PCFGs: The Inside-Outside Algorithm

CMSC 473/673
UMBC
November 15th, 2017
Course Announcement 1: Assignment 3

Due next Wednesday (~7 days)

Any questions?
Course Announcement 2: Graduate Paper 2

Due Monday 11/27 (~12 days)

Any questions?
Course Announcement 3: Assignment 4

Out Friday

Due Monday, 12/11
Recap from last time...
CKY Recognizer

Input: * string of N words
* grammar in CNF
Output: True (with parse)/False

Data structure: N*N table $T$
- Rows indicate span start (0 to N-1)
- Columns indicate span end (1 to N)

$T[i][j]$ lists constituents spanning $i \rightarrow j$
“Papa ate the caviar with a spoon”
CKY Viterbi Parser

Input: * string of N words
   * probabilistic grammar in CNF
Output: Parse with probability (or None)

Data structure: K*N*N table \( T \)
   K non-terminal symbols in the grammar
   Rows indicate span start (0 to N-1)
   Columns indicate span end (1 to N)

\( T[X][i][j] \) lists *most likely* constituents beginning with rule X spanning i \( \rightarrow \) j
“Papa ate the caviar with a spoon”
CKY Comparison

**Recognizer**

\[ T = \text{bool}[K][N][N+1] \]

\[ T[*][*][*] = \text{False} \]

for(j = 1; j ≤ N; ++j) {
    \[ T[X][j-1][j] = \{ \text{True for non-terminal X in G if X \rightarrow word}_j \} \]
}

for(width = 2; width ≤ N; ++width) {
    for(start = 0; start < N - width; ++start) {
        end = start + width
        for(mid = start+1; mid < end; ++mid) {
            \[ T[X][start][end] = \{ \text{True \& T[Y][start][mid] \& T[Z][mid][end] \} \]
            for rule X \rightarrow Y Z : G
        }
    }
}

**Viterbi**

\[ T = \text{WeightedCell}[K][N][N+1] \]

\[ T[*][*][*] = 0 \]

for(j = 1; j ≤ N; ++j) {
    \[ T[X][j-1][j] = \text{argmax } \{ p(X \rightarrow \text{word}_j) \text{ for non-terminal X in G if X \rightarrow word}_j \} \]
}

for(width = 2; width ≤ N; ++width) {
    for(start = 0; start < N - width; ++start) {
        end = start + width
        for(mid = start+1; mid < end; ++mid) {
            \[ T[X][start][end] = \text{argmax } \{ p(X \rightarrow Y Z) * T[Y][start][mid] * T[Z][mid][end] \} \]
            for rule X \rightarrow Y Z : G
        }
    }
}
\[ T = \text{SemiRingCell}^{[K][N][N+1]} \]

\[ T[*][*][*] = 0 \]

for \( j = 1; j \leq N; ++j \) {
    \[ T[X][j-1][j] \oplus = \{ p(X \rightarrow \text{word}_j) \text{ for non-terminal } X \text{ in } G \text{ if } X \rightarrow \text{word}_j \} \]
}

for \( \text{width} = 2; \text{width} \leq N; ++\text{width} \) {
    for \( \text{start} = 0; \text{start} < N - \text{width}; ++\text{start} \) {
        \[ \text{end} = \text{start} + \text{width} \]
        for \( \text{mid} = \text{start} + 1; \text{mid} < \text{end}; ++\text{mid} \) {
            \[ T[X][\text{start}][\text{end}] \oplus = \] \[ \big[ p(X \rightarrow Y Z) \otimes T[Y][\text{start}][\text{mid}] \otimes T[Z][\text{mid}][\text{end}] \big] \]
            for rule \( X \rightarrow Y Z : G \)
        }
    }
}

Adapted from Jason Eisner
# CKY Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Weights</th>
<th>$\oplus$</th>
<th>$\otimes$</th>
<th>$\odot$</th>
<th>$\mathbb{1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recognizer</td>
<td>Boolean</td>
<td>or</td>
<td>and</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>(True/False)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Viterbi</td>
<td>[0,1]</td>
<td>max</td>
<td>*</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Adapted from Jason Eisner
Probabilistic Context Free Grammar (PCFG) Tasks

Find the most likely parse (for an observed sequence)

Calculate the (log) likelihood of an observed sequence $w_1, \ldots, w_N$

Learn the grammar parameters
PCFG Likelihood

\[ p(w_1 \ w_2 \ w_3 \ \ldots \ w_N) \]

likelihood of word sequence \( w_1w_2\ldots w_N \)
PCFG Likelihood

\[ p(w_1 \ w_2 \ w_3 \ ... \ w_N) \]

*likelihood of word sequence \( w_1w_2...w_N \)

\[ p(S \rightarrow^+ w_1 \ w_2 \ w_3 \ ... \ w_N) \]

*likelihood of word sequence \( w_1w_2...w_N \) based on starting at \( S \)
PCFG Likelihood

\[ p(w_1 \ w_2 \ w_3 \ \ldots \ \ w_N) \]

likelihood of word sequence \( w_1w_2\ldots w_N \)

\[ p(S \rightarrow^+ \ w_1 \ w_2 \ w_3 \ \ldots \ \ w_N) \]

likelihood of word sequence \( w_1w_2\ldots w_N \)

based on starting at \( S \)

\[ p(\quad) + p(\quad) + p(\quad) + \ldots \]
PCFG Likelihood

\[ p(S \rightarrow^+ w_1 w_2 w_3 \ldots w_N) \]
PCFG Likelihood

$$p(S \rightarrow^+ w_1 w_2 w_3 \ldots w_N)$$
PCFG Likelihood

\[ p(S \rightarrow^+ w_1 \ w_2 \ w_3 \ ... \ w_N) \]
PCFG Likelihood

\[ p(S \rightarrow^+ w_1 w_2 w_3 \ldots w_N) \]
PCFG Likelihood

$p(S \rightarrow^+ w_1 w_2 w_3 \ldots w_N)$

compute all of the ways a sequence of words could be generated from ("inside") a particular constituent
PCFG Likelihood (Inside Algorithm)

Input: * string of N words  
  * probabilistic grammar in CNF
Output: Likelihood of words being generated by the root

Data structure: K*N*N table $\beta$
  K non-terminal symbols in the grammar
  Rows indicate span start (0 to N-1)
  Columns indicate span end (1 to N)

$\beta[X][i][j]$ lists total probability of X generating words spanning i $\rightarrow$ j
Recall: Backward Algorithm

\[ \beta_{\text{HMM}}(i, s) = \sum_{s'} \beta_{\text{HMM}}(i + 1, s') \times p(s'|s) \times p(\text{obs at } i + 1|s') \]

\( \beta_{\text{HMM}}(i, s) \) is the total probability of all paths:
1. that start at step \( i \) at state \( s \)
2. that terminate at the end
3. (that emit the observation \( \text{obs at } i+1 \))
PCFGs: Inside Algorithm

$$\beta(X, s, t) = \sum_{Y,Z} \sum_{k:s<k<t} \beta(Y, s, k) \times \beta(Z, k, t) \times p(X \rightarrow Y Z)$$

\(\beta(X, s, t)\) is the total probability of all derivations:

1. that start from non-terminal \(X\), with left index \(s\)
2. that terminate after the \(t^{th}\) word
3. that emit the observations from \(s\) (inclusive) to \(t\) (exclusive)
PCFGs: Inside Algorithm

\[ \beta(X, s, t) = \sum_{Y,Z} \sum_{k:s<k<t} \beta(Y, s, k) \times \beta(Z, k, t) \times p(X \rightarrow Y Z) \]

1. that start from non-terminal \( X \), with left index \( s \)
2. that terminate after the \( t \)th word
3. that emit the observations from \( s \) (inclusive) to \( t \) (exclusive)

All valid rules \( X \rightarrow Y Z \)

All possible splits of the \( s \rightarrow t \) span
\[ \beta = \text{WeightedCell}[K][N][N+1] \]

\[
\begin{align*}
\text{for} (j = 1; j \leq N; ++j) \{ \\
\quad & \beta[X][j-1][j] += \left[ p(X \rightarrow \text{word}_j) \text{ for non-terminal } X \text{ in } G \text{ if } X \rightarrow \text{word}_j \right] \\
\}
\end{align*}
\]

\[
\begin{align*}
\text{for} (\text{width} = 2; \text{width} \leq N; ++\text{width}) \{ \\
\quad \text{for} (\text{start} = 0; \text{start} < N - \text{width}; ++\text{start}) \{ \\
\quad\quad & \text{end} = \text{start} + \text{width} \\
\quad\quad \text{for} (\text{mid} = \text{start} + 1; \text{mid} < \text{end}; ++\text{mid}) \{ \\
\quad\quad\quad \text{for} (\text{rule } X \rightarrow Y Z : G) \{ \\
\quad\quad\quad\quad & \beta[X][\text{start}][\text{end}] += \\
\quad\quad\quad\quad\quad & \left[ p(X \rightarrow Y Z) \times \beta[Y][\text{start}][\text{mid}] \times \beta[Z][\text{mid}][\text{end}] \right] \\
\quad\quad\quad\quad \text{for rule } X \rightarrow Y Z : G \\
\quad\quad\quad \}
\quad\quad \}
\quad \}
\end{align*}
\]
# CKY Algorithms

<table>
<thead>
<tr>
<th>Recognizer</th>
<th>Weights</th>
<th>$\oplus$</th>
<th>$\otimes$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean (True/False)</td>
<td>or</td>
<td>and</td>
<td>False</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>Viterbi</td>
<td>[0,1]</td>
<td>max</td>
<td>*</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Likelihood/Inside</td>
<td>[0,1]</td>
<td>+</td>
<td>*</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
"Papa ate the caviar with a spoon"

1.0 $S \rightarrow NP \ VP$

.6 $NP \rightarrow \text{Det} \ N$

.3 $NP \rightarrow NP \ PP$

.6 $VP \rightarrow V \ NP$

.4 $VP \rightarrow VP \ PP$

1.0 $PP \rightarrow P \ NP$

.1 $NP \rightarrow \text{Papa}$

.6 $N \rightarrow \text{caviar}$

.4 $N \rightarrow \text{spoon}$

.1 $V \rightarrow \text{spoon}$

.9 $V \rightarrow \text{ate}$

1.0 $P \rightarrow \text{with}$

.5 $\text{Det} \rightarrow \text{the}$

.5 $\text{Det} \rightarrow \text{a}$

Entire grammar
“Papa ate the caviar with a spoon”
Papa ate the caviar with a spoon
“Papa ate the caviar with a spoon”
"Papa ate the caviar with a spoon"
"Papa ate the caviar with a spoon"
"Papa ate the caviar with a spoon"
Probabilistic Context Free Grammar (PCFG) Tasks

Find the most likely parse (for an observed sequence)

Calculate the (log) likelihood of an observed sequence $w_1, ..., w_N$

Learn the grammar parameters
PCFG Outside Score

\[ p(w_1 \ w_2 \ \ldots \ w_{l-1} \ A \ w_{m+1} \ \ldots \ w_N) \]

likelihood of non-terminal A from l to m, and surrounding words

\[ p(S \rightarrow^+ w_1 \ w_2 \ \ldots \ w_{l-1} \ A \ w_{m+1} \ \ldots \ w_N) \]
PCFG Outside Score

\[ p(S \rightarrow^+ w_1 w_2 \ldots w_{l-1} A w_{m+1} \ldots w_N) \]
PCFG Outside Score

\[ p(S \rightarrow^+ w_1 w_2 \ldots w_{l-1} A w_{m+1} \ldots w_N) \]

Option 1: \( B \rightarrow C A \)

must have already considered this word
PCFG Outside Score

\[ p(S \rightarrow^+ w_1 w_2 ... w_{l-1} A w_{m+1} ... w_N) \]

Option 1: \( B \rightarrow C A \)

Option 2: \( B \rightarrow A C \)

must have already considered these words
PCFGs: Outside Algorithm

\[ \alpha(X, s, t) \]

\[ \alpha(X, s, t) \] is the total probability of all derivations:

1. that include non-terminal X (but not words) from s to t
2. and all (& only) the observed words before s and after t
PCFGs: Outside Algorithm

\[ \alpha(X, s, t) = \sum_{Y,Z} \sum_{k:0 \leq k < s} \beta(Z, k, s) \times \alpha(Y, k, t) \times p(Y \rightarrow Z X) + \]

Option 1: \( B \rightarrow C A \)

Option 2: \( B \rightarrow A C \)

All indices where C can start

Total probability of words \([0,k)\) and \([t, N)\)

Y spans \([k, t)\)

\( \alpha(X, s, t) \) is the total probability of all derivations:

1. that include non-terminal X (but not words) from s to t
2. and all (& only) the observed words before s and after t
PCFGs: Outside Algorithm

\[ \alpha(X, s, t) = \]

Option 1: \( B \rightarrow CA \)

\[ \sum_{Y,Z} \sum_{k:0 \leq k < s} \beta(Z, k, s) \cdot \alpha(Y, k, t) \cdot p(Y \rightarrow Z X) + \]

Option 2: \( B \rightarrow AC \)

\[ \sum_{Y,Z} \sum_{k:t \leq k \leq N} \beta(Z, t, k) \cdot \alpha(Y, s, k) \cdot p(Y \rightarrow X Z) \]

\[ \alpha(X, s, t) \] is the total probability of all derivations:

1. that include non-terminal \( X \) (but not words) from \( s \) to \( t \)
2. and all (& only) the observed words before \( s \) and after \( t \)
\[ \alpha(A, s, t) = \]
\[ \sum_{B,C} \sum_{k:0 \leq k < s} \beta(C, k, s) \alpha(B, k, t) \ast p(B \rightarrow CA) + \]
\[ \sum_{B,C} \sum_{k:t \leq k \leq N} \beta(C, t, k) \alpha(B, s, k) \ast p(B \rightarrow AC) \]

\[ \alpha(A, 3, 6) = \cdots + \]
\[ \beta(C, 1, 3) \ast \alpha(B, 1, 6) \ast p(B \rightarrow CA) + \]
\[ \cdots \]

Option 1: B \rightarrow CA

must have already considered this word

must have already considered this word
\[ \alpha(A, s, t) = \sum_B \sum_C \sum_{k:0 \leq k < s} \beta(C, k, s) \cdot \alpha(B, k, t) \cdot p(B \rightarrow C \, A) + \]

\[ \sum_B \sum_C \sum_{k:t \leq k \leq N} \beta(C, t, k) \cdot \alpha(B, s, k) \cdot p(B \rightarrow A \, C) \]

\[ \alpha(A, 3, 6) = \cdots + \beta(C, 1, 3) \cdot \alpha(B, 1, 6) \cdot p(B \rightarrow C \, A) + \]

\[ \cdots \]

\[ \alpha(B, 1, 6) \text{ will have already considered these words} \]

Option 1: \( B \rightarrow C \, A \)

must have already considered this word

must have already considered this word

PCFG Outside Score
\( \alpha = \text{WeightedCell}[K][N][N+1] \)

\[
\text{for}(\text{width} = N; \text{width} \geq 1; \text{--width}) \{ \\
\quad \text{for}(\text{start} = 0; \text{start} \leq N - \text{width}; ++\text{start}) \{ \\
\quad \quad \text{end} = \text{start} + \text{width} \\
\quad \quad \text{for}(\text{mid} = \text{start}+1; \text{mid} < \text{end}; ++\text{mid}) \{ \\
\quad \quad \quad \alpha[Y][\text{start}][\text{mid}] += \\
\quad \quad \quad \quad \quad \quad \quad \quad \quad [p(X \rightarrow Y Z) \ast \alpha[Y][\text{start}][\text{end}] \ast \beta[Z][\text{mid}][\text{end}] \\
\quad \quad \quad \quad \quad \quad \quad \quad \quad \text{for rule } X \rightarrow Y Z : G] \\
\quad \quad \alpha[Z][\text{mid}][\text{end}] += \\
\quad \quad \quad \quad \quad \quad \quad \quad \quad [p(X \rightarrow Y Z) \ast \alpha[Z][\text{start}][\text{end}] \ast \beta[Y][\text{mid}][\text{end}] \\
\quad \quad \quad \quad \quad \quad \quad \quad \quad \text{for rule } X \rightarrow Y Z : G] \\
\quad \quad \} \\
\quad \} \\
\} \\
\]
Inside-Outside Algorithm

1. Run inside algorithm (bottom-up, small-to-large)
2. Run outside algorithm (top-down, large-to-small)
Inside-Outside Algorithm

1. Run inside algorithm (bottom-up, small-to-large)

2. Run outside algorithm (top-down, large-to-small)

Q: Why?
Inside-Outside Algorithm

1. Run inside algorithm (bottom-up, small-to-large)

2. Run outside algorithm (top-down, large-to-small)

$\alpha(X, s, t)$ is the total probability of all derivations:
1. that include non-terminal $X$ (but not words) from $s$ to $t$
2. and all (& only) the observed words before $s$ and after $t$

$\beta(X, s, t)$ is the total probability of all derivations:
1. that start from non-terminal $X$, with left index $s$
2. that terminate after the $t^{th}$ word
3. that emit the observations from $s$ (inclusive) to $t$ (exclusive)

Q: Why?
Inside-Outside Algorithm

1. Run inside algorithm (bottom-up, small-to-large)

2. Run outside algorithm (top-down, large-to-small)

Q: Why?
Inside-Outside Algorithm

1. Run inside algorithm (bottom-up, small-to-large)

2. Run outside algorithm (top-down, large-to-small)

Q: Why?

A: Compute Posterior Probabilities (Expectations)
Getting Expected Counts

Papa ate the caviar with a spoon
Papa ate the caviar with a spoon.

Getting Expected Counts

Papa ate the caviar with a spoon.
Getting Expected Counts

Papa ate the caviar with a spoon

\[ \alpha[VP, 1, 4] \]

\[ \beta[VP, 1, 4] \]

Papillate the caviar with a spoon
Getting Expected Counts

\[ \alpha[VP, 1, 4] = p(\text{Papa VP with a spoon}) \]

\[ \beta[VP, 1, 4] = p(\text{ate the caviar} \mid VP) \]
Getting Expected Counts

\[ \alpha_{[VP, 1, 4]} = p(\text{Papa VP with a spoon}) \]

\[ \beta_{[VP, 1, 4]} = p(\text{ate the caviar | VP}) \]
Getting Expected Counts

\[ \alpha_{[VP, 1, 4]} = p(\text{Papa VP with a spoon}) \]

\[ \beta_{[VP, 1, 4]} = p(\text{ate the caviar} \mid VP) \]

\[ \frac{\alpha_{[VP, 1, 4]} \ast \beta_{[VP, 1, 4]}}{\beta_{[S, 0, 7]}} \]
Getting Expected Counts

\[ \alpha[VP, 1, 4] = p(\text{Papa VP with a spoon}) \]

\[ \beta[VP, 1, 4] = p(\text{ate the caviar} \mid VP) \]

\[ \frac{\alpha[VP, 1, 4] \times \beta[VP, 1, 4]}{\beta[S, 0, 7]} = p(\text{VP}_{1 \rightarrow 4} \mid \text{Papa ... spoon}) \]
Papa ate the caviar with a spoon.
Getting Expected Counts

\[ \alpha[VP, 1, 4] = p(\text{Papa VP with a spoon}) \]

\[ \beta[V, 1, 2] = p(\text{ate} | V) \]

\[ \beta[NP, 2, 4] = p(\text{the caviar} | NP) \]

\[ p(\text{VP} \rightarrow V \text{ NP}) \]
Getting Expected Counts

\[ \alpha[VP, 1, 4] = p(\text{Papa VP with a spoon}) \]

\[ \beta[V, 1, 2] = p(\text{ate | V}) \]

\[ \beta[NP, 2, 4] = p(\text{the caviar | NP}) \]

\[ \frac{\alpha[VP, 1, 4] \ast \beta[V, 1, 2] \ast \beta[NP, 2, 4] \ast P(VP \rightarrow V NP)}{\beta[S, 0, 7]} \]
Getting Expected Counts

\[ \alpha[VP, 1, 4] = p(\text{Papa VP with a spoon}) \]

\[ \alpha[VP, 1, 4] \times \beta[V, 1, 2] \times \beta[NP, 2, 4] \times P(VP \rightarrow V NP) \]

\[ \beta[S, 0, 7] \]

\[ = p(VP_{1 \rightarrow 4} \rightarrow V_1 \ NP_{2 \rightarrow 4}|\text{Papa ... spoon}) \]
Expected Counts

\[ \mathbb{E}[X \rightarrow a \mid w_1 w_2 \cdots w_N] = \]
\[ \frac{p(X \rightarrow a)}{L(w_1 w_2 \cdots w_N)} \sum_{0 \leq i < N: w_i = a} \alpha(X, i, i + 1) \]
Expected Counts

\[
\mathbb{E}[X \rightarrow a \mid w_1w_2\cdots w_N] = \frac{p(X \rightarrow a)}{L(w_1w_2\cdots w_N)} \sum_{0 \leq i < N: w_i = a} \alpha(X, i, i + 1)
\]

\[
\mathbb{E}[X \rightarrow YZ \mid w_1w_2\cdots w_N] = \frac{p(X \rightarrow YZ)}{L(w_1w_2\cdots w_N)} \sum_{0 \leq i < k < j \leq N} \alpha(X, i, j)\beta(Y, i, k)\beta(Z, k, j)
\]
Expectation Maximization (EM)

0. Assume *some* value for your parameters

Two step, iterative algorithm

1. E-step: count under uncertainty, assuming these parameters

   \[ p(z_i) \quad \Rightarrow \quad \text{count}(z_i, w_i) \]

2. M-step: maximize log-likelihood, assuming these uncertain counts

   \[ p^{(t)}(z) \quad \leftrightarrow \quad p^{(t+1)}(z) \]

\[ p(X \rightarrow Y Z) \]
Expectation Maximization (EM)

0. Assume *some* value for your parameters $p(X \rightarrow Y Z)$

Two step, iterative algorithm

1. **E-step**: count under uncertainty, assuming these parameters

   $$\mathbb{E}[X \rightarrow a \mid w_1 w_2 \cdots w_N] = \frac{\frac{p(X \rightarrow a)}{L(w_1 w_2 \cdots w_N)}}{\sum_{0 \leq i < N; \exists i, i+1} \alpha(X, i, i+1)}$$

   $p(z_i)$ $\quad \rightarrow \quad$ count$(z_i, w_i)$

2. **M-step**: maximize log-likelihood, assuming these uncertain counts

   $$\mathbb{E}[X \rightarrow Y Z \mid w_1 w_2 \cdots w_N] = \frac{p(X \rightarrow Y Z)}{L(w_1 w_2 \cdots w_N)} \sum_{0 \leq i < k < j \leq N} \alpha(X, i, j)\beta(Y, i, k)\beta(Z, k, j)$$

   $p^{(t)}(z)$ $\quad \leftrightarrow \quad$ estimated counts $\quad \rightarrow \quad p^{(t+1)}(z)$
Expectation Maximization (EM)

0. Assume *some* value for your parameters \( p(X \rightarrow Y Z) \)

Two step, iterative algorithm

1. E-step: count under uncertainty, assuming these parameters

\[
\mathbb{E}[X \rightarrow a \mid w_1w_2 \ldots w_N] = \frac{p(X \rightarrow a)}{L(w_1w_2 \ldots w_N) \sum_{0 \leq i < N: w_i = a} \alpha(x, i, i + 1)}
\]

\[
p(z_i) \quad \text{count}(z_i, w_i)
\]

2. M-step: maximize log-likelihood, assuming these uncertain counts

\[
\mathbb{E}[X \rightarrow Y Z \mid w_1w_2 \ldots w_N] = \frac{p(X \rightarrow Y Z)}{L(w_1w_2 \ldots w_N) \sum_{0 \leq i < k < j \leq N} \alpha(x, i, j) \beta(y, i, k) \beta(z, k, j)}
\]

\[
p^{(t)}(z) \quad \text{estimated counts} \quad p^{(t+1)}(z)
\]

“Inside-outside”
Estimating PCFGs

Attempt 1:
• Get access to a treebank & count productions
• ~75 F1

Attempt 2:
• Get access to a treebank & run inside-outside
• ~75-91 F1
EM for PCFG Re-estimation

Viterbi or Regular

Start with a good enough grammar

Parse a corpus of unparsed sentences

Reestimate:

Collect counts: $c(X \rightarrow Y Z), c(X)$

Smooth and normalize
Uses for Inside-Outside

As the E step in the EM training algorithm

Predicting which nonterminals are probably where
   Posterior decoding (Goodman, 1998)
   Using uncertain syntax as features in classifiers
   Prune extra backpointers from parse chart

Pruning heuristics

Help learn grammars with more hidden structure

Adapted from Jason Eisner (Lect 26)
Clinton and Congress agreed on a plan.

He said Clinton would try the same tactic again.

Talk to me after class

Ferraro et al. (2012)
Ferraro (2017)