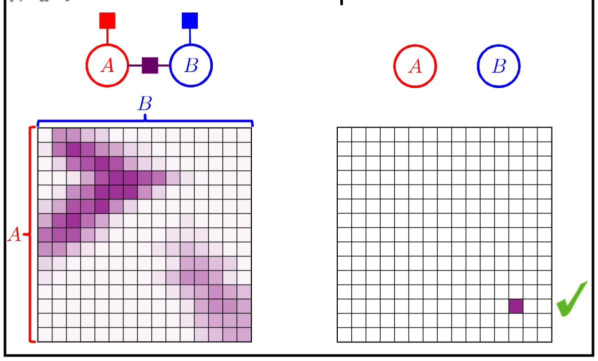
Iterated Conditional Modes Example



Iterated Conditional Modes Example



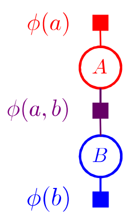
Iterated Conditional Modes Example



Mean Field Intro

Mean Field is coordinate ascent, just like Iterated Conditional Modes, but with soft assignments to each variable!

Mean Field Intro

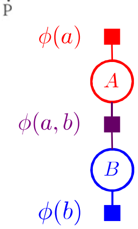


Wanted: $P(a|x), P(b|x)$

Idea: coordinate ascent

Key object: (approx) marginals

Mean Field Intro



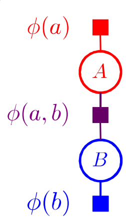
$= \exp(w^\top f(a))$

$= \exp(w^\top f(a,b))$

$= \exp(w^\top f(b))$

$$P(a,b|x) \propto \phi(a)\phi(b)\phi(a,b)$$

Mean Field Intro



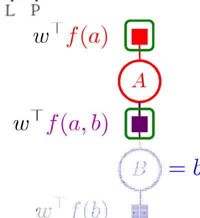
$= \exp(w^\top f(a))$

$= \exp(w^\top f(a,b))$

$= \exp(w^\top f(b))$

$$P(a,b|x) \propto \phi(a)\phi(b)\phi(a,b)$$
$$= \exp(w^\top f(a) + w^\top f(b) + w^\top f(a,b))$$

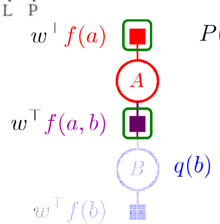
Mean Field Intro



$$P(a|b,x) \propto \exp(w^\top f(a) + w^\top f(a,b))$$

Wanted: $P(a|x), P(b|x)$

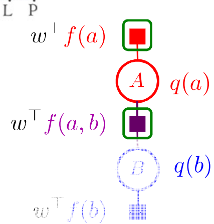
Mean Field Intro



$$P(a|b,x) \propto \exp(w^\top f(a) + w^\top f(a,b))$$

Wanted: $P(a|x), P(b|x)$

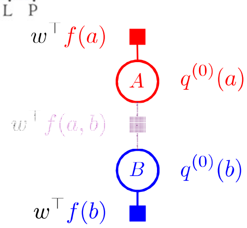
Mean Field Intro



$$q(a) \propto \exp(w^\top f(a) + w^\top \mathbb{E}_{q(b)} f(a,b))$$

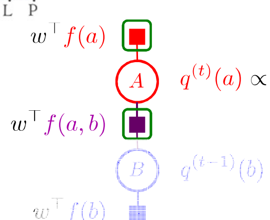
Wanted: $P(a|x), P(b|x)$

Mean Field Procedure



Wanted: $P(a|x), P(b|x)$

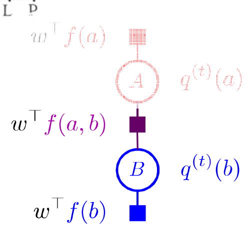
Mean Field Procedure



$$q^{(t)}(a) \propto \exp(w^\top f(a) + w^\top \mathbb{E}_{q(b)} f(a,b))$$

Wanted: $P(a|x), P(b|x)$

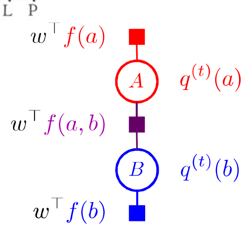
Mean Field Procedure



$$q^{(t)}(b) \propto \exp(w^\top f(b) + w^\top \mathbb{E}_{q(a)} f(a,b))$$

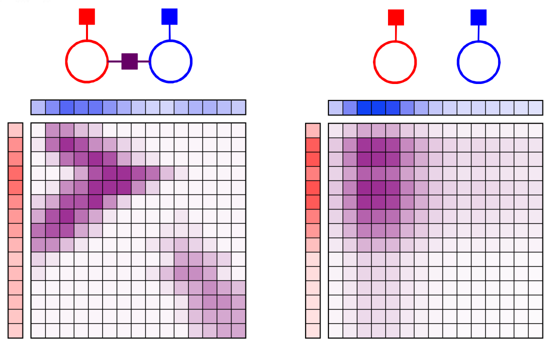
Wanted: $P(a|x), P(b|x)$

Mean Field Procedure



Wanted: $P(a|x), P(b|x)$

Example Results



Mean Field Derivation

- Goal: $p(y) = P(y|x) \propto \exp\left(\sum_c w^\top f(y_c)\right)$
- Approximation: $q(y) \approx p(y)$
- Constraint: $q(y) = \prod_i q(y_i)$
- Objective: $q(y) = \underset{q}{\operatorname{argmin}}\, KL(q||p)$
- Procedure: Coordinate ascent on each $q(y_i)$
- What's the update?

Mean Field Update

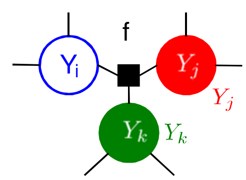
1) $q(y_i) = \underset{q(y_i)}{\operatorname{argmin}}\, KL(q||p)$

2) $\dfrac{\partial KL(q||p)}{\partial q(y_i)} = 0$

3-9) Lots of algebra

10) $q(y_i) \propto \exp\left(\sum_{c \in \mathcal{N}(i)} w^\top \mathbb{E}_{q(y_{-i})} f_c(y_c)\right)$

Approximate Expectations



$$\mathbb{E}_{q(y_{-i})} f(y_i, y_j, y_k) = \sum_{y_j} \sum_{y_k} q(y_j) q(y_k) f(y_i, y_j, y_k)$$

General: $\mathbb{E}_{q(y_{-i})} f_c(y_c) = \sum_{y_{c \setminus \{i\}}} \left(\prod_{j \in c \setminus \{i\}} q(y_j) \right) f_c(y_c)$

General Update *

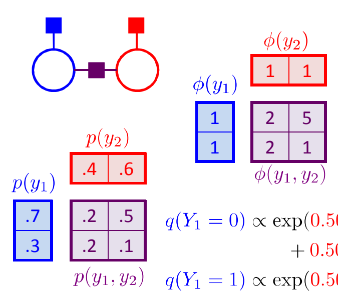
Exponential Family:

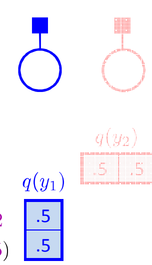
$$q(y_i) \propto \exp\left(\sum_{c \in \mathcal{N}(i)} w^\top \mathbb{E}_{q(y_{-i})} f_c(y_c)\right)$$

Generic:

$$q(y_i) \propto \exp\left(\sum_{c \in \mathcal{N}(i)} \mathbb{E}_{q(y_{-i})} \log \phi_c(y_c)\right)$$

Mean Field Inference Example

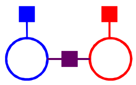
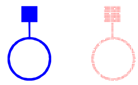




$q(Y_1 = 0) \propto \exp(0.50 \log 2$
$+ 0.50 \log 5)$

$q(Y_1 = 1) \propto \exp(0.50 \log 2$
$+ 0.50 \log 1)$

Mean Field Inference Example

$\phi(y_1)$: 1, 1 — $\phi(y_2)$: 1, 1

$\phi(y_1, y_2)$:

2	5
2	1

$p(y_1)$: .7, .3 — $p(y_2)$: .4, .6

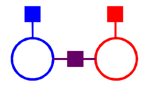
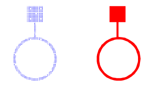
$p(y_1, y_2)$:

.2	.5
.2	.1

$q(y_1)$: .69, .31 — $q(y_2)$: .5, .5

$$q(Y_1 = 0) \propto \exp(0.50 \log 2 + 0.50 \log 5)$$
$$q(Y_1 = 1) \propto \exp(0.50 \log 2 + 0.50 \log 1)$$

Mean Field Inference Example

$\phi(y_1)$: 1, 1 — $\phi(y_2)$: 1, 1

$\phi(y_1, y_2)$:

2	5
2	1

$p(y_1)$: .7, .3 — $p(y_2)$: .4, .6

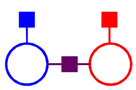
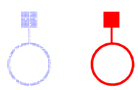
$p(y_1, y_2)$:

.2	.5
.2	.1

$q(y_1)$: .69, .31 — $q(y_2)$: .5, .5

$$q(Y_2 = 0) \propto \exp(0.69 \log 2 + 0.31 \log 2)$$
$$q(Y_2 = 1) \propto \exp(0.69 \log 5 + 0.31 \log 1)$$

Mean Field Inference Example

$\phi(y_1)$: 1, 1 — $\phi(y_2)$: 1, 1

$\phi(y_1, y_2)$:

2	5
2	1

$p(y_1)$: .7, .3 — $p(y_2)$: .4, .6

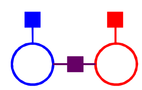
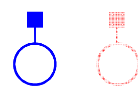
$p(y_1, y_2)$:

.2	.5
.2	.1

$q(y_1)$: .69, .31 — $q(y_2)$: .40, .60

$$q(Y_2 = 0) \propto \exp(0.69 \log 2 + 0.31 \log 2)$$
$$q(Y_2 = 1) \propto \exp(0.69 \log 5 + 0.31 \log 1)$$

Mean Field Inference Example

$\phi(y_1)$: 1, 1 — $\phi(y_2)$: 1, 1

$\phi(y_1, y_2)$:

2	5
2	1

$p(y_1)$: .7, .3 — $p(y_2)$: .4, .6

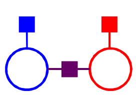
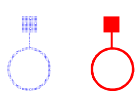
$p(y_1, y_2)$:

.2	.5
.2	.1

$q(y_1)$: .73, .27 — $q(y_2)$: .40, .60

$$q(Y_1 = 0) \propto \exp(0.40 \log 2 + 0.60 \log 5)$$
$$q(Y_1 = 1) \propto \exp(0.40 \log 2 + 0.60 \log 1)$$

Mean Field Inference Example

$\phi(y_1)$: 1, 1 — $\phi(y_2)$: 1, 1

$\phi(y_1, y_2)$:

2	5
2	1

$p(y_1)$: .7, .3 — $p(y_2)$: .4, .6

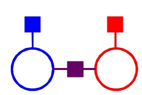
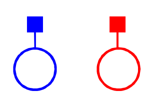
$p(y_1, y_2)$:

.2	.5
.2	.1

$q(y_1)$: .73, .27 — $q(y_2)$: .38, .62

$$q(Y_2 = 0) \propto \exp(0.73 \log 2 + 0.27 \log 2)$$
$$q(Y_2 = 1) \propto \exp(0.73 \log 5 + 0.27 \log 1)$$

Mean Field Inference Example

$\phi(y_1)$: 1, 1 — $\phi(y_2)$: 1, 1

$\phi(y_1, y_2)$:

2	5
2	1

$p(y_1)$: .7, .3 — $p(y_2)$: .4, .6

$p(y_1, y_2)$:

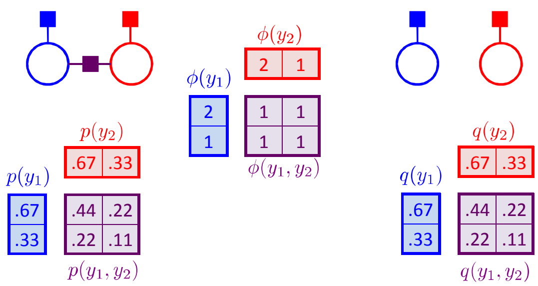
.2	.5
.2	.1

$q(y_1)$: .73, .27 — $q(y_2)$: .38, .62

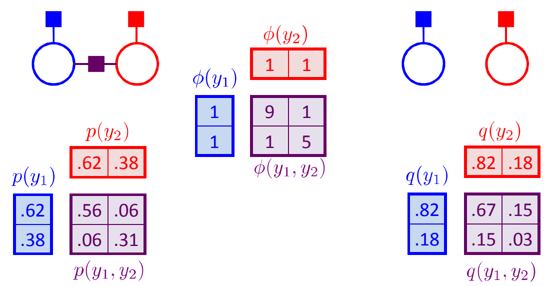
$q(y_1, y_2)$:

.28	.45
.10	.17

Mean Field Inference Example



Mean Field Inference Example



Mean Field Q&A

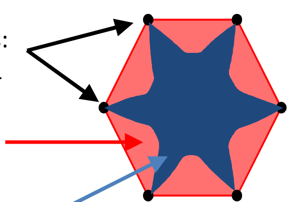
- Are the marginals guaranteed to converge to the right thing, like in sampling?
  No
- Is the algorithm at least guaranteed to converge to something?
  Yes
- So it's just like EM?
  Yes

Why Only Local Optima?!

Variables: $Y_1, Y_2, \ldots Y_n$

Discrete distributions:
e.g. P(0,1,0,...0) = 1

All distributions
(all convex combos)

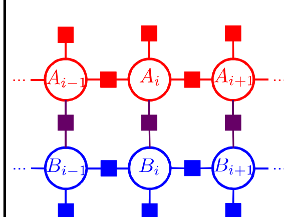


Mean field approximable
(can represent all discrete ones, but not all)

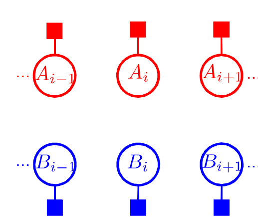
Part 3: Structured Mean Field

Mean Field Approximation

Model:



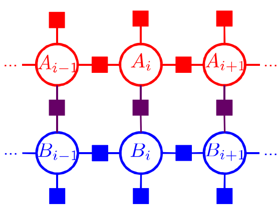
Approximate Graph:



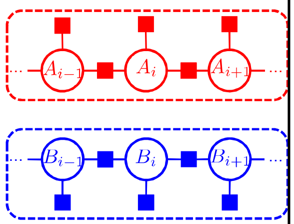
$$p(a, b) \approx \prod_i q(a_i) q(b_i)$$

Structured Mean Field Approximation

Model:



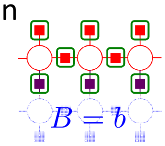
Approximate Graph:



$$p(a, b) \approx q(a)q(b)$$

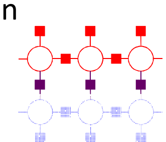
(Xing et al, 2003)

Structured Mean Field Approximation



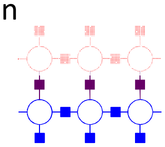
$$P(a|b, x) \propto \exp\left(\sum_i w^\top f(a_i) + \sum_i w^\top f(a_{i-1}, a_i) + \sum_i w^\top f(a_i, b_i)\right)$$

Structured Mean Field Approximation



$$q(a) \propto \exp\left(\sum_i w^\top f(a_i) + \sum_i w^\top f(a_{i-1}, a_i) + \sum_i w^\top \mathbb{E}_{q(b)} f(a_i, b_i)\right)$$

Structured Mean Field Approximation



$$q(b) \propto \exp\left(\sum_i w^\top f(b_i) + \sum_i w^\top f(b_{i-1}, b_i) + \sum_i w^\top \mathbb{E}_{q(a)} f(a_i, b_i)\right)$$

Computing Structured Updates

$$q(a) \propto \exp\left(\sum_i w^\top f(a_i) + \sum_i w^\top f(a_{i-1}, a_i) + \sum_i w^\top \mathbb{E}_{q(b)} f(a_i, b_i)\right)$$

w ✓

$f(a_i)$ ✓

$f(a_{i-1}, a_i)$ ✓

$\mathbb{E}_{q(b)} f(a_i, b_i)$??

Computing Structured Updates

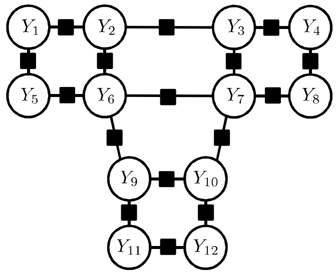
$$\mathbb{E}_{q(b)} f(a_i, b_i) = \sum_b q(b) f(a_i, b_i)$$
$$= \sum_{b_i} q(b_i) f(a_i, b_i)$$

Marginal probability of b_i under $q(b)$

Computed with forward-backward

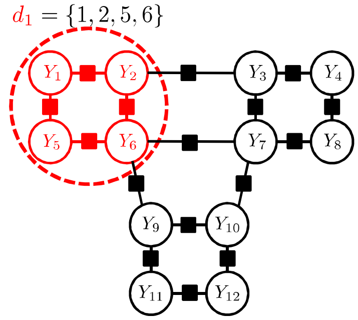
Updating $q(a)$ consists of computing all marginals $q(b_i)$

Structured Mean Field Notation



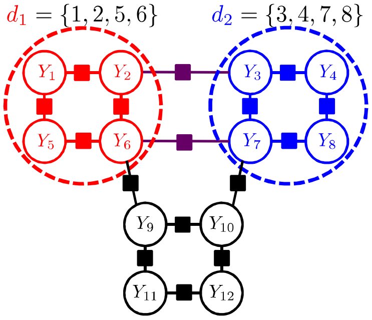
Structured Mean Field Notation

$d_1 = \{1, 2, 5, 6\}$



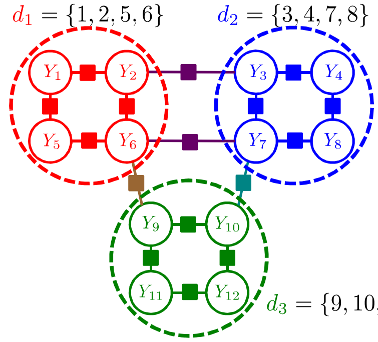
Structured Mean Field Notation

$d_1 = \{1, 2, 5, 6\}$ $d_2 = \{3, 4, 7, 8\}$



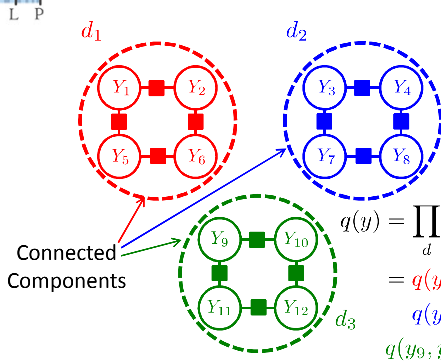
Structured Mean Field Notation

$d_1 = \{1, 2, 5, 6\}$ $d_2 = \{3, 4, 7, 8\}$



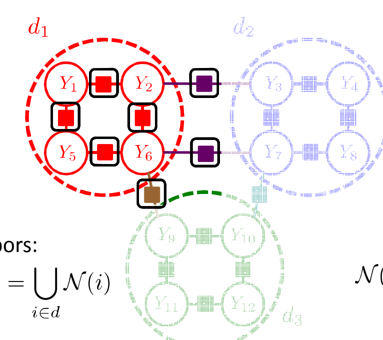
$d_3 = \{9, 10, 11, 12\}$

Structured Mean Field Notation



$$q(y) = \prod_d q(y_d)$$
$$= q(y_1, y_2, y_5, y_6) \cdot q(y_3, y_4, y_7, y_8) \cdot q(y_9, y_{10}, y_{11}, y_{12})$$

Structured Mean Field Notation



Neighbors:

$$\mathcal{N}(d) = \bigcup_{i \in d} \mathcal{N}(i)$$

$\mathcal{N}(d_1)$

Structured Mean Field Updates

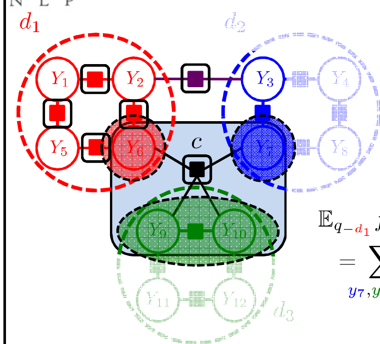
Naïve Mean Field:

$$q(y_i) \propto \exp\left(\sum_{c \in \mathcal{N}(i)} w^\top \mathbb{E}_{q_{-i}} f(y_c)\right)$$

Structured Mean Field:

$$q(y_d) \propto \exp\left(\sum_{c \in \mathcal{N}(d)} w^\top \mathbb{E}_{q_{-d}} f(y_c)\right)$$

Expected Feature Counts



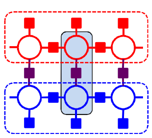
$q(d_1)$

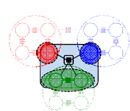
$\mathcal{N}(d_1)$

$$\mathbb{E}_{q_{-d_1}} f(y_6, y_7, y_9, y_{10}) = \sum_{y_7, y_9, y_{10}} f(y_6, y_7, y_9, y_{10})$$

Component Factorizability *

Condition	Example Feature	Generic Condition
$f(a_i, b_i) = f(a_i) f(b_i)$	$f(a_i, b_i) = \begin{cases} 1 & a_i = \text{NNP} \ \& \\ & b_i = \text{B-PER} \\ 0 & \text{otherwise} \end{cases}$ $= \left(\begin{cases} 1 & a_i = \text{NNP} \\ 0 & \text{otherwise} \end{cases}\right) \cdot \left(\begin{cases} 1 & b_i = \text{B-PER} \\ 0 & \text{otherwise} \end{cases}\right)$ $= f(a_i) f(b_i)$	$f_c(y_c) = \prod_{d: c \cap d \neq \emptyset} f_{c \cap d}(y_{c \cap d})$ (pointwise product)



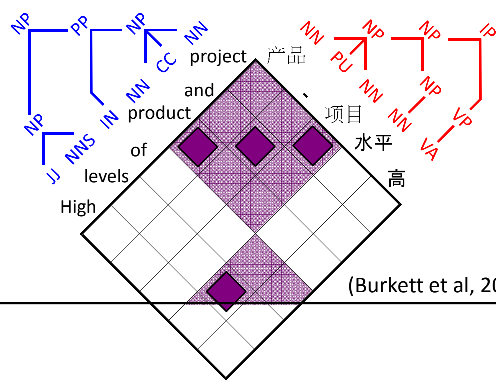


Component Factorizability *

(Abridged)

Use conjunctive indicator features

Joint Parsing and Alignment



(Burkett et al, 2010)

Joint Parsing and Alignment

Input:

Sentences

(s, s')

project 产品

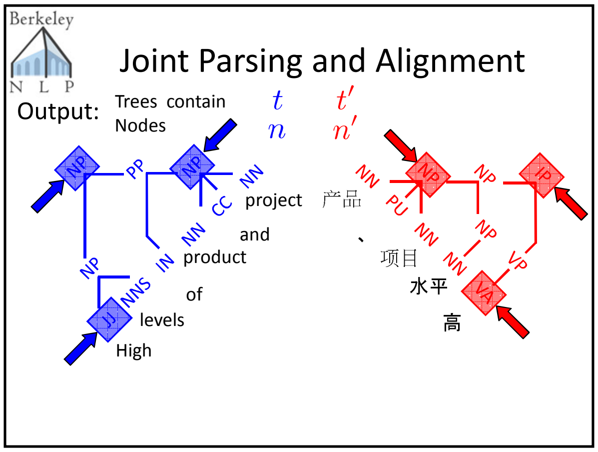
and 、

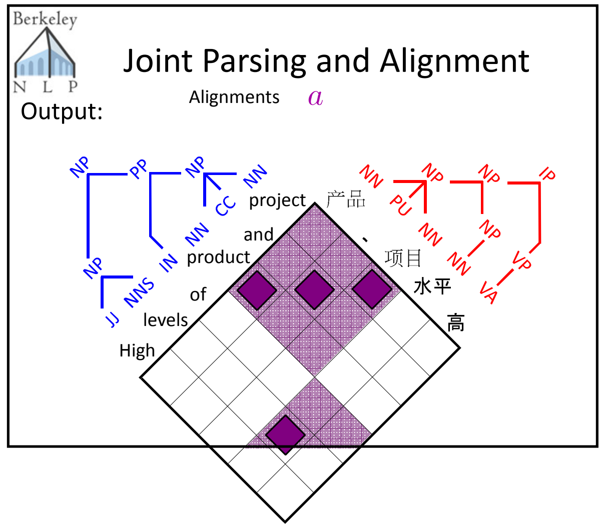
product 项目

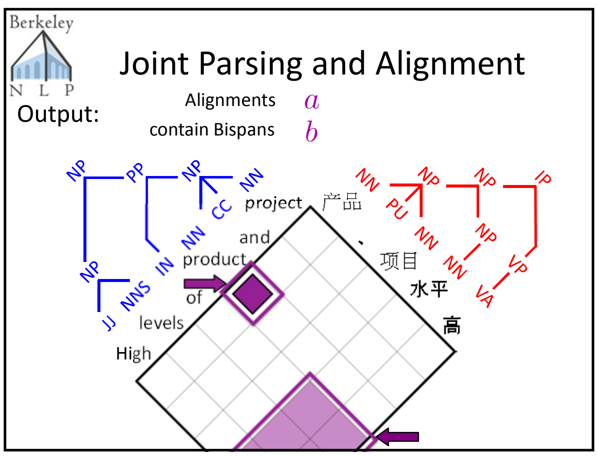
of 水平

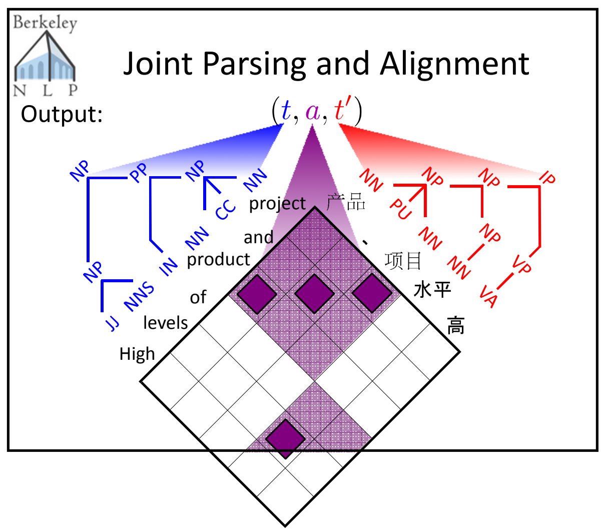
levels 高

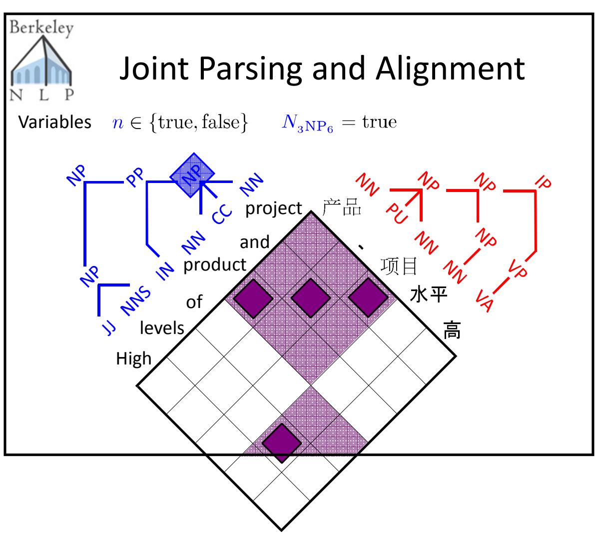
High

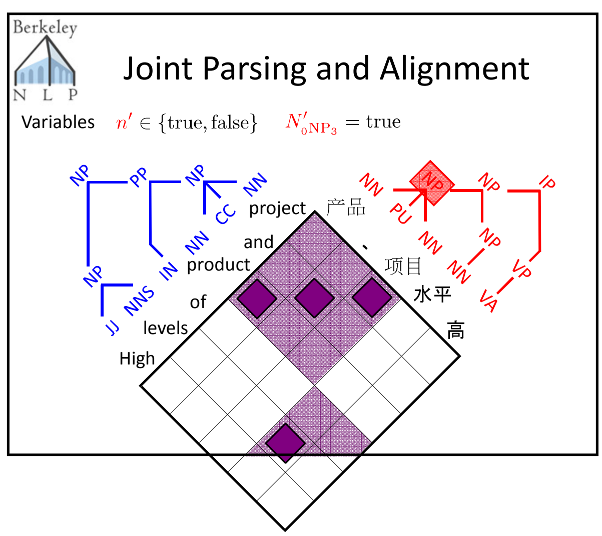

Berkeley N L P
Joint Parsing and Alignment
Output:
Trees contain t t'
Nodes n n'
NP PP NP NN CC NN IN NP NNS JJ
project and product of levels High
NN PU NN NN NP NP NP IP VP VA
产品 、 项目 水平 高


Berkeley N L P
Joint Parsing and Alignment
Output:
Alignments a
project and product of levels High
产品 、 项目 水平 高


Berkeley N L P
Joint Parsing and Alignment
Output:
Alignments a
contain Bispans b
project and product of levels High
产品 、 项目 水平 高

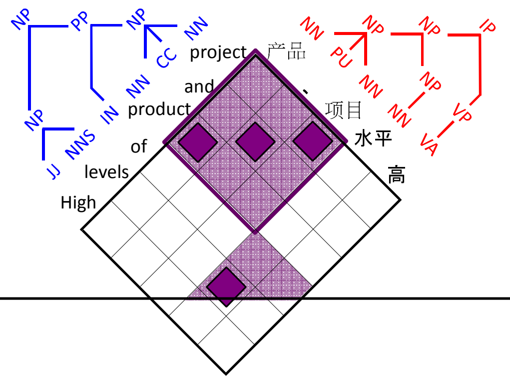

Berkeley N L P
Joint Parsing and Alignment
Output:
(t, a, t')
project and product of levels High
产品 、 项目 水平 高


Berkeley N L P
Joint Parsing and Alignment
Variables $n \in \{\text{true}, \text{false}\}$ $N_{{}_3\text{NP}_6} = \text{true}$
project and product of levels High
产品 、 项目 水平 高


Berkeley N L P
Joint Parsing and Alignment
Variables $n' \in \{\text{true}, \text{false}\}$ $N'_{{}_0\text{NP}_3} = \text{true}$
project and product of levels High
产品 、 项目 水平 高

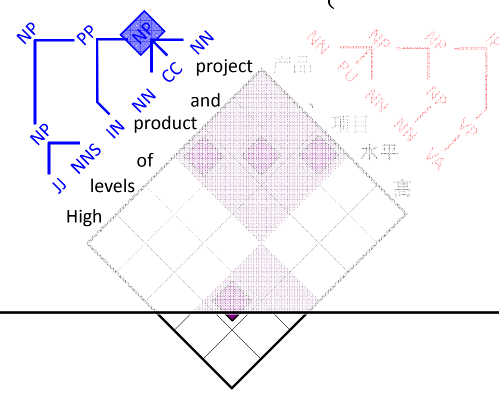
Joint Parsing and Alignment

Variables $b \in \{\text{true}, \text{false}\}$ $\quad B_{36,03} = \text{true}$



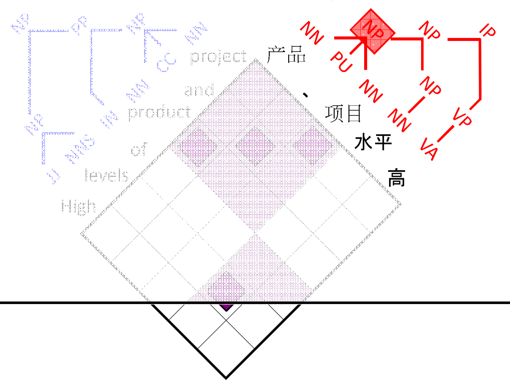
Joint Parsing and Alignment

Factors $\phi(n) = \exp(w^\top f_t(n))$ $\quad \phi(t) = \begin{cases} 1 & t \text{ forms a tree} \\ 0 & \text{otherwise} \end{cases}$



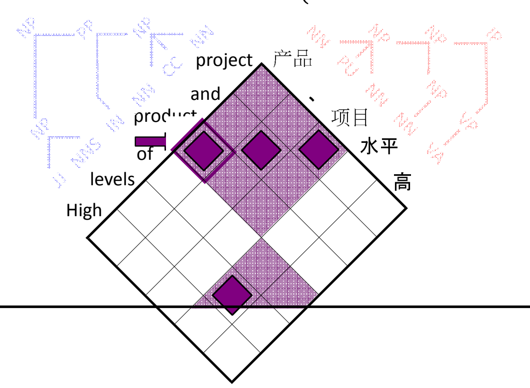
Joint Parsing and Alignment

Factors $\phi(n') = \exp(w^\top f_{t'}(n'))$ $\quad \phi(t') = \begin{cases} 1 & t' \text{ forms a tree} \\ 0 & \text{otherwise} \end{cases}$



Joint Parsing and Alignment

Factors $\phi(b) = \exp(w^\top f_a(b))$ $\quad \phi(a) = \begin{cases} 1 & a \text{ is an ITG derivation} \\ 0 & \text{otherwise} \end{cases}$

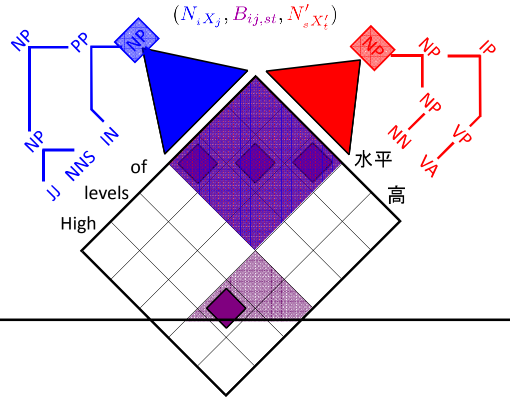


Joint Parsing and Alignment

Factors

$$\phi(n, b, n') = \exp(w^\top f_{tat'}(n, b, n'))$$

$$(N_{{}_iX_j}, B_{ij,st}, N'_{{}_sX'_t})$$



Notational Abuse

Subscript Omission:

$f_t(n) = f_t(n_{{}_iX_j})$

Shorthand:

$n \in t \quad \Leftrightarrow \quad N_{{}_iX_j} = \text{true}$

$n \rhd b \lhd n' \quad \Leftrightarrow \quad n \in t \ \ \& \ \ b \in a \ \ \& \ \ n' \in t' \ \ \&$

$(N_{{}_iX_j}, B_{ij,st}, N'_{{}_sX'_t})$ match up

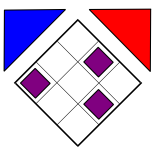
Skip Nonexistent Substructures:

$n \notin t \quad \Rightarrow \quad f_t(n) = 0$

Structural factors $\phi(t), \phi(a), \phi(t')$ are implicit

Model Form

$$P(t, a, t'|s, s') \propto \exp\left(\sum_{n\in t} w^\top f_t(n) + \sum_{b\in a} w^\top f_a(b) + \sum_{n'\in t'} w^\top f_{t'}(n') + \sum_{n \triangleright b \triangleleft n'} w^\top f_{tat'}(n, b, n')\right)$$



Training

Expected Feature Counts	Marginals
$\mathbb{E} f_t(n)$	$P(n \in t \| s, s')$
$\mathbb{E} f_a(b)$	$P(b \in a \| s, s')$
$\mathbb{E} f_{t'}(n')$	$P(n' \in t' \| s, s')$
$\mathbb{E} f_{tat'}(n, b, n')$	$P(n \triangleright b \triangleleft n' \| s, s')$

Structured Mean Field Approximation

$$P(t, a, t'|s, s') \propto \exp\left(\sum_{n\in t} w^\top f_t(n) + \sum_{b\in a} w^\top f_a(b) + \sum_{n'\in t'} w^\top f_{t'}(n') + \sum_{n \triangleright b \triangleleft n'} w^\top f_{tat'}(n, b, n')\right)$$

$$\approx q(t)q(a)q(t')$$

Approximate Component Scores

Monolingual parser:

Score for $n = w^\top f_t(n)$

If we knew (a, t'):

Score for $n = w^\top f_t(n) + w^\top f_{tat'}(n, b, n')$

To compute $q(t)$:

Score for $n = w^\top f_t(n) + w^\top \mathbb{E}_{q(a,t')} f_{tat'}(n, b, n')$

Expected Feature Counts

For fixed $n_{{}_iX_j}$:

$$\mathbb{E}_{q(a,t')} f_{tat'}(n, b, n')$$

$$= \sum_{{}_sX'_t} P_q(n_{{}_iX_j} \triangleright b_{ij,st} \triangleleft n'_{{}_sX'_t}) f_{tat'}(n, b, n')$$

$$= \sum_{{}_sX'_t} q(b_{ij,st}) q(n'_{{}_sX'_t}) f_{tat'}(n, b, n')$$

Marginals computed with bitext inside-outside

Marginals computed with inside-outside

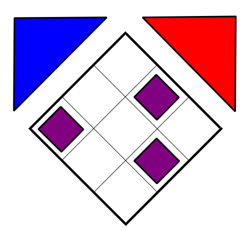
Inference Procedure

Initialize:

$$q(t) \propto \exp\left(\sum_{n\in t} w^\top f_t(n)\right)$$

$$q(a) \propto \exp\left(\sum_{b\in a} w^\top f_a(b)\right)$$

$$q(t') \propto \exp\left(\sum_{n'\in t'} w^\top f_{t'}(n')\right)$$



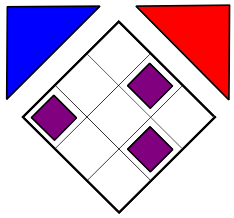
Inference Procedure

Iterate marginal updates:

$q(n)$

$q(b)$

$q(n')$



...until convergence!

Approximate Marginals

$$P(n \in t | s, s') \approx q(n)$$

$$P(b \in a | s, s') \approx q(b)$$

$$P(n' \in t' | s, s') \approx q(n')$$

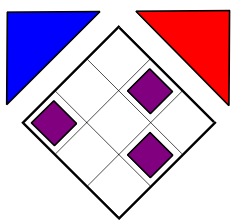
$$P(n \rhd b \lhd n' | s, s') \approx q(n)q(b)q(n')$$

Decoding

$$\hat{t} = \operatorname*{argmax}_t q(t)$$

$$\hat{a} = \operatorname*{argmax}_a q(a)$$

$$\hat{t'} = \operatorname*{argmax}_{t'} q(t')$$



(Minimum Risk)

Structured Mean Field Summary

- Split the model into pieces you have dynamic programs for
- Substitute expected feature counts for actual counts in cross-component factors
- Iterate computing marginals until convergence

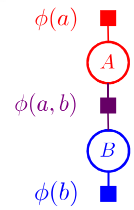
Structured Mean Field Tips

- Try to make sure cross-component features are products of indicators
- You don't have to run all the way to convergence; marginals are usually pretty good after just a few rounds
- Recompute marginals for fast components more frequently than for slow ones
  - e.g. For joint parsing and alignment, the two monolingual tree marginals ($O(n^3)$) were updated until convergence between each update of the ITG marginals ($O(n^6)$)

Break Time!

Part 4: Belief Propagation

Belief Propagation



Wanted: $P(a|x)$, $P(b|x)$

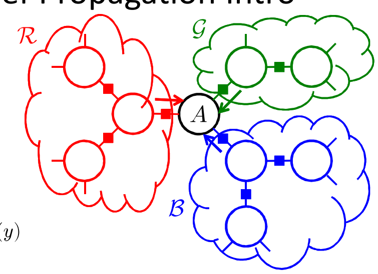
Idea: pretend graph is a tree

Key objects:

- Beliefs (marginals)
- Messages

Belief Propagation Intro

Assume we have a tree



$$P(a|x) \propto \sum_{y \setminus \{a\}} score(y)$$

$$= \left(\sum_{\mathcal{R}} score(\mathcal{R}, a)\right)\left(\sum_{\mathcal{G}} score(\mathcal{G}, a)\right)\left(\sum_{\mathcal{B}} score(\mathcal{B}, a)\right)$$

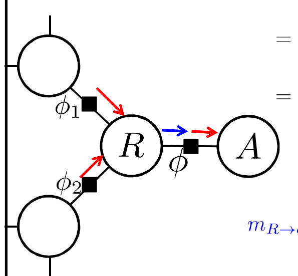
$$= m_{\mathcal{R}\to A}(a) \cdot m_{\mathcal{G}\to A}(a) \cdot m_{\mathcal{B}\to A}(a)$$

Belief Propagation Intro

$$m_{\phi\to A}(a) = \sum_{\mathcal{R}} score(\mathcal{R}, a)$$

$$= \sum_r \phi(r,a) m_{\phi_1\to R}(r) m_{\phi_2\to R}(r)$$

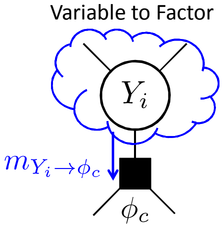
$$= \sum_r \phi(r,a) m_{R\to\phi}(r)$$



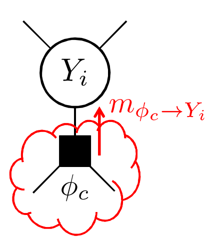
$$m_{R\to\phi}(r) = m_{\phi_1\to R}(r) m_{\phi_2\to R}(r)$$

Messages

Variable to Factor



Factor to Variable

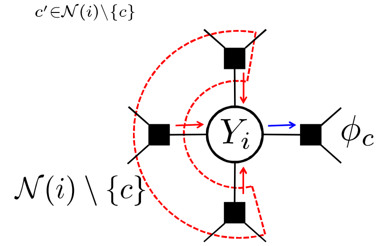


Both take form of "distribution" over Y_i

Messages General Form

- Messages from variables to factors:

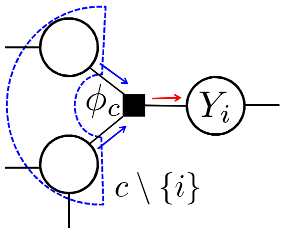
$$m_{Y_i\to\phi_c}(y_i) \propto \prod_{c' \in \mathcal{N}(i)\setminus\{c\}} m_{\phi_{c'}\to Y_i}(y_i)$$



Messages General Form

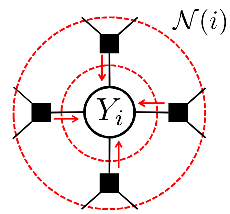
- Messages from factors to variables:

$$m_{\phi_c \to Y_i}(y_i) \propto \sum_{y_{c \setminus \{i\}}} \phi_c(y_c) \prod_{i' \in c \setminus \{i\}} m_{Y_{i'} \to \phi_c}(y_{i'})$$



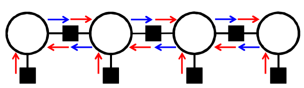
Marginal Beliefs

$$b_{Y_i}(y_i) \propto \prod_{c \in \mathcal{N}(i)} m_{\phi_c \to Y_i}(y_i)$$



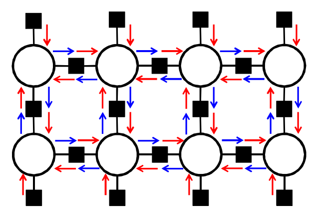
Belief Propagation on Tree-Structured Graphs

- If the factor graph has no cycles, BP is exact
  - Can always order message computations



- After one pass, marginal beliefs are correct

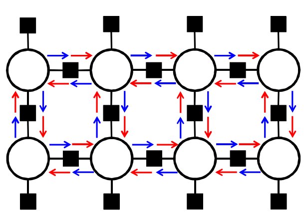
"Loopy" Belief Propagation



Problem: we no longer have a tree

Solution: ignore problem

"Loopy" Belief Propagation



Just start passing messages anyway!

Belief Propagation Q&A

- Are the marginals guaranteed to converge to the right thing, like in sampling?

  No

- Well, is the algorithm at least guaranteed to converge to something, like mean field?

  No

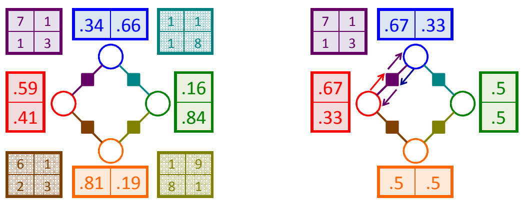
- Will everything often work out more or less OK in practice?

  Maybe

Berkeley NLP

Belief Propagation Example

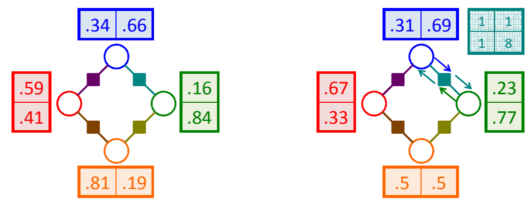
Exact BP



Berkeley NLP

Belief Propagation Example

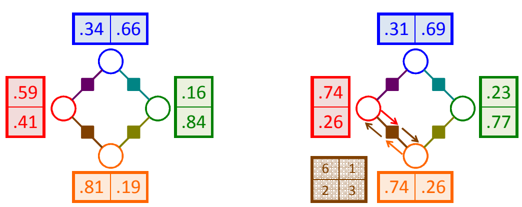
Exact BP



Berkeley NLP

Belief Propagation Example

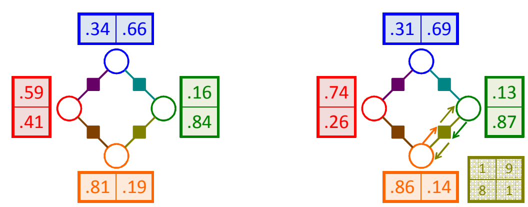
Exact BP



Berkeley NLP

Belief Propagation Example

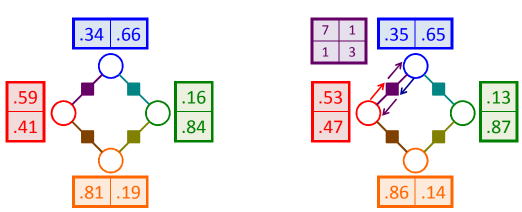
Exact BP



Berkeley NLP

Belief Propagation Example

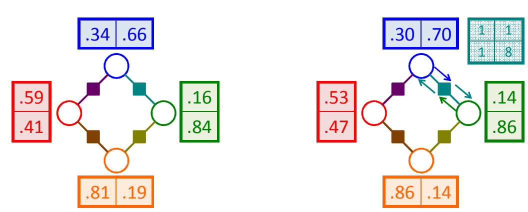
Exact BP



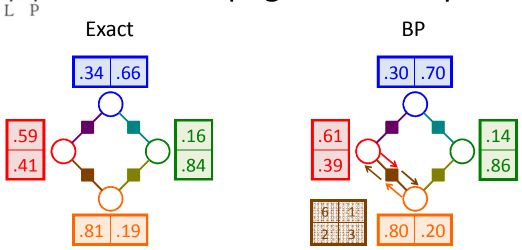
Berkeley NLP

Belief Propagation Example

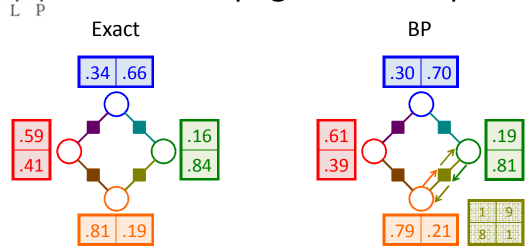
Exact BP



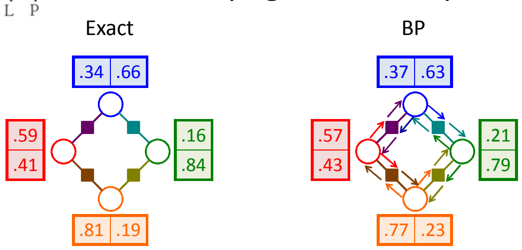
Belief Propagation Example



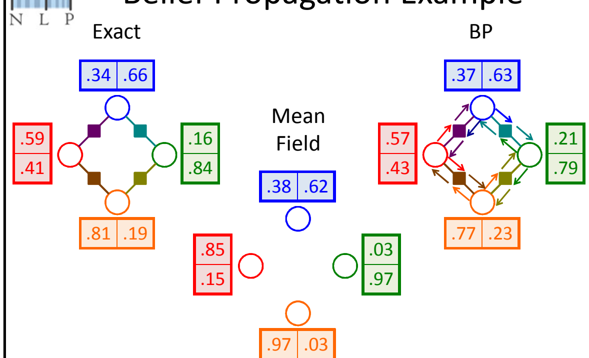
Belief Propagation Example



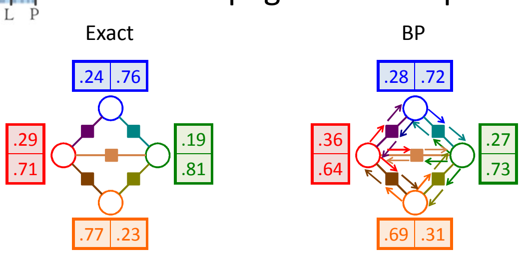
Belief Propagation Example



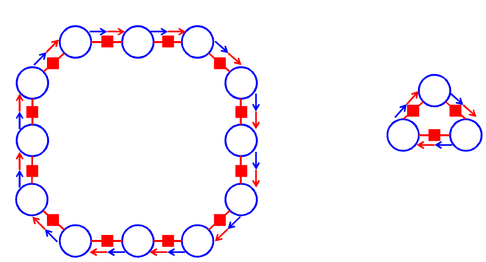
Belief Propagation Example



Belief Propagation Example



Playing Telephone



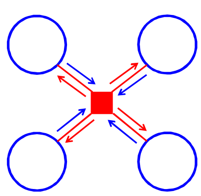
Part 5: Belief Propagation with Structured Factors

Structured Factors

- Problem:
  - Computing factor messages is exponential in arity
  - Many models we care about have high-arity factors
- Solution:
  - Take advantage of NLP tricks for efficient sums
- Examples:
  - Word Alignment (at-most-one constraints)
  - Dependency Parsing (tree constraint)

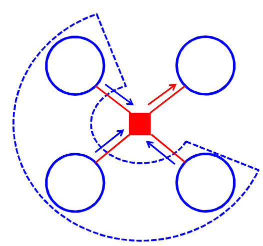
Warm-up Exercise

$$m_{\phi_c \to Y_i}(y_i) \propto \sum_{y_{c \setminus \{i\}}} \phi_c(y_c) \prod_{i' \in c \setminus \{i\}} m_{Y_{i'} \to \phi_c}(y_{i'})$$



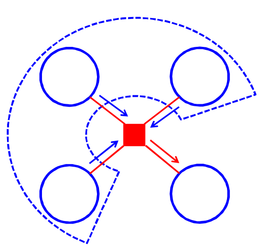
Warm-up Exercise

$$m_{\phi_c \to Y_i}(y_i) \propto \sum_{y_{c \setminus \{i\}}} \phi_c(y_c) \prod_{i' \in c \setminus \{i\}} m_{Y_{i'} \to \phi_c}(y_{i'})$$



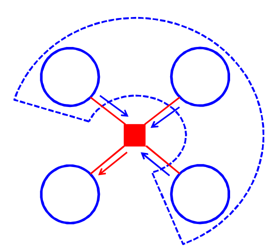
Warm-up Exercise

$$m_{\phi_c \to Y_i}(y_i) \propto \sum_{y_{c \setminus \{i\}}} \phi_c(y_c) \prod_{i' \in c \setminus \{i\}} m_{Y_{i'} \to \phi_c}(y_{i'})$$



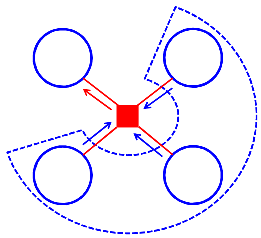
Warm-up Exercise

$$m_{\phi_c \to Y_i}(y_i) \propto \sum_{y_{c \setminus \{i\}}} \phi_c(y_c) \prod_{i' \in c \setminus \{i\}} m_{Y_{i'} \to \phi_c}(y_{i'})$$



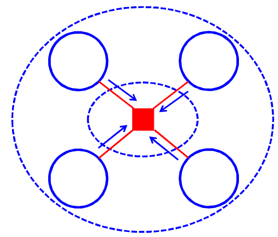
Warm-up Exercise

$$m_{\phi_c \to Y_i}(y_i) \propto \sum_{y_c \setminus \{i\}} \phi_c(y_c) \prod_{i' \in c\setminus\{i\}} m_{Y_{i'} \to \phi_c}(y_{i'})$$



Warm-up Exercise

$$m_{\phi_c \to Y_i}(y_i) \propto \sum_{y_c \setminus \{i\}} \phi_c(y_c) \prod_{i' \in c\setminus\{i\}} m_{Y_{i'} \to \phi_c}(y_{i'})$$

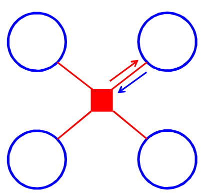


$$s(y_c) = \prod_{i \in c} m_{Y_i \to \phi_c}(y_i)$$

Warm-up Exercise

$$m_{\phi_c \to Y_i}(y_i) \propto \sum_{y_c \setminus \{i\}} \phi_c(y_c) \prod_{i' \in c\setminus\{i\}} m_{Y_{i'} \to \phi_c}(y_{i'})$$

$$= \frac{s(y_c)}{m_{Y_i \to \phi_c}(y_i)}$$



$$s(y_c) = \prod_{i \in c} m_{Y_i \to \phi_c}(y_i)$$

Warm-up Exercise

$$m_{\phi_c \to Y_i}(y_i) \propto \sum_{y_c \setminus \{i\}} \phi_c(y_c) \boxed{\prod_{i' \in c\setminus\{i\}} m_{Y_{i'} \to \phi_c}(y_{i'})}$$

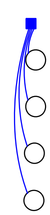
$$= \frac{s(y_c)}{m_{Y_i \to \phi_c}(y_i)}$$

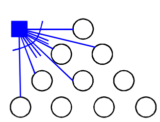
- Benefits:
  - Cleans up notation
  - Saves time multiplying
  - Enables efficient summing

$$\boxed{s(y_c)} = \prod_{i \in c} m_{Y_i \to \phi_c}(y_i)$$

The Shape of Structured BP

- Isolate the combinatorial factors
- Figure out how to compute efficient sums
  - Directly exploiting sparsity
  - Dynamic programming
- Work out the bookkeeping
  - Or, use a reference!





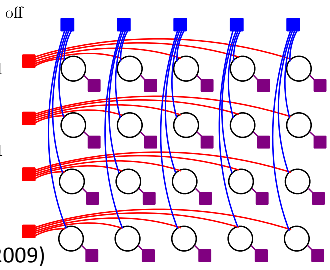
Word Alignment with BP

$$\phi(y_{ij}) = \begin{cases} \exp(w^\top f(i,j)) & y_{ij} = \text{on} \\ 1 & y_{ij} = \text{off} \end{cases}$$

$$y_{ij} \in \{\text{on}, \text{off}\}$$

$$\phi(y_{i*}) = \begin{cases} 1 & |\{j : y_{ij} = \text{on}\}| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\phi(y_{*j}) = \begin{cases} 1 & |\{i : y_{ij} = \text{on}\}| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$



(Cromières & Kurohashi, 2009)
(Burkett & Klein, 2012)

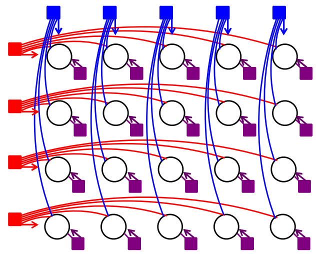
Computing Messages from Factors

Exponential in arity of factor
(have to sum over all assignments)

$\phi_{ij}(y_{ij})$ Arity 1

$\phi_i(y_{i*})$ Arity $O(n)$

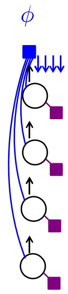
$\phi_j(y_{*j})$ Arity $O(n)$



Computing Constraint Factor Messages

- Input: $m_{Y_j \to \phi}(y_j)\ \forall j$
- Goal: $m_{\phi \to Y_j}(y_j)\ \forall j$

$$m_{\phi \to Y_j}(y_j) \propto \frac{\sum_{y:\ Y_j = y_j} \phi(y) s(y)}{m_{Y_j \to \phi}(y_j)}$$

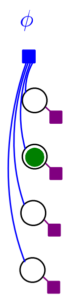


Computing Constraint Factor Messages

$y(j)$: Assignment to variables where $Y_j = \text{on}$

$$y(2) = \{Y_1 = \text{off},\ Y_2 = \text{on},\ Y_3 = \text{off},\ Y_4 = \text{off}\}$$

$$\phi(y) = \begin{cases} 1 & |\{j : y_j = \text{on}\}| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$



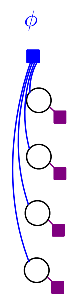
Computing Constraint Factor Messages

$y(j)$: Assignment to variables where $Y_j = \text{on}$

$y(0)$: Special case for all off

$$y(0) = \{Y_1 = \text{off},\ Y_2 = \text{off},\ Y_3 = \text{off},\ Y_4 = \text{off}\}$$

$$\phi(y) = \begin{cases} 1 & |\{j : y_j = \text{on}\}| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

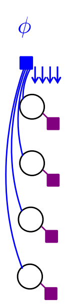


Computing Constraint Factor Messages

- Input: $m_{Y_j \to \phi}(y_j)\ \forall j$
- Goal: $m_{\phi \to Y_j}(y_j)\ \forall j$

$$m_{\phi \to Y_j}(y_j) \propto \frac{\sum_{y:\ Y_j = y_j} \phi(y) s(y)}{m_{Y_j \to \phi}(y_j)}$$

Only need to consider
$y(j')$ for $0 \leq j' \leq n$

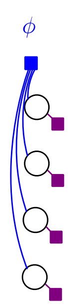


Computing Constraint Factor Messages

$$s(y) = \prod_j m_{Y_i \to \phi}(y_j)$$

$$s(y(0)) = m_{Y_1 \to \phi}(\text{off}) \cdot m_{Y_2 \to \phi}(\text{off}) \cdot m_{Y_3 \to \phi}(\text{off}) \cdot m_{Y_4 \to \phi}(\text{off})$$

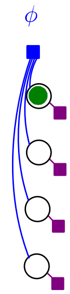
$$s(y(0)) = \prod_{1 \leq j \leq n} m_{Y_j \to \phi}(\text{off})$$



Computing Constraint Factor Messages

$$s(y) = \prod_j m_{Y_i \to \phi}(y_j)$$

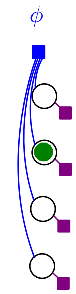
$$s(y(1)) = m_{Y_1 \to \phi}(\text{on}) \cdot m_{Y_2 \to \phi}(\text{off}) \cdot m_{Y_3 \to \phi}(\text{off}) \cdot m_{Y_4 \to \phi}(\text{off})$$



Computing Constraint Factor Messages

$$s(y) = \prod_j m_{Y_i \to \phi}(y_j)$$

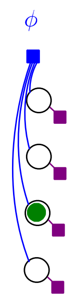
$$s(y(2)) = m_{Y_1 \to \phi}(\text{off}) \cdot m_{Y_2 \to \phi}(\text{on}) \cdot m_{Y_3 \to \phi}(\text{off}) \cdot m_{Y_4 \to \phi}(\text{off})$$



Computing Constraint Factor Messages

$$s(y) = \prod_j m_{Y_i \to \phi}(y_j)$$

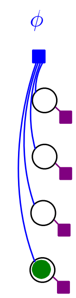
$$s(y(3)) = m_{Y_1 \to \phi}(\text{off}) \cdot m_{Y_2 \to \phi}(\text{off}) \cdot m_{Y_3 \to \phi}(\text{on}) \cdot m_{Y_4 \to \phi}(\text{off})$$



Computing Constraint Factor Messages

$$s(y) = \prod_j m_{Y_i \to \phi}(y_j)$$

$$s(y(4)) = m_{Y_1 \to \phi}(\text{off}) \cdot m_{Y_2 \to \phi}(\text{off}) \cdot m_{Y_3 \to \phi}(\text{off}) \cdot m_{Y_4 \to \phi}(\text{on})$$



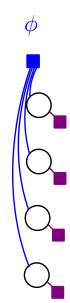
Computing Constraint Factor Messages

$$s(y) = \prod_j m_{Y_i \to \phi}(y_j)$$

$$s(y(0)) = \prod_{1 \leq j \leq n} m_{Y_j \to \phi}(\text{off})$$

$\forall j > 0:$

$$s(y(j)) = s(y(0)) \frac{m_{Y_j \to \phi}(\text{on})}{m_{Y_j \to \phi}(\text{off})}$$

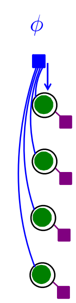


Computing Constraint Factor Messages

$$m_{\phi \to Y_1}(\text{on}) \propto \frac{s(y(1))}{m_{Y_1 \to \phi}(\text{on})}$$

$$m_{\phi \to Y_1}(\text{off}) \propto \frac{s(*) - s(y(1))}{m_{Y_1 \to \phi}(\text{off})}$$

$$s(*) = \sum_{0 \leq j \leq n} s(y(j))$$



Computing Constraint Factor Messages

1. Precompute: $s(y(0)) = \prod_{1 \le j \le n} m_{Y_j \to \phi}(\text{off})$
$O(n)$

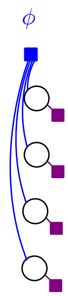
2. $\forall j > 0: \; s(y(j)) = s(y(0)) \frac{m_{Y_j \to \phi}(\text{on})}{m_{Y_j \to \phi}(\text{off})}$
$O(n)$

3. Partition: $s(*) = \sum_{0 \le j \le n} s(y(j))$
$O(n)$

4. Messages:
$O(n)$

$$m_{\phi \to Y_j}(\text{on}) \propto \frac{s(y(j))}{m_{Y_j \to \phi}(\text{on})}$$

$$m_{\phi \to Y_j}(\text{off}) \propto \frac{s(*) - s(y(j))}{m_{Y_j \to \phi}(\text{off})}$$



Using BP Marginals

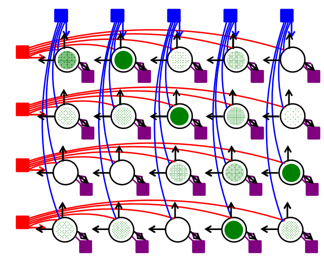
$$P(y_{ij}|x) \approx b_{Y_{ij}}(y_{ij})$$

Expected Feature Counts:

$$\mathbb{E}f(i,j) \approx b_{Y_{ij}}(\text{on}) f(i,j)$$

Marginal Decoding:

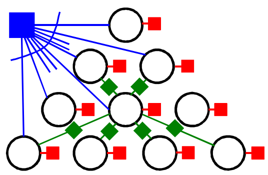
$$\hat{y}_{ij} = \begin{cases} \text{on} & b_{Y_{ij}}(\text{on}) \ge \tau \\ \text{off} & \text{otherwise} \end{cases}$$



Dependency Parsing with BP

$y_{ij} \in \{\text{left, right, off}\}$

$$\phi(y) = \begin{cases} 1 & y \text{ forms a tree} \\ 0 & \text{otherwise} \end{cases}$$

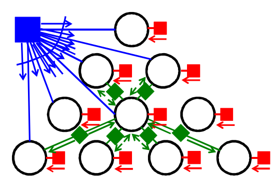


$$\phi(y_{ij}) = \begin{cases} \exp(w^\top f(i,j)) & y_{ij} = \text{left} \\ \exp(w^\top f(j,i)) & y_{ij} = \text{right} \\ 1 & y_{ij} = \text{off} \end{cases}$$

$\phi(y_{ij}, y_{jk})$

(Smith & Eisner, 2008)
(Martins et al., 2010)

Dependency Parsing with BP



$\phi_{ij}(y_{ij})$	Arity 1	✓
$\phi_{ijk}(y_{ij}, y_{jk})$	Arity 2	✓
$\phi_{\text{TREE}}(y)$	Arity $O(n^2)$	✗

Exponential in arity of factor

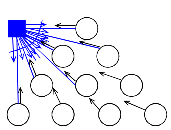
Messages from the Tree Factor

- Input: $m_{Y_{ij} \to \phi_{\text{TREE}}}(y_{ij})$ for all variables
- Goal: $m_{\phi_{\text{TREE}} \to Y_{ij}}(y_{ij})$ for all variables

$$m_{\phi_{\text{TREE}} \to Y_{ij}}(y_{ij}) \propto \sum \phi_{\text{TREE}}(y) s(y) \frac{1}{m_{Y_{ij} \to \phi_{\text{TREE}}}(y_{ij})}$$

$$\phi_{\text{TREE}}(y) = \begin{cases} 1 & y \text{ forms a tree} \\ 0 & \text{otherwise} \end{cases}$$

$$T = \{y \; : \; y \text{ forms a tree}\}$$



What Do Parsers Do?

- Initial state:
  - Value of an edge (i has parent j): $v(i,j)$
  - Value of a tree: $v(t) = \prod_{(i,j) \in t} v(i,j)$
- Run inside-outside to compute:
  - Total score for all trees: $Z = \sum_t v(t)$
  - Total score for an edge: $Z(i,j) = \sum_{t:\,(i,j) \in t} v(t)$

Initializing the Parser

(Klein & Manning, 2002)

Problem:

$$v(t) = \prod_{(i,j)\in t} v(i,j) \qquad s(y) = \prod_{ij} m_{Y_{ij}\to\phi_{\text{TREE}}}(y_{ij})$$

Product over edges in t: $y_{ij} = \text{left}$ or $y_{ji} = \text{right}$

Product over ALL edges, including $y_{ij} = \text{off}$

Solution: Use odds ratios

$$\pi = \prod_{ij} m_{Y_{ij}\to\phi_{\text{TREE}}}(\text{off})$$

$$v(y_{ij}) = \begin{cases} \frac{m_{Y_{ij}\to\phi_{\text{TREE}}}(y_{ij})}{m_{Y_{ij}\to\phi_{\text{TREE}}}(\text{off})} & y_{ij} \neq \text{off} \\ 1 & y_{ij} = \text{off} \end{cases}$$

$$\pi v(t) = s(y)$$

Running the Parser

$$Z = \sum_t v(t) \qquad \pi v(t) = s(y) \qquad \pi Z = \sum_{y\in T} s(y)$$

Sums we want:

$$\pi Z(i,j) = \sum_{\substack{y\in T \\ y_{ij}=\text{left}}} s(y) \qquad \pi Z(j,i) = \sum_{\substack{y\in T \\ y_{ij}=\text{right}}} s(y)$$

$$\pi(Z - Z(i,j) - Z(j,i)) = \sum_{\substack{y\in T \\ y_{ij}=\text{off}}} s(y)$$

Computing Tree Factor Messages

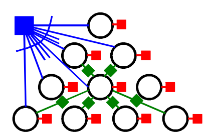
1. Precompute: $\pi = \prod_{ij} m_{Y_{ij}\to\phi_{\text{TREE}}}(\text{off})$
2. Initialize: $v(i,j) = \begin{cases} \frac{m_{Y_{ij}\to\phi_{\text{TREE}}}(\text{left})}{m_{Y_{ij}\to\phi_{\text{TREE}}}(\text{off})} & i < j \\ \frac{m_{Y_{ji}\to\phi_{\text{TREE}}}(\text{right})}{m_{Y_{ji}\to\phi_{\text{TREE}}}(\text{off})} & j < i \end{cases}$
3. Run inside-outside
4. Messages: $m_{\phi_{\text{TREE}}\to Y_{ij}}(y_{ij}) \propto \begin{cases} \frac{\pi Z(i,j)}{m_{Y_{ij}\to\phi_{\text{TREE}}}(y_{ij})} & y_{ij} = \text{left} \\ \frac{\pi Z(j,i)}{m_{Y_{ij}\to\phi_{\text{TREE}}}(y_{ij})} & y_{ij} = \text{right} \\ \frac{\pi(Z - Z(i,j) - Z(j,i))}{m_{Y_{ij}\to\phi_{\text{TREE}}}(y_{ij})} & y_{ij} = \text{off} \end{cases}$

Using BP Marginals

$$P(y_{ij}|x) \approx b_{Y_{ij}}(y_{ij})$$

- Expected Feature Counts:

$$\mathbb{E}f(i,j) \approx \begin{cases} b_{Y_{ij}}(\text{left}) f(i,j) & i < j \\ b_{Y_{ji}}(\text{right}) f(i,j) & j < i \end{cases}$$



- Minimum Risk Decoding:

1. Initialize:

$$v(i,j) = \begin{cases} \frac{b_{Y_{ij}}(\text{left})}{b_{Y_{ij}}(\text{off})} & i < j \\ \frac{b_{Y_{ji}}(\text{right})}{b_{Y_{ji}}(\text{off})} & j < i \end{cases}$$

2. Run parser:

$$\hat{t} = \operatorname*{argmax}_t s(t)$$

Structured BP Summary

- Tricky part is factors whose arity grows with input size
- Simplify the problem by focusing on sums of total scores
- Exploit problem-specific structure to compute sums efficiently
- Use odds ratios to eliminate "default" values that don't appear in dynamic program sums

Belief Propagation Tips

- Don't compute unary messages multiple times
- Store variable beliefs to save time computing variable to factor messages (divide one out)
- Update the slowest messages less frequently
- You don't usually need to run to convergence; measure the speed/performance tradeoff

Part 6: Wrap-Up

Berkeley
N L P

Mean Field vs Belief Propagation

- When to use Mean Field:
  - Models made up of weakly interacting structures that are individually tractable
  - Joint models often have this flavor
- When to use Belief Propagation:
  - Models with intersecting factors that are tractable in isolation but interact badly
  - You often get models like this when adding non-local features to an existing tractable model

Mean Field vs Belief Propagation

- Mean Field Advantages
  - For models where it applies, the coordinate ascent procedure converges quite quickly
- Belief Propagation Advantages
  - More broadly applicable
  - More freedom to focus on factor graph design when modeling
- Advantages of Both
  - Work pretty well when the real posterior is peaked (like in NLP models!)

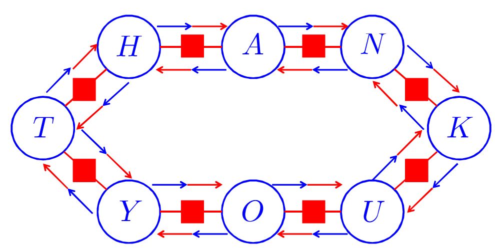
Other Variational Techniques

- Variational Bayes
  - Mean Field for models with parametric forms (e.g. Liang et al., 2007; Cohen et al., 2010)
- Expectation Propagation
  - Theoretical generalization of BP
  - Works kind of like Mean Field in practice; good for product models (e.g. Hall and Klein, 2012)
- Convex Relaxation
  - Optimize a convex approximate objective

Related Techniques

- Dual Decomposition
  - Not probabilistic, but good for finding maxes in similar models (e.g. Koo et al., 2010; DeNero & Machery, 2011)
- Search approximations
  - E.g. pruning, beam search, reranking
  - Orthogonal to approximate inference techniques (and often stackable!)

Thank You



Appendix A: Bibliography

Berkeley NLP

References

- Conditional Random Fields
  - John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In ICML.
- Edge-Factored Dependency Parsing
  - Ryan McDonald, Koby Crammer, and Fernando Pereira (2005). Online Large-Margin Training of Dependency Parsers. In ACL.
  - Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič (2005). Non-projective Dependency Parsing using Spanning Tree Algorithms. In HLT/EMNLP.

References

- Factorial Chain CRF
  - Charles Sutton, Khashayar Rohanimanesh, and Andrew McCallum (2004). Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data. In ICML.
- Second-Order Dependency Parsing
  - Ryan McDonald and Fernando Pereira (2006). Online Learning of Approximate Dependency Parsing Algorithms. In EACL.
  - Xavier Carreras (2007). Experiments with a Higher-Order Projective Dependency Parser. In CoNLL Shared Task Session.

References

- Max Matching Word Alignment
  - Ben Taskar, Simon, Lacoste-Julien, and Dan Klein (2005). A discriminative matching approach to word alignment. In HLT/EMNLP.
- Iterated Conditional Modes
  - Julian Besag (1986). On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society, Series B.* Vol. 48, No. 3, pp. 259-302.
- Structured Mean Field
  - Eric P. Xing, Michael I. Jordan, and Stuart Russell (2003). A Generalized Mean Field Algorithm for Variational Inference in Exponential Families. In UAI.

References

- Joint Parsing and Alignment
  - David Burkett, John Blitzer, and Dan Klein (2010). Joint Parsing and Alignment with Weakly Synchronized Grammars. In NAACL.
- Word Alignment with Belief Propagation
  - Jan Niehues and Stephan Vogel (2008). Discriminative Word Alignment via Alignment Matrix Modelling. In ACL:HLT.
  - Fabien Cromières and Sadao Kurohashi (2009). An Alignment Algorithm using Belief Propagation and a Structure-Based Distortion Model. In EACL.
  - David Burkett and Dan Klein (2012). Fast Inference in Phrase Extraction Models with Belief Propagation. In NAACL.

References

- Dependency Parsing with Belief Propagation
  - David A. Smith and Jason Eisner (2008). Dependency Parsing by Belief Propagation. In EMNLP.
  - André F. T. Martins, Noah A. Smith, Eric P. Xing, Pedro M. Q. Aguiar, and Mário A. T. Figueiredo (2010). Turbo Parsers: Dependency Parsing by Approximate Variational Inference. In EMNLP.
- Odds Ratios
  - Dan Klein and Chris Manning (2002). A Generative Constituent-Context Model for Improved Grammar Induction. In ACL.
- Variational Bayes
  - Percy Liang, Slav Petrov, Michael I. Jordan, and Dan Klein (2007). The Infinite PCFG using Hierarchical Dirichlet Processes. In EMNLP/CoNLL.
  - Shay B. Cohen, David M. Blei, and Noah A. Smith (2010). Variational Inference for Adaptor Grammars. In NAACL.

References

- Expectation Propagation
  - David Hall and Dan Klein (2012). Training Factored PCFGs with Expectation Propagation. In EMNLP-CoNLL.
- Dual Decomposition
  - Terry Koo, Alexander M. Rush, Michael Collins, Tommi Jaakkola, and David Sontag (2010). Dual Decomposition for Parsing with Non-Projective Head Automata. In EMNLP.
  - Alexander M. Rush, David Sontag, Michael Collins, and Tommi Jaakkola (2010). On Dual Decomposition and Linear Programming Relaxations for Natural Language Processing. In EMNLP.
  - John DeNero and Klaus Macherey (2011). Model-Based Aligner Combination Using Dual Decomposition. In ACL.

Further Reading

- Theoretical Background
  - Martin J. Wainwright and Michael I. Jordan (2008). Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, Vol. 1, No. 1-2, pp. 1-305.
- Gentle Introductions
  - Christopher M. Bishop (2006). *Pattern Recognition and Machine Learning*. Springer.
  - David J.C. MacKay (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

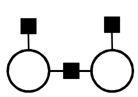
Further Reading

- More Variational Inference for Structured NLP
  - Zhifei Li, Jason Eisner, and Sanjeev Khudanpur (2009). Variational Decoding for Statistical Machine Translation. In ACL.
  - Michael Auli and Adam Lopez (2011). A Comparison of Loopy Belief Propagation and Dual Decomposition for Integrated CCG Supertagging and Parsing. In ACL.
  - Veselin Stoyanov and Jason Eisner (2012). Minimum-Risk Training of Approximate CRF-Based NLP Systems. In NAACL.
  - Jason Naradowsky, Sebastian Riedel, and David A. Smith (2012). Improving NLP through Marginalization of Hidden Syntactic Structure. In EMNLP-CoNLL.
  - Greg Durrett, David Hall, and Dan Klein (2013). Decentralized Entity-Level Modeling for Coreference Resolution. In ACL.

Appendix B: Mean Field Update Derivation

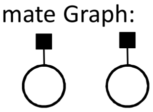
Mean Field Update Derivation

Model:



$$p(y) \propto \phi(y_1)\phi(y_2)\phi(y_1, y_2)$$

Approximate Graph:



$$q(y) = q(y_1)q(y_2)$$

Goal: $q(y_1) = \underset{q(y_1)}{\operatorname{argmin}} KL(q||p)$

Mean Field Update Derivation

$$KL(q||p) = \sum_{y_1,y_2} q(y_1)q(y_2) \frac{\log q(y_1)q(y_2)}{\log p(y_1,y_2)}$$

Mean Field Update Derivation

$$
\begin{aligned}
KL(q||p) &= \sum_{y_1,y_2} q(y_1)q(y_2)\frac{\log q(y_1)q(y_2)}{\log p(y_1,y_2)} \\
&= \sum_{y_1,y_2} q(y_1)q(y_2)
\end{aligned}
$$

Mean Field Update Derivation

$$
\begin{aligned}
KL(q||p) &= \sum_{y_1,y_2} q(y_1)q(y_2)\frac{\boxed{\log q(y_1)q(y_2)}}{\log p(y_1,y_2)} \\
&= \sum_{y_1,y_2} q(y_1)q(y_2)\left(\log q(y_1) + \log q(y_2)\right)
\end{aligned}
$$

Mean Field Update Derivation

$$
\begin{aligned}
KL(q||p) &= \sum_{y_1,y_2} q(y_1)q(y_2)\frac{\log q(y_1)q(y_2)}{\boxed{\log p(y_1,y_2)}} \\
&= \sum_{y_1,y_2} q(y_1)q(y_2)\left(\log q(y_1) + \log q(y_2) - \log\phi(y_1) - \log\phi(y_2) - \log\phi(y_1,y_2) + \log Z_x\right)
\end{aligned}
$$

Mean Field Update Derivation

$$
\begin{aligned}
KL(q||p) &= \sum_{y_1,y_2} q(y_1)q(y_2)\frac{\log q(y_1)q(y_2)}{\log p(y_1,y_2)} \\
&= \sum_{y_1,y_2} q(y_1)q(y_2)\left(\log q(y_1) + \log q(y_2) - \log\phi(y_1) - \log\phi(y_2) - \log\phi(y_1,y_2) + \log Z_x\right) \\
&= \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log q(y_1)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1,y_2)\right) + \\
&\quad \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log q(y_2)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_2)\right) + \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log Z_x\right)
\end{aligned}
$$

Mean Field Update Derivation

$$
\begin{aligned}
KL(q||p) &= \sum_{y_1,y_2} q(y_1)q(y_2)\frac{\log q(y_1)q(y_2)}{\log p(y_1,y_2)} \\
&= \sum_{y_1,y_2} q(y_1)q(y_2)\left(\log q(y_1) + \log q(y_2) - \log\phi(y_1) - \log\phi(y_2) - \log\phi(y_1,y_2) + \log Z_x\right) \\
&= \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log q(y_1)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1,y_2)\right) + \\
&\quad \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log q(y_2)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_2)\right) + \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log Z_x\right) \\
&= \left(\sum_{y_1} q(y_1)\log q(y_1)\right) - \left(\sum_{y_1} q(y_1)\log\phi(y_1)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1,y_2)\right) + \\
&\quad \left(\sum_{y_2} q(y_2)\log q(y_2)\right) - \left(\sum_{y_2} q(y_2)\log\phi(y_2)\right) + \log Z_x
\end{aligned}
$$

Mean Field Update Derivation

$$
\begin{aligned}
KL(q||p) &= \sum_{y_1,y_2} q(y_1)q(y_2)\frac{\log q(y_1)q(y_2)}{\log p(y_1,y_2)} \\
&= \sum_{y_1,y_2} q(y_1)q(y_2)\left(\log q(y_1) + \log q(y_2) - \log\phi(y_1) - \log\phi(y_2) - \log\phi(y_1,y_2) + \log Z_x\right) \\
&= \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log q(y_1)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1,y_2)\right) + \\
&\quad \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log q(y_2)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_2)\right) + \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log Z_x\right) \\
&= \left(\sum_{y_1} q(y_1)\log q(y_1)\right) - \left(\sum_{y_1} q(y_1)\log\phi(y_1)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1,y_2)\right) + \\
&\quad \left(\sum_{y_2} q(y_2)\log q(y_2)\right) - \left(\sum_{y_2} q(y_2)\log\phi(y_2)\right) + \log Z_x
\end{aligned}
$$

$$
\frac{\partial KL(q||p)}{\partial q(y_1)} =
$$

Mean Field Update Derivation

$$
\begin{aligned}
KL(q||p) &= \sum_{y_1,y_2} q(y_1)q(y_2)\frac{\log q(y_1)q(y_2)}{\log p(y_1,y_2)} \\
&= \sum_{y_1,y_2} q(y_1)q(y_2)\left(\log q(y_1) + \log q(y_2) - \log\phi(y_1) - \log\phi(y_2) - \log\phi(y_1,y_2) + \log Z_x\right) \\
&= \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log q(y_1)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1,y_2)\right) + \\
&\quad \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log q(y_2)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_2)\right) + \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log Z_x\right) \\
&= \boxed{\left(\sum_{y_1} q(y_1)\log q(y_1)\right)} - \left(\sum_{y_1} q(y_1)\log\phi(y_1)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1,y_2)\right) + \\
&\quad \left(\sum_{y_2} q(y_2)\log q(y_2)\right) - \left(\sum_{y_2} q(y_2)\log\phi(y_2)\right) + \log Z_x
\end{aligned}
$$

$$\frac{\partial KL(q||p)}{\partial q(y_1)} = (\log q(y_1) + 1)$$

Mean Field Update Derivation

$$
\begin{aligned}
KL(q||p) &= \sum_{y_1,y_2} q(y_1)q(y_2)\frac{\log q(y_1)q(y_2)}{\log p(y_1,y_2)} \\
&= \sum_{y_1,y_2} q(y_1)q(y_2)\left(\log q(y_1) + \log q(y_2) - \log\phi(y_1) - \log\phi(y_2) - \log\phi(y_1,y_2) + \log Z_x\right) \\
&= \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log q(y_1)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1,y_2)\right) + \\
&\quad \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log q(y_2)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_2)\right) + \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log Z_x\right) \\
&= \left(\sum_{y_1} q(y_1)\log q(y_1)\right) - \boxed{\left(\sum_{y_1} q(y_1)\log\phi(y_1)\right)} - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1,y_2)\right) + \\
&\quad \left(\sum_{y_2} q(y_2)\log q(y_2)\right) - \left(\sum_{y_2} q(y_2)\log\phi(y_2)\right) + \log Z_x
\end{aligned}
$$

$$\frac{\partial KL(q||p)}{\partial q(y_1)} = (\log q(y_1) + 1) - \log\phi(y_1)$$

Mean Field Update Derivation

$$
\begin{aligned}
KL(q||p) &= \sum_{y_1,y_2} q(y_1)q(y_2)\frac{\log q(y_1)q(y_2)}{\log p(y_1,y_2)} \\
&= \sum_{y_1,y_2} q(y_1)q(y_2)\left(\log q(y_1) + \log q(y_2) - \log\phi(y_1) - \log\phi(y_2) - \log\phi(y_1,y_2) + \log Z_x\right) \\
&= \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log q(y_1)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1,y_2)\right) + \\
&\quad \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log q(y_2)\right) - \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_2)\right) + \left(\sum_{y_1,y_2} q(y_1)q(y_2)\log Z_x\right) \\
&= \left(\sum_{y_1} q(y_1)\log q(y_1)\right) - \left(\sum_{y_1} q(y_1)\log\phi(y_1)\right) - \boxed{\left(\sum_{y_1,y_2} q(y_1)q(y_2)\log\phi(y_1,y_2)\right)} + \\
&\quad \cancel{\left(\sum_{y_2} q(y_2)\log q(y_2)\right) - \left(\sum_{y_2} q(y_2)\log\phi(y_2)\right) + \log Z_x}
\end{aligned}
$$

$$\frac{\partial KL(q||p)}{\partial q(y_1)} = (\log q(y_1) + 1) - \log\phi(y_1) - \sum_{y_2} q(y_2)\log\phi(y1, y2)$$

Mean Field Update Derivation

$$0 = (\log q(y_1) + 1) - \log\phi(y_1) - \sum_{y_2} q(y_2)\log\phi(y_1, y_2)$$

Mean Field Update Derivation

$$0 = (\log q(y_1) + 1) - \log\phi(y_1) - \sum_{y_2} q(y_2)\log\phi(y_1, y_2)$$

$$\log q(y_1) = \log\phi(y_1) + \sum_{y_2} q(y_2)\log\phi(y_1, y_2) - 1$$

Mean Field Update Derivation

$$0 = (\log q(y_1) + 1) - \log\phi(y_1) - \sum_{y_2} q(y_2)\log\phi(y_1, y_2)$$

$$\log q(y_1) = \log\phi(y_1) + \sum_{y_2} q(y_2)\log\phi(y_1, y_2) - 1$$

$$q(y_1) = \exp\left(\log\phi(y_1) + \sum_{y_2} q(y_2)\log\phi(y_1, y_2) - 1\right)$$

Mean Field Update Derivation

$$0 = (\log q(y_1) + 1) - \log \phi(y_1) - \sum_{y_2} q(y_2) \log \phi(y_1, y_2)$$

$$\log q(y_1) = \log \phi(y_1) + \sum_{y_2} q(y_2) \log \phi(y_1, y_2) - 1$$

$$q(y_1) \propto \exp\left(\log \phi(y_1) + \sum_{y_2} q(y_2) \log \phi(y_1, y_2) - 1\right)$$

Mean Field Update Derivation

$$0 = (\log q(y_1) + 1) - \log \phi(y_1) - \sum_{y_2} q(y_2) \log \phi(y_1, y_2)$$

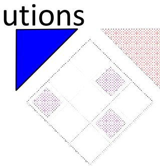
$$\log q(y_1) = \log \phi(y_1) + \sum_{y_2} q(y_2) \log \phi(y_1, y_2) - 1$$

$$q(y_1) \propto \exp\left(\log \phi(y_1) + \sum_{y_2} q(y_2) \log \phi(y_1, y_2)\right)$$

$$q(y_i) \propto \exp\left(\sum_{c:i \in c} \mathbb{E}_{q_{-Y_i}} \log \phi_c(y_c)\right)$$

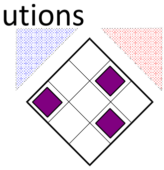
Appendix C: Joint Parsing and Alignment Component Distributions

Joint Parsing and Alignment Component Distributions



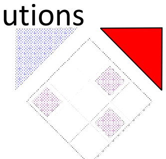
$$q(t) \propto \exp\left(\sum_{n_{i}X_{j} \in t} w^\top f_t(n) + \sum_{n_{i}X_{j} \in t} \sum_{{}_{s}X'_{t}} q(b_{ij,st}) q(n'_{{}_{s}X'_{t}}) w^\top f_{tat'}(n, b, n')\right)$$

Joint Parsing and Alignment Component Distributions



$$q(a) \propto \exp\left(\sum_{b_{ij,st} \in a} w^\top f_a(b) + \sum_{b_{ij,st} \in a} \sum_{X, X'} q(n_{i}X_{j}) q(n'_{{}_{s}X'_{t}}) w^\top f_{tat'}(n, b, n')\right)$$

Joint Parsing and Alignment Component Distributions

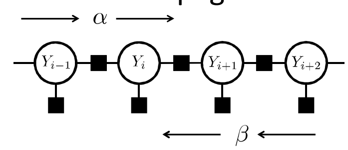


$$q(t') \propto \exp\left(\sum_{n'_{{}_{s}X'_{t}} \in t'} w^\top f_{t'}(n') + \sum_{n'_{{}_{s}X'_{t}} \in t'} \sum_{{}_{i}X_{j}} q(n_{i}X_{j}) q(b_{ij,st}) w^\top f_{tat'}(n, b, n')\right)$$

Appendix D: Forward-Backward as Belief Propagation

Berkeley
N L P

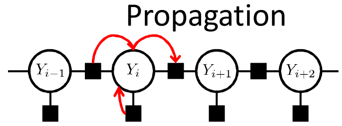
Forward-Backward as Belief Propagation



$$\alpha_i(y_i) = \phi_i(y_i) \sum_{y_{i-1}} \alpha_{i-1}(y_{i-1})\phi_{i-1,i}(y_{i-1}, y_i)$$

$$\beta_i(y_i) = \sum_{y_{i+1}} \beta_{i+1}(y_{i+1})\phi_{i,i+1}(y_i, y_{i+1})\phi_{i+1}(y_{i+1})$$

Forward-Backward as Belief Propagation

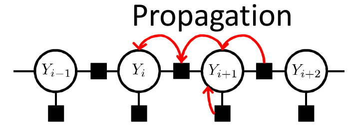


$$\alpha_i(y_i) = \phi_i(y_i) \sum_{y_{i-1}} \alpha_{i-1}(y_{i-1})\phi_{i-1,i}(y_{i-1}, y_i)$$
$$= m_{\phi_i \to Y_i}(y_i)\ m_{\phi_{i-1,i} \to Y_i}(y_i)$$

$$m_{\phi_i \to Y_i}(y_i) = \phi_i(y_i) \qquad m_{Y_i \to \phi_{i,i+1}}(y_i) = \alpha_i(y_i)$$

$$m_{\phi_{i-1,i} \to Y_i}(y_i) = \sum_{y_{i-1}} \alpha_{i-1}(y_{i-1})\phi_{i-1,i}(y_{i-1}, y_i)$$

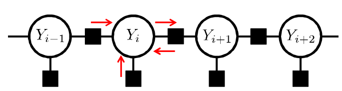
Forward-Backward as Belief Propagation



$$\beta_i(y_i) = \sum_{y_{i+1}} \beta_{i+1}(y_{i+1})\phi_{i,i+1}(y_i, y_{i+1})\phi_{i+1}(y_{i+1})$$
$$= m_{\phi_{i,i+1} \to Y_i}(y_i)$$
$$= \sum_{y_{i+1}} m_{Y_{i+1} \to \phi_{i,i+1}}(y_{i+1})\phi_{i,i+i}(y_i, y_{i+1})$$

$$m_{Y_{i+1} \to \phi_{i,i+1}}(y_{i+1}) = m_{\phi_{i+1} \to Y_{i+1}}(y_{i+1}) *$$
$$m_{\phi_{i+1,i+2} \to Y_{i+1}}(y_{i+1})$$

Forward-Backward Marginal Beliefs



$$P(y_i|x) \propto \alpha_i(y_i)\beta_i(y_i)$$

$$= m_{Y_i \to \phi_{i,i+1}}(y_i)\ m_{\phi_{i,i+1} \to Y_i}(y_i)$$

$$= m_{\phi_{i-1,i} \to Y_i}(y_i)\ m_{\phi_i \to Y_i}(y_i)\ m_{\phi_{i,i+1} \to Y_i}(y_i)$$