人工知能特論II　第14回

二宮　崇

今日の講義の予定

- PCFGの学習
  - 教師付学習
  - 教師無し学習
    - EMアルゴリズム+内側外側アルゴリズム
- 教科書
  - 北研二(著) 辻井潤一(編) 言語と計算4 確率的言語モデル 東大出版会
  - C. D. Manning & Hinrich Schütze “FOUNDATIONS OF STATISTICAL NATURAL LANGUAGE PROCESSING” MIT Press, 1999
  - D. Jurafsky, J. H. Martin, A. Kehler, K.V. Linden & N. Ward “**Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**” Prentice Hall Series in Artificial Intelligence, 2000

パラメータ推定

PCFGの学習

PCFGの学習

- 教師付学習
  - 正解構文木の集合（＝ツリーバンクと呼ばれる）が存在する場合の学習
- 教師無学習
  - 正解構文木の集合が与えられない状況で、パラメータの推定を行う学習

PCFGの学習

- 最尤推定によるパラメータ推定
  - 教師付学習
    - 正解構文木の集合(ツリーバンクと呼ばれる)に対する単純な数え上げ
    - 文法規則の確率モデル化、語彙化、補語や下位範疇化フレームの導入、粒度の異なる混合モデルに対する補間係数の学習
  - 教師無学習
    - 正解構文木がない場合や書換規則パラメータ以外のパラメータが存在する場合に使われる
    - EMアルゴリズム+内側外側アルゴリズム

パラメータ推定: 単純な数え上げ

- 正解構文木の集合（ツリーバンク）
  Treebankが与えられた時、

$$\theta_{A\to\alpha} = \frac{\sum_{t\in\textbf{Treebank}} C(A\to\alpha;t)}{\sum_{t\in\textbf{Treebank}}\sum_{\beta} C(A\to\beta;t)}$$

有名なツリーバンク

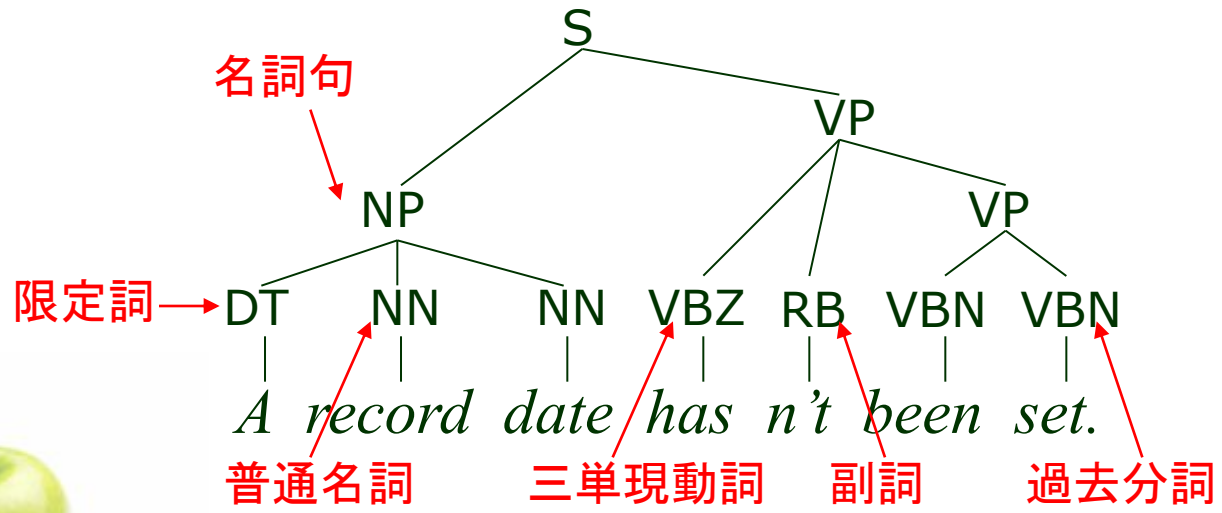
- 構文木や係り受け木を人手で付与したコーパス（ツリーバンク）の登場
  - Penn Treebank [Marcus et al. 1993]
  - SUSANNE [Sampson 1995]
  - TIGER Treebank [Brants et al. 2002]
  - Prague Dependency Treebank [Hajic 1998]
  - Verbmobil [Hinrichs et al. 2000]
  - EDRコーパス [EDR 1995]
  - 京都大学テキストコーパス [黒橋ら 1997]
  - 日本語話し言葉コーパス [前川ら 2000]

Penn Treebank (1/2)

- 構文木が付与された最初の大規模英語ツリーバンク [Marcus et al. 1993]
- 様々な分野の英語テキストを収録
  - Wall Street Journal （新聞） 約5万文、100万語
  - ATIS （航空券予約の会話）
  - Brown （様々な分野のテキスト）
  - Switchboard （電話の自由発話）

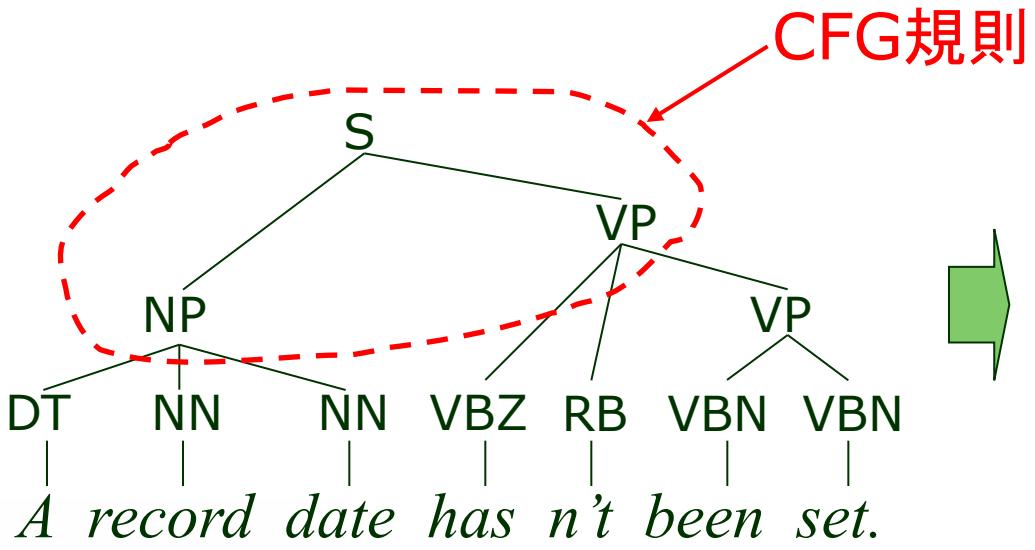
Penn Treebank (2/2)

- 品詞： NN（普通名詞）, VBZ（三単現動詞）…
- 構文木： NP（名詞句）, VP（動詞句）…
- Function tag, null element: 述語項構造を計算するための付加情報 （詳細省略）



PCFGの自動抽出(1/2)

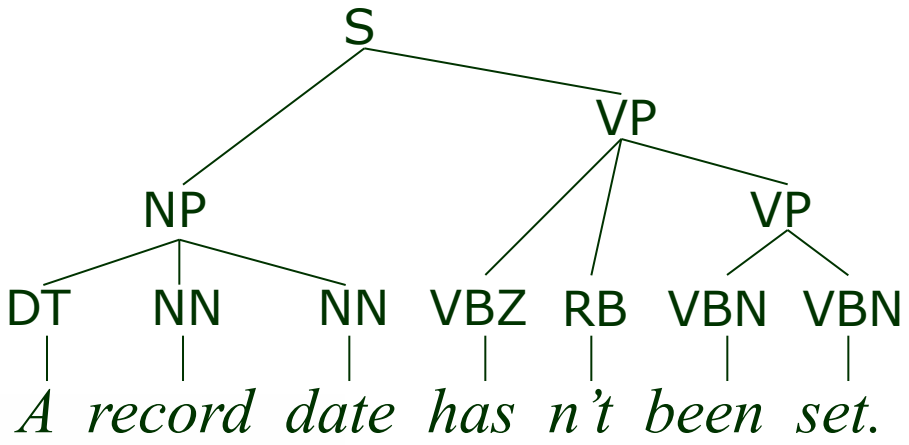
- ツリーバンクの各分岐をCFG規則だと仮定して抽出する [Charniak 1996; 1997] c.f. [Sekine1995]



S → NP VP
NP → DT NN NN
VP → VBZ RB VP
VP → VBN VBN

PCFGの自動抽出(2/2)

- ツリーバンクでの出現頻度から確率値を推定
- 確率値最大の木を探索することで、構文解析の曖昧性解消ができる



S → NP VP 0.5
NP → DT NN NN 0.03
VP → VBZ RB VP 0.02
VP → VBN VBN 0.1

ツリーバンクから学習した文法は「**ツリーバンク文法**」と呼ばれる

ツリーバンク文法の改良

- CFG規則の自動圧縮 [Krotov et al. 1998; 1999]
- CFG規則の確率モデル化 [Magerman 1995; Collins 1997; Charniak 2000]
- 非終端記号の細分化 [Magerman 1995; Collins 1996; 1997; Johnson 1998; Charniak 2000]

マルコフ文法 (1/4)

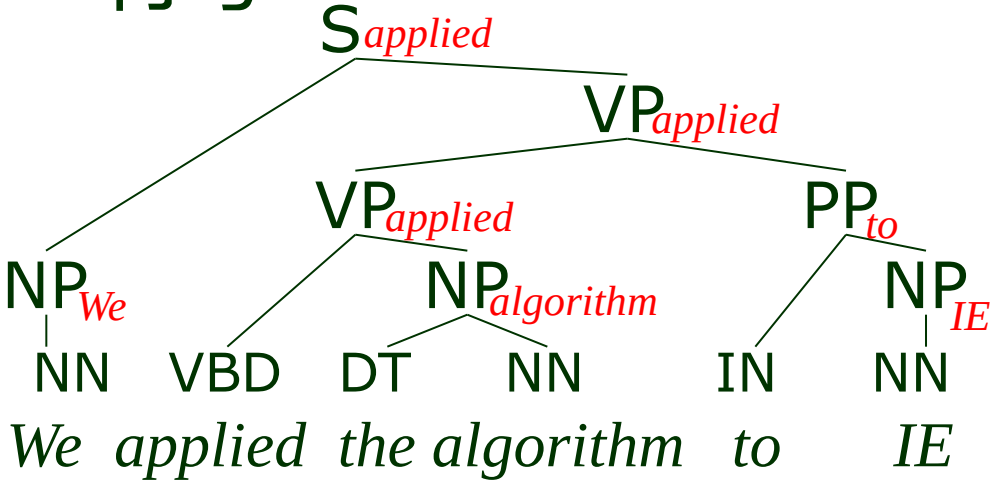
- CFG規則の確率モデル化
  - CFG規則を確率的に生成する [Collins 1997; Charniak 2000]

$$p(\mathrm{NP} \rightarrow \mathrm{DT\ \ NN\ \ NN} \mid \mathrm{NP})$$
$$= p(\mathrm{NN} \mid \mathrm{NP})\, p(\mathrm{NN} \mid \mathrm{NN}, \mathrm{NP})\, p(\mathrm{DT} \mid \mathrm{NN}, \mathrm{NN}, \mathrm{NP})$$

  - 原理的には、全てのCFG規則をもつ PCFG
  - Penn Treebank から抽出したそのままのPCFGよりも高精度

マルコフ文法 (2/4)

- 語彙化: Head percolation table (Magerman 1995) を用いて、非終端記号に head word を付与



Head percolation table

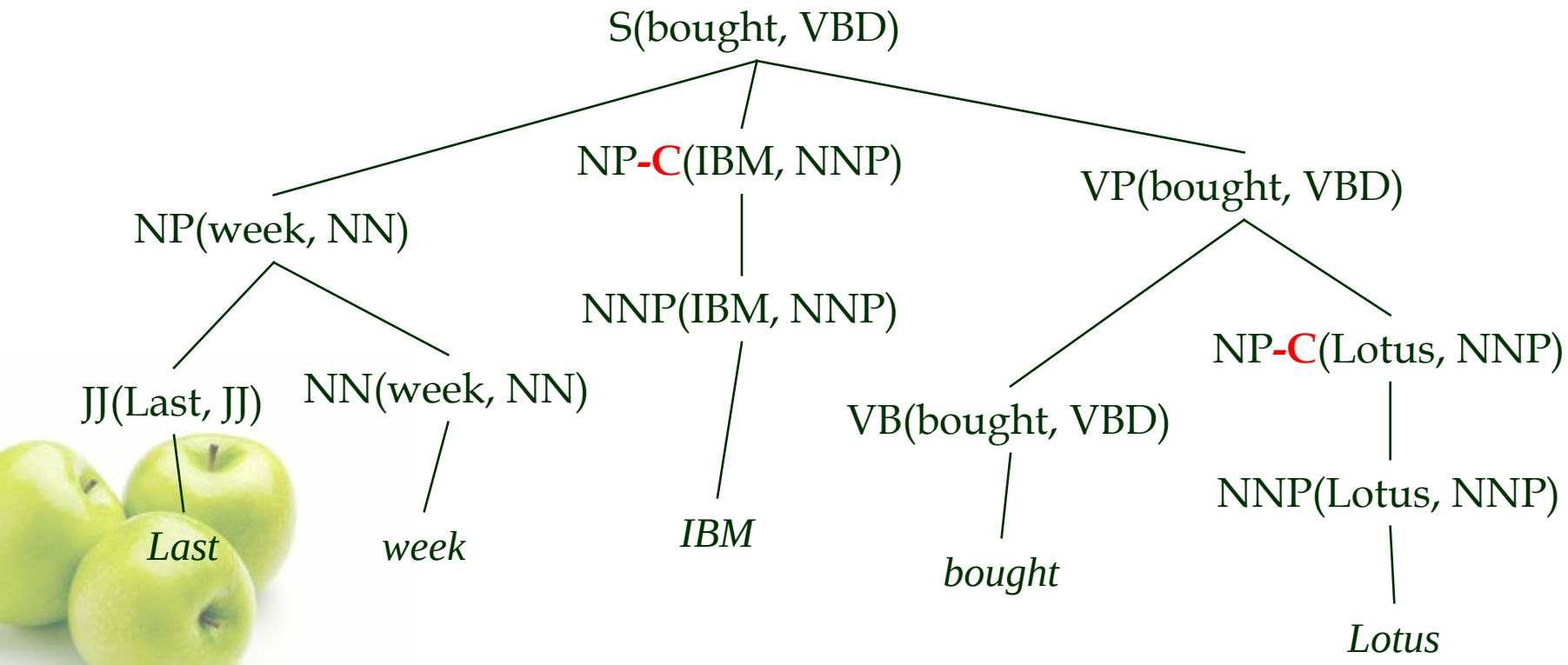
親の記号	主辞になる子の記号
S	VP, …
VP	VP, VBD, VBZ, …
NP	NN, …
PP	IN, …

Charniak [1996]: 80% vs. Magerman [1995]: 86%

- （参考）語彙化の意味 [Gildea 2001; Bikel 2004]

マルコフ文法 (3/4)

- 語彙化+補語(句)/修飾語(句)の区別
- 下位範疇化フレーム(subcat frame)を導入
  - 補語として取る非終端記号の多重集合(multi set)



マルコフ文法 (4/4)

- 現在の（おおよそ）最高性能のパーザー(Charniak&Johnson2005)の基礎
  - Charniak(2000)のパーザーの出力をエントロピー最大化法を用いてreranking
- Charniakパーザーもマルコフ文法の一種

モデル	精度(LF)
collins1999	88.19%
charniak2000	89.55%
charniak&johnson2005	91.02%

EMアルゴリズムと内側外側アルゴリズム

PCFGの教師無し学習

PCFGの教師無し学習

- PCFG教師無し学習におけるパラメータ推定

$$\tilde{\theta} = \arg\max_{\theta} \prod_{s \in \mathbf{Text}} \sum_{t \in T(s)} \prod_{r \in P} \theta_r^{C(r;t)}$$

ただし $\sum_{\alpha} \theta_{A \to \alpha} = 1.0$

$C(r;t)$: 書換規則rが構文木t中に出現する回数
$T(s)$: 文sに対して、フルパージングで得られる全構文木集合

EMアルゴリズムによるPCFGのパラメータ更新式導出

- PCFGのQ関数

$$
\begin{aligned}
Q(\theta,\theta') &= \sum_{i=1}^{N} \sum_{t \in T(s_i)} p(t \mid s_i;\theta) \log p(s_i,t;\theta') \\
&= \sum_{i=1}^{N} \sum_{t \in T(s_i)} \left\{ p(t \mid s_i;\theta) \log \prod_{r \in P} \theta'^{C(r;t)}_r \right\} \\
&= \sum_{i=1}^{N} \sum_{t \in T(s_i)} \left\{ p(t \mid s_i;\theta) \sum_{r \in P} C(r;t) \log \theta'_r \right\}
\end{aligned}
$$

PCFGのパラメータ更新式導出

- PCFGのQ関数

$$Q(\theta,\theta')=\sum_{i=1}^{N}\sum_{t\in T(s_i)}\left\{p(t\mid s_i;\theta)\sum_{r\in P}C(r;t)\log\theta'_r\right\}$$

- ラグランジュ関数

$$L(\theta',\lambda)=Q(\theta,\theta')-\sum_{A\in V_N}\lambda_A(1-\sum_{\beta}\theta_{A\to\beta})$$

PCFGのパラメータ更新式導出

- ラグランジュ関数をパラメータで偏微分

$$\frac{\partial L(\theta',\lambda)}{\partial \theta'_{A\to\beta}} = \sum_{i=1}^{N}\sum_{t\in T(s_i)}\left[\frac{\partial p(t\mid s_i;\theta)}{\partial \theta'_{A\to\beta}}\sum_{r\in P}C(r;t)\log\theta'_r + p(t\mid s_i;\theta)\sum_{r\in P}C(r;t)\frac{\partial \log\theta'_r}{\partial \theta'_{A\to\beta}}\right]+\lambda_A$$

$$= \sum_{i=1}^{N}\sum_{t\in T(s_i)}\left[p(t\mid s_i;\theta)\sum_{r\in P}C(r;t)\frac{\partial \log\theta'_r}{\partial \theta'_{A\to\beta}}\right]+\lambda_A$$

$$= \sum_{i=1}^{N}\sum_{t\in T(s_i)}\left[p(t\mid s_i;\theta)C(A\to\beta;t)\frac{1}{\theta'_{A\to\beta}}\right]+\lambda_A = 0$$

PCFGのパラメータ更新式導出

- パラメータ更新式

$$\theta'_{A\to\beta} = \frac{1}{-\lambda}\sum_{i=1}^{N}\sum_{t\in T(s_i)} p(t\mid s_i;\theta)C(A\to\beta;t)$$

$\sum_{\beta}\theta'_{A\to\beta} = 1$であるから、

$$-\lambda = \sum_{i=1}^{N}\sum_{t\in T(s_i)}\sum_{\beta} p(t\mid s_i;\theta)C(A\to\beta;t)$$

よって

$$\theta'_{A\to\beta} = \frac{\sum_{i=1}^{N}\sum_{t\in T(s_i)} p(t\mid s_i;\theta)C(A\to\beta;t)}{\sum_{i=1}^{N}\sum_{t\in T(s_i)}\sum_{\beta} p(t\mid s_i;\theta)C(A\to\beta;t)}$$

書換規則の適用回数の期待値になっていることに注意

PCFGのEMアルゴリズム

1. $\theta^{(0)}$:= 適当な値
2. **[Eステップ]** $\theta^{(i)}$を用いて各構文木の確率を計算。文sに対する各構文木の相対確率を計算。

$$p(t) = \prod_{r \in P} (\theta_r^{(i)})^{C(r;t)} \qquad p(t \mid s) = \frac{p(t)}{\sum_{u \in T(s)} p(u)}$$

3. **[Mステップ]** $\theta^{(i+1)}$を求める

$$\theta_{A \to \alpha}^{(i+1)} = \frac{\sum_{s \in \mathbf{Text}} \sum_{t \in T(s)} p(t \mid s) C(A \to \alpha; s, t)}{\sum_{s \in \mathbf{Text}} \sum_{t \in T(s)} \sum_{\beta} p(t \mid s) C(A \to \beta; s, t)}$$

4. 2.に戻る

PCFGに対するEMアルゴリズムの問題点

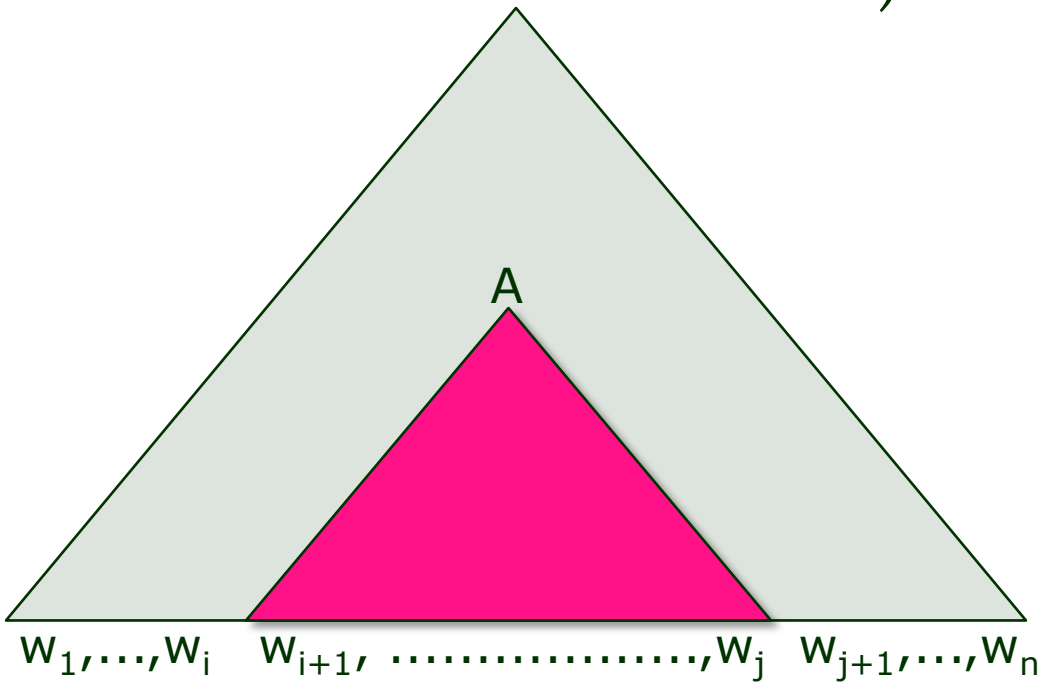
- 構文木が多すぎて現実的な時間で各構文木の相対確率を計算できない！(文長に対して指数爆発的に増加。簡単に数百億ぐらいになる。)

内側外側アルゴリズム

- 畳みこまれた構文木集合に対して効率的にEMアルゴリズムを実行する手法
- アイデア
  - CKY法での構文解析後、CKYテーブルから構文木を展開することなく各書換規則の適用回数の期待値が計算できればよい
  - バイナリールールを仮定し、内側確率と外側確率を動的計画法で効率的に計算
  - 内側確率と外側確率から書換規則の適用回数の期待値が計算

内側確率

- 内側確率 $\beta(i,j,A)$
  - 非終端記号Aから単語列$w_{i+1},\ldots,w_j$を導出する確率(=単語列$w_{i+1},\ldots,w_j$を構文解析してルート(根)がAとなる構文木集合の確率の和)



内側確率の計算

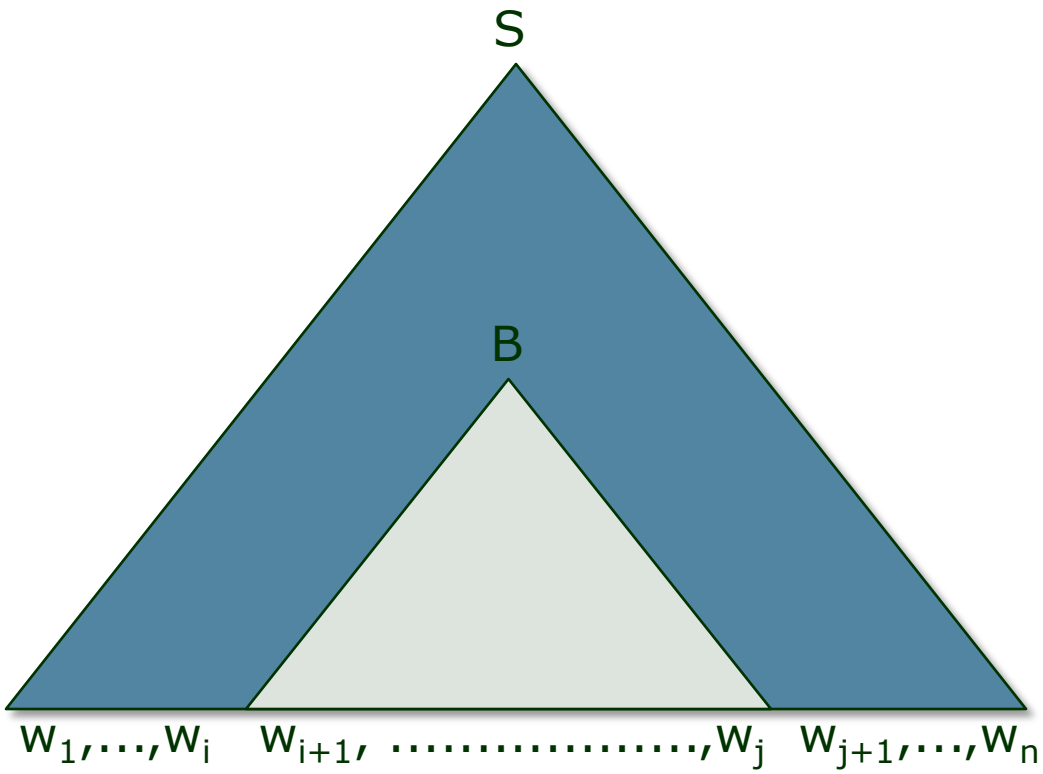
- $S_{i,j}$をビタビアルゴリズムと同様に計算
  - ただし、ファクタリングの際にmaxをとっていたのをsumにする
- $S_{i,j}$: <X, p>の集合 (X: 非終端記号, p: 部分木の確率)
- $S_{i,j}$の求め方

for $k = i+1$ to $j-1$
  forall <B, p_X> $\in S_{i,k}$
    forall <C, p_Y> $\in S_{k,j}$
      forall A $\in$ G(B, C)
        if(<A, p> exists in $S_{i,j}$)
          $p := p + p_X \times p_Y \times \theta_{Z \to X\,Y}$
        else
          $S_{i,j} := S_{i,j} \cup$ <A, $p_X \times p_Y \times \theta_{Z \to X\,Y}$>

sumをmaxにするとビタビアルゴリズムになる

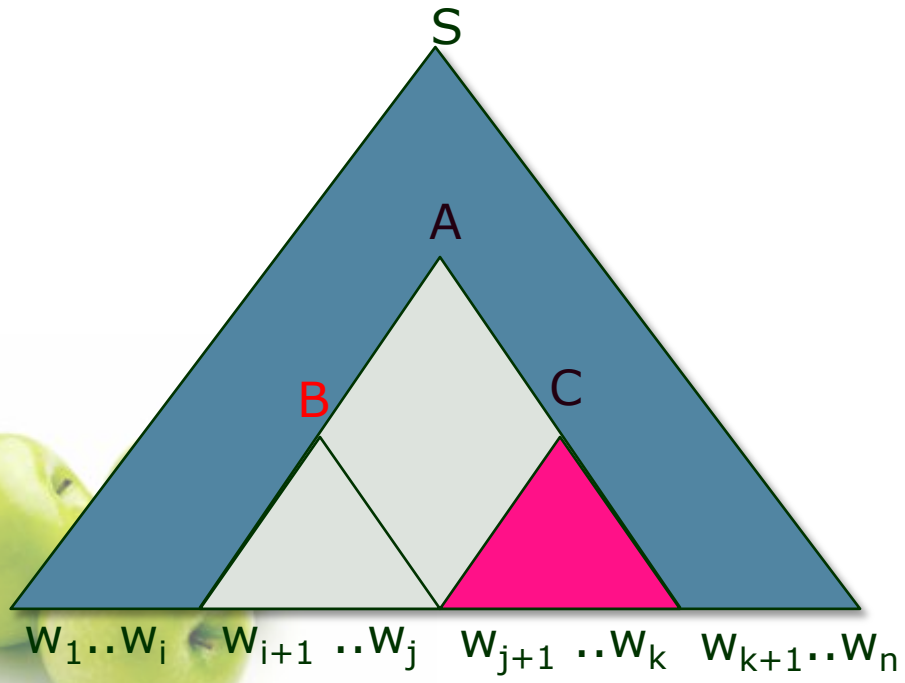
外側確率

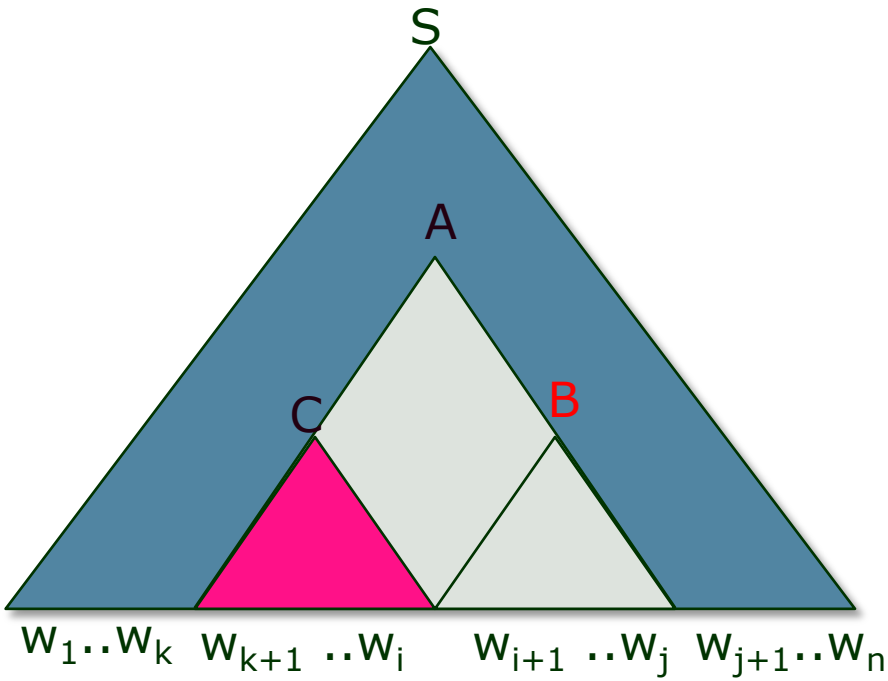
- 外側確率$\alpha(i,j,B)$
  - S(開始記号)から$w_1 \ldots w_i B w_{j+1} \ldots w_n$を導出する確率



外側確率計算のアイデア

- バイナリールールなので、次の2通りとACの組み合わせから$\alpha(i,j,B)$が計算できる
  - A→B C: Aの外側確率×Cの内側確率×$\theta_{A\to B\ C}$
  - A→C B :Aの外側確率×Cの内側確率×$\theta_{A\to C\ B}$





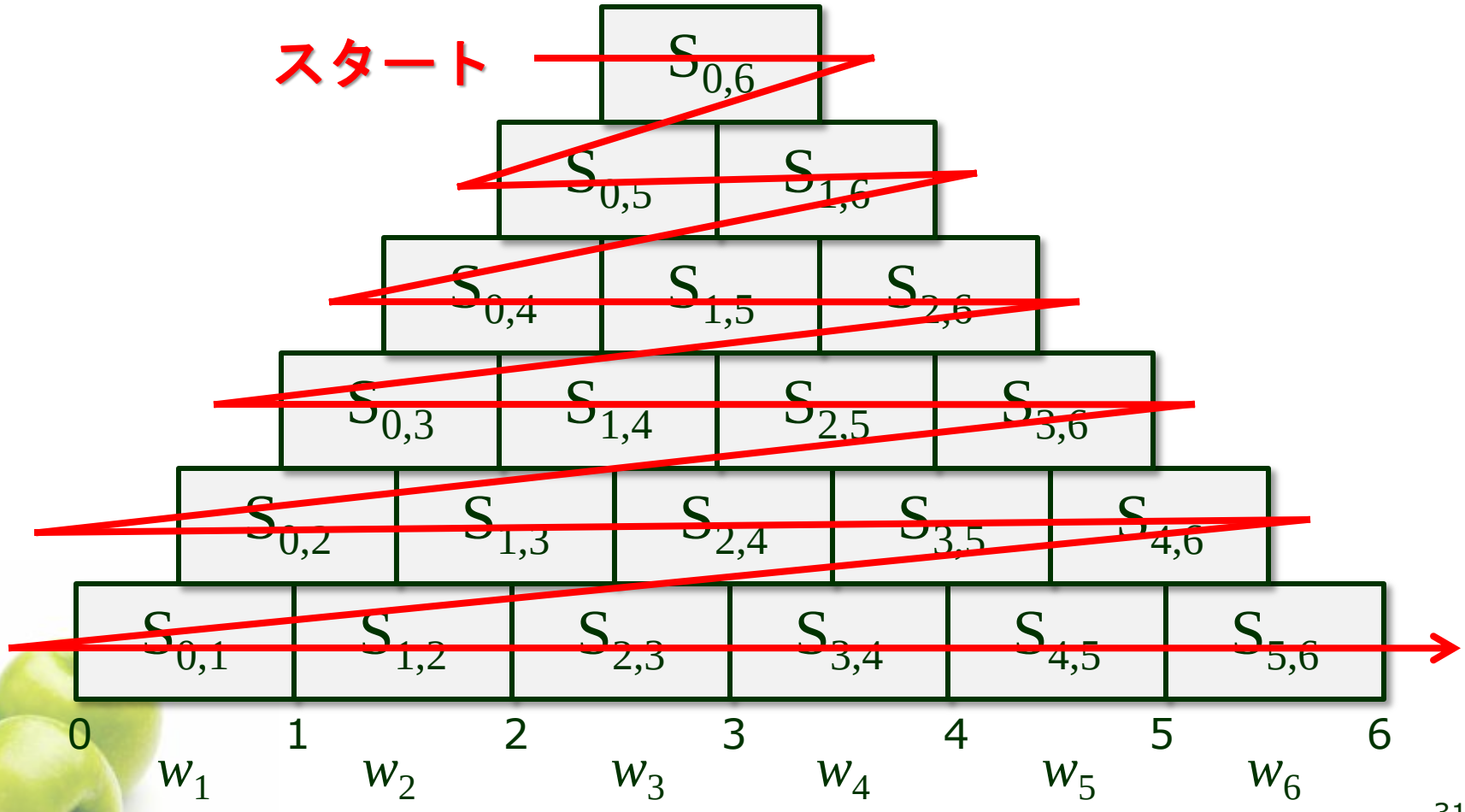
外側確率の計算

- $\alpha(i,j,B)$の計算

$$\alpha(i,j,B) = \sum_{A,C} \theta_{A \to BC} \sum_{k=j+1}^{n} \alpha(i,k,A)\beta(j,k,C) + \sum_{A,C} \theta_{A \to CB} \sum_{k=i-1}^{0} \alpha(k,j,A)\beta(k,i,C)$$

外側確率の計算

- $\alpha(i,j,X)$の計算の順番



外側確率計算アルゴリズム

フルパージングと内側確率計算後を仮定

for all $0 \leqq i < j \leqq n$, $X \in V_N$

$\alpha(i,j,X) := 0$

$\alpha(0,n,S) := 1.0$

for $l = n - 1$ to 1

for $i = 0$ to $n - l$

$j := i + l$

forall $A_{i,j} \rightarrow B_{i,k}\ C_{k,j}$ in $S_{i,j}$

$\alpha(i, k, B) := \alpha(i, k, B) + \alpha(i,j,A) \times \beta(k,j,C) \times \theta_{A \rightarrow B\,C}$

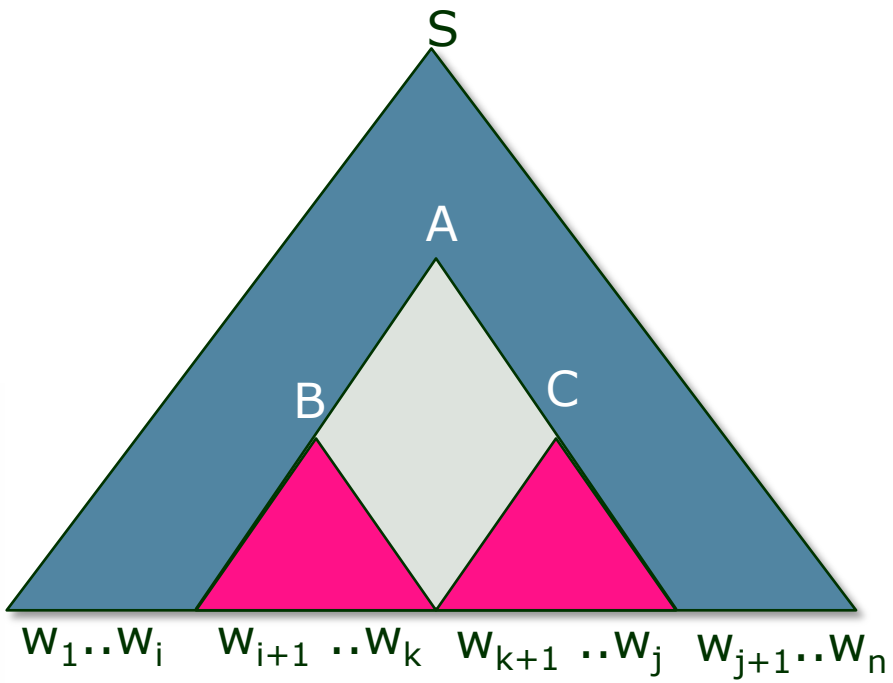
$\alpha(k, j, C) := \alpha(k, j, C) + \alpha(i,j,A) \times \beta(i,k,B) \times \theta_{A \rightarrow B\,C}$

$A_{i,j} \rightarrow B_{i,k}\ C_{k,j}$ in $S_{i,j}$はフルパージングの際に$S_{i,j}$中の非終端記号Aを生成するに到った履歴

- 書換規則が$A \rightarrow B\ C$
- Bは$S_{i,k}$の中の要素
- Cは$S_{k,j}$の中の要素

書換規則の適用回数の期待値

- $A_{i,j} \rightarrow B_{i,k}\ C_{k,j}$の適用回数の期待値
  $=1/Z \times (A_{i,j} \rightarrow B_{i,k}\ C_{k,j}$ の出現した構文木の確率の和)
  $=1/Z \times (B_{i,k}$の内側確率$\times C_{k,j}$の内側確率$\times \theta_{A \rightarrow B\ C} \times A_{i,j}$の外側確率)



内側外側アルゴリズム

1. $\theta^{(0)}$:= 適当な値
2. **[Eステップ]** $\theta^{(i)}$を用いて内側確率と外側確率を計算。
3. **[Mステップ]** 書換規則の適用回数の期待値を計算。$\theta^{(i+1)}$を求める

$$C'(A \to BC; s) = \frac{1}{\beta(0, n, S)} \theta_{A \to BC} \sum_{i=0}^{n-1} \sum_{j=i+2}^{n} \sum_{k=i+1}^{j-1} \alpha(i, j, A)\beta(i, k, B)\beta(k, j, C)$$

$$\theta_{A \to BC}^{(i+1)} = \frac{\sum_{s \in \mathbf{Text}} C'(A \to BC; s)}{\sum_{s \in \mathbf{Text}} \sum_{B,C} C'(A \to BC; s)}$$

4. 2.に戻る

まとめ

- PCFGの学習
  - 教師有り学習
    - 単純な数え上げ
    - ツリーバンクからの文法抽出
    - マルコフ文法
  - 教師無し学習
    - EMアルゴリズムと内側外側アルゴリズム
- 資料

http://aiweb.cs.ehime-u.ac.jp/~ninomiya/ai2/