PCFGのEMアルゴリズムとスムージング

二宮　崇

今日の講義の予定

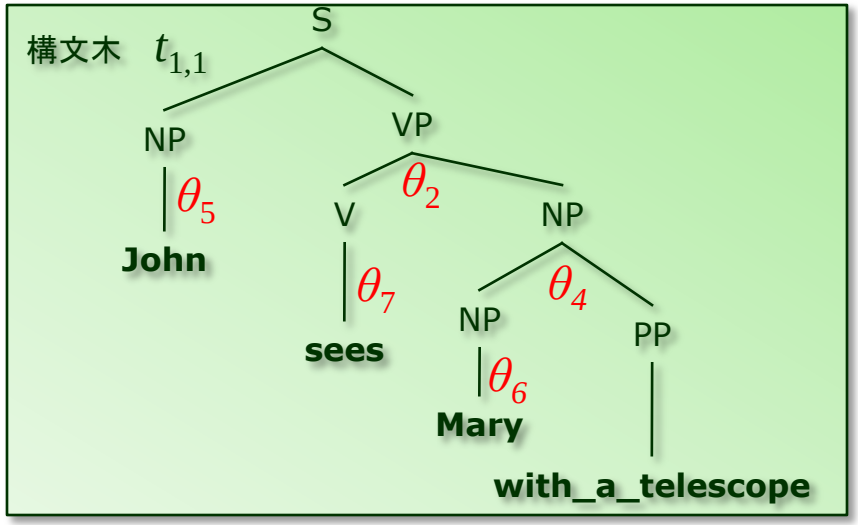
- PCFG (Probabilistic Context Free Grammar, 確率付文脈自由文法)
  - EMアルゴリズム
  - スムージング
- 教科書
  - 北研二(著) 辻井潤一(編) 言語と計算4 確率的言語モデル 東大出版会
  - C. D. Manning & Hinrich Schütze “FOUNDATIONS OF STATISTICAL NATURAL LANGUAGE PROCESSING” MIT Press, 1999
  - D. Jurafsky, J. H. Martin, A. Kehler, K.V. Linden & N. Ward “**Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**” Prentice Hall Series in Artificial Intelligence, 2000

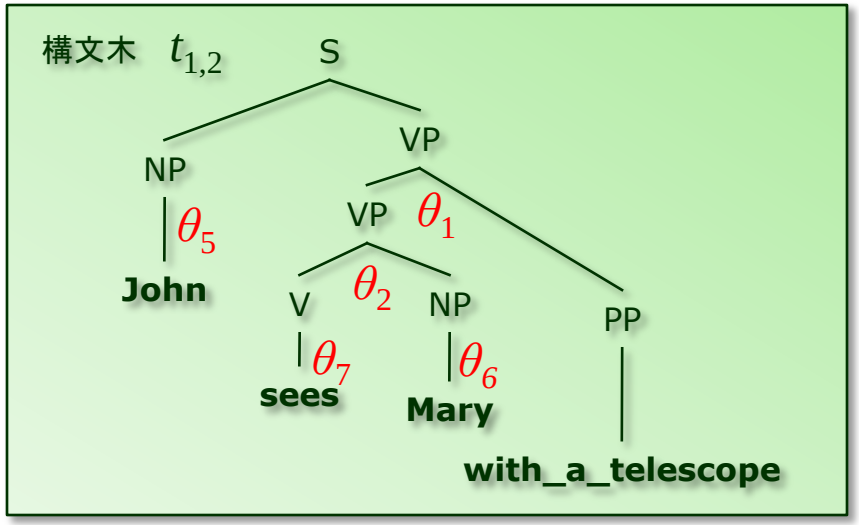
PCFGの最尤推定

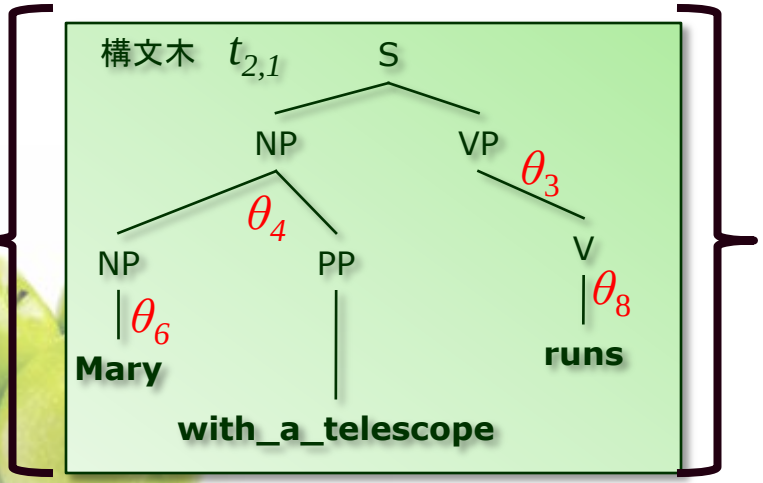
- 次の二文を訓練データとして、パラメータ推定
  - “John sees Mary with_a_telescope”
  - “Mary with_a_telescope runs”

r	P_r
S → NP VP	1.0
VP → VP PP	θ_1
VP → V NP	θ_2
VP → V	θ_3
NP → NP PP	θ_4
NP → John	θ_5
NP → Mary	θ_6
PP → with_a_telescope	1.0
V → sees	θ_7
V → runs	θ_8

PCFGの最尤推定







$t_{s,u}$: sは文ID
　　uはsに対する構文木集合の中での各々の木ID

PCFGの最尤推定

- 問題

文1に対する確率　　文2に対する確率

$$\tilde{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} (\theta_2\theta_4\theta_5\theta_6\theta_7 + \theta_1\theta_2\theta_5\theta_6\theta_7)(\theta_3\theta_4\theta_6\theta_8)$$

$$\theta_1 + \theta_2 + \theta_3 = 1$$

$$\theta_4 + \theta_5 + \theta_6 = 1$$

$$\theta_7 + \theta_8 = 1$$

PCFGの場合この制約を満たすように最大値を求めなければならない
制約付き極値問題⇒ラグランジュの未定乗数法

PCFGの最尤推定

- ラグランジュの未定乗数法

$$\arg\max_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) \text{ただし} g_1(\boldsymbol{\theta}) = 0, \ldots, g_m(\boldsymbol{\theta}) = 0$$

$$\Rightarrow$$

$$L(\boldsymbol{\theta}) = f(\boldsymbol{\theta}) - \lambda_1 g_1(\boldsymbol{\theta}) - \ldots - \lambda_m g_m(\boldsymbol{\theta})$$

$$\frac{\partial L}{\partial \theta_1} = 0, \frac{\partial L}{\partial \theta_2} = 0, \ldots, \frac{\partial L}{\partial \theta_n} = 0$$

$$\begin{aligned} L(\boldsymbol{\theta}) = \quad & (\theta_2\theta_4\theta_5\theta_6\theta_7 + \theta_1\theta_2\theta_5\theta_6\theta_7)(\theta_3\theta_4\theta_6\theta_8) \\ & - \lambda_1(\theta_1 + \theta_2 + \theta_3 - 1) - \lambda_2(\theta_4 + \theta_5 + \theta_6 - 1) \\ & - \lambda_3(\theta_7 + \theta_8 - 1) \end{aligned}$$

PCFGの最尤推定

- 結果
  - θ_1= 0.081357
  - θ_2= 0.459321
  - θ_3= 0.459321
  - θ_4= 0.377964
  - θ_5= 0.207345
  - θ_6= 0.41469
  - θ_7= 0.5
  - θ_8= 0.5

r	P_r
S → NP VP	1.0
VP → VP PP	θ_1
VP → V NP	θ_2
VP → V	θ_3
NP → NP PP	θ_4
NP → John	θ_5
NP → Mary	θ_6
PP → with_a_telescope	1.0
V → sees	θ_7
V → runs	θ_8

EMアルゴリズム

- 最尤推定をコンピュータで行うためによく用いられるアルゴリズム
- アルゴリズム
  1. **θ** := 適当な値
  2. [Eステップ] **θ**を用いて各構文木の確率を計算
  3. [Mステップ]全体の尤度がより高くなる新しい**θ**を求める
  4. 2.に戻る

EMアルゴリズム: Eステップ

- $\theta^{(i)}$: 前回求めたパラメータ
- 各構文木の確率

$$p(t_{1,1};\boldsymbol{\theta}^{(i)}) = \theta_2^{(i)}\theta_4^{(i)}\theta_5^{(i)}\theta_6^{(i)}\theta_7^{(i)}$$

$$p(t_{1,2};\boldsymbol{\theta}^{(i)}) = \theta_1^{(i)}\theta_2^{(i)}\theta_5^{(i)}\theta_6^{(i)}\theta_7^{(i)}$$

$$p(t_{2,1};\boldsymbol{\theta}^{(i)}) = \theta_3^{(i)}\theta_4^{(i)}\theta_6^{(i)}\theta_8^{(i)}$$

EMアルゴリズム: Mステップ

- 書換規則の適用回数

r	P_r	$C(r; t_{11})$	$C(r; t_{12})$	$C(r; t_{21})$	$C'(r; t_{11})$	$C'(r; t_{12})$	$C'(r; t_{21})$
S → NP VP	1.0	1	1	1	?	?	1
VP → VP PP	θ_1	0	1	0	?	?	0
VP → V NP	θ_2	1	1	0	?	?	0
VP → V	θ_3	0	0		?	?	1
NP → NP PP	θ_4	1	0		?	?	1
NP → Jo					?	?	0
NP → M					?	?	1
PP → with_a_					?	?	0
V → sees					?	?	0
V → runs	θ_8	0	0	1	?	?	1

$$C'(r;t_{11}) = \frac{p(t_{11})}{p(t_{11}) + p(t_{12})} C(r;t_{11})$$

EMアルゴリズム: Mステップ

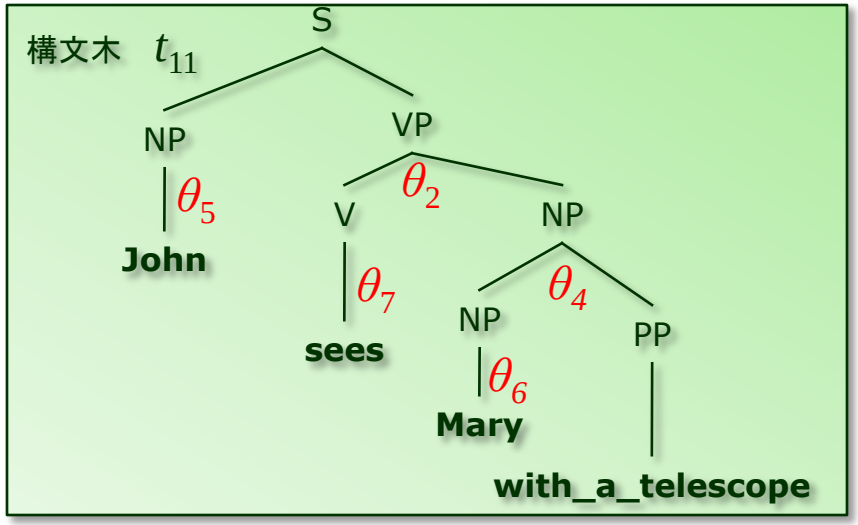
- 各構文木ごとの書換規則の適用回数の期待値

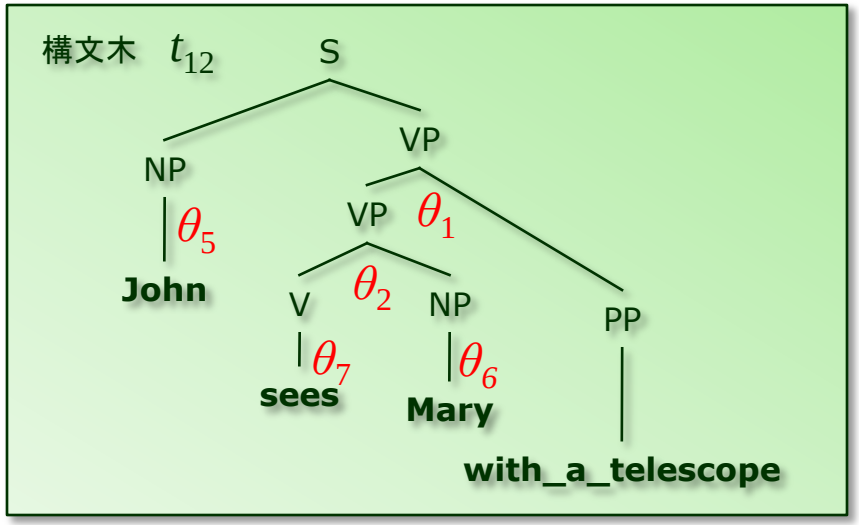
$$C'(r;t_{s,u}) = \frac{p(t_{s,u})}{\sum_{v} p(t_{s,v})} C(r;t_{s,u})$$

- 更新パラメータ

$$\theta_{A\to\alpha}^{(i+1)} = \frac{C'(A\to\alpha)}{\sum_{\beta} C'(A\to\beta)}$$

EMアルゴリズムの心





・新しいパラメータは単純な数え上げと同様に書換規則の適用頻度から求まる
・ただし、曖昧性のある文に対しては、書換規則の適用頻度の期待値として数え上げる
・構文木の確率は現在のパラメータから求まる

EMアルゴリズム: まとめ

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

$$p(t) = \prod_{r \in P} (\theta_r^{(i)})^{C(r;t)}$$

3. **[Mステップ]** $\theta^{(i+1)}$を求める

$$\theta_{A \to \alpha}^{(i+1)} = \frac{C'(A \to \alpha)}{\sum_{\beta} C'(A \to \beta)} \qquad C'(r;t_{s,u}) = \frac{p(t_{s,u})}{\sum_{v} p(t_{s,v})} C(r;t_{s,u})$$

4. 2.に戻る

EMアルゴリズム(一般) 1/2

- パラメータ: $\boldsymbol{\theta}$
- 入力: $\mathbf{x}$
- 隠れ状態: $\mathbf{z}$
- データ: $S=\{\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \ldots, \mathbf{x}_{(n)}\}$
- 対数尤度: $L_S(\boldsymbol{\theta})$

$$L_S(\boldsymbol{\theta}) = \mathrm{E}_S[\log p_{\boldsymbol{\theta}}(\mathbf{x})] = \mathrm{E}_S\left[\log \sum_{\mathbf{z}} p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})\right] = \mathrm{E}_S\left[\log \sum_{\mathbf{z}} q(\mathbf{z}\mid\mathbf{x}) \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q(\mathbf{z}\mid\mathbf{x})}\right]$$

$$\geq \mathrm{E}_S\left[\sum_{\mathbf{z}} q(\mathbf{z}\mid\mathbf{x}) \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q(\mathbf{z}\mid\mathbf{x})}\right] = F(q,\boldsymbol{\theta})$$

(Jensenの不等式)

$$\mathrm{E}_S[f(\mathbf{x})] = \frac{1}{n}\sum_i f(\mathbf{x}_{(i)})$$

EMアルゴリズム(一般) 2/2

- Eステップ

$$q^{(t+1)}(\mathbf{z}\,|\,\mathbf{x}) = \arg\max_{q(\mathbf{z}|\mathbf{x})} F(q, \boldsymbol{\theta}^{(t)}) = \arg\min_{q(\mathbf{z}|\mathbf{x})} KL(q(\mathbf{z}\,|\,\mathbf{x}) \,\|\, p_{\boldsymbol{\theta}^{(t)}}(\mathbf{z}\,|\,\mathbf{x})) = p_{\boldsymbol{\theta}^{(t)}}(\mathbf{z}\,|\,\mathbf{x})$$

$$KL(q \,\|\, p) = \mathrm{E}_q\left[\log \frac{q(\cdot)}{p(\cdot)}\right]$$

- Mステップ

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} F(q^{(t+1)}, \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \mathrm{E}_S\left[\sum_{\mathbf{z}} q^{(t+1)}(\mathbf{z}\,|\,\mathbf{x}) \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\right]$$

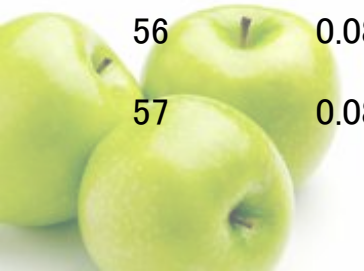
隠れ状態の確率とパラメータを交互に動かして、
Fを最大化

EMアルゴリズム: 局所解

- 極値を求めているので最適解とは限らない
- 良い解が得られるかどうかは初期値に依存している
  - 色々な初期値を試す
  - 他の頻度情報を使って初期値を設定

EMアルゴリズム:結果

i	θ 1	θ 2	θ 3	θ 4	θ 5	θ 6	θ 7	θ 8
1	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
2	0.2	0.4	0.4	0.333333	0.222222	0.444444	0.5	0.5
3	0.157895	0.421053	0.421053	0.351351	0.216216	0.432432	0.5	0.5
4	0.13422	0.43289	0.43289	0.360334	0.213222	0.426444	0.5	0.5
5	0.119484	0.440258	0.440258	0.365563	0.211479	0.422958	0.5	0.5
6	0.109661	0.44517	0.44517	0.368908	0.210364	0.420728	0.5	0.5
.....								
53	0.081358	0.459321	0.459321	0.377964	0.207345	0.414691	0.5	0.5
54	0.081358	0.459321	0.459321	0.377964	0.207345	0.41469	0.5	0.5
55	0.081358	0.459321	0.459321	0.377964	0.207345	0.41469	0.5	0.5
56	0.081357	0.459321	0.459321	0.377964	0.207345	0.41469	0.5	0.5
57	0.081357	0.459321	0.459321	0.377964	0.207345	0.41469	0.5	0.5

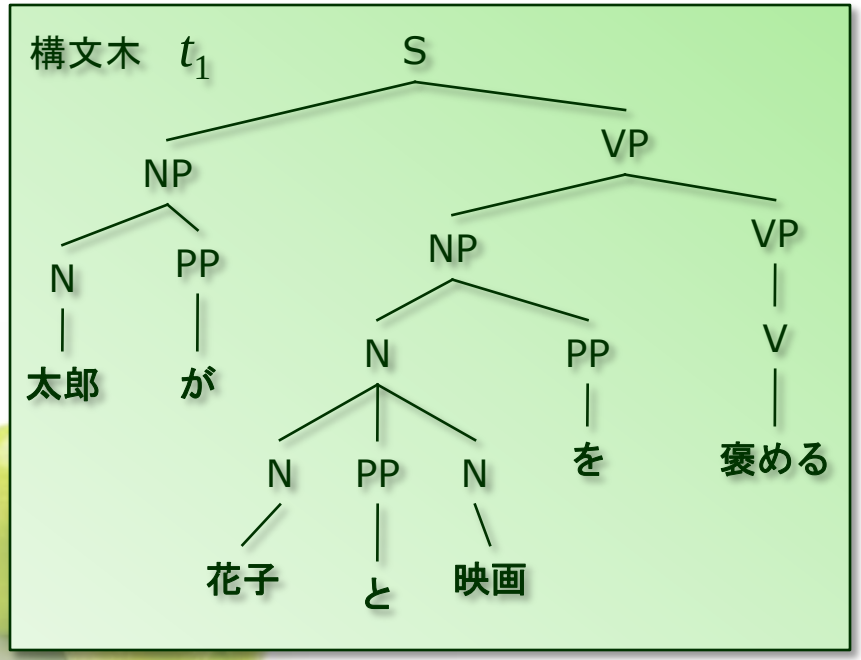


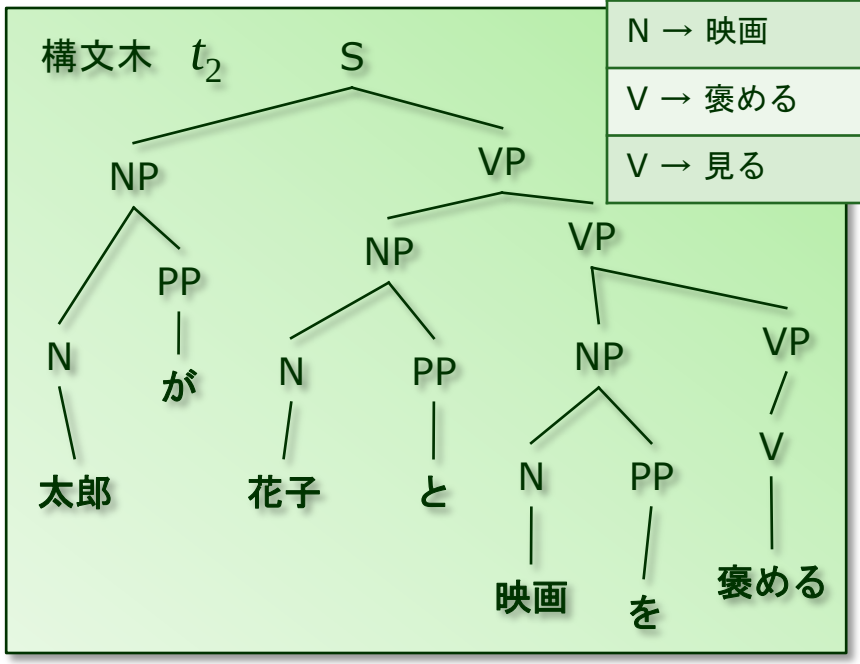
おまけ: 解析的に求めるのが難しいPCFGの例

r	P_r
S → NP VP	θ_1
NP → N PP	θ_2
N → N PP N	θ_3
VP → NP VP	θ_4
VP → V	θ_5
PP → が	θ_6
PP → を	θ_7
PP → と	θ_8
N → 太郎	θ_9
N → 花子	θ_{10}
N → 映画	θ_{11}
V → 褒める	θ_{12}
V → 見る	θ_{13}

- “太郎が花子と映画を褒める”

$$p(t_1) = \theta_3\theta_4\theta_5\theta_6\theta_7\theta_8\theta_9\theta_{10}\theta_{11}\theta_{12}$$
$$p(t_2) = \theta_4^2\theta_5\theta_6\theta_7\theta_8\theta_9\theta_{10}\theta_{11}\theta_{12}$$
$$\theta_3+\theta_9+\theta_{10}+\theta_{11}=1,\ \theta_4+\theta_5=1,\ \theta_6+\theta_7+\theta_8=1,\ \theta_{12}+\theta_{13}=1$$





頻度のディスカウンティング

- ゼロ頻度問題
  - ある単語がたまたま訓練コーパス中に出現しなかったら、その単語に対するパラメータは0になってしまう
  - その単語が出現するテストコーパスの構文木の確率は0になってしまう!
- 対策: 出現回数を補正

加算法 (additive method)

- ラプラス法
  - 頻度に1を加える

$$p(w_1 \cdots w_n) = \frac{C(w_1 \cdots w_n) + 1}{N + V}$$

N: 訓練データ中に出現した単語の総数

V: 出現確率の合計を1にするための定数(n単語列の異なり総数に等しい)

- 一般の方法(リッドストーン法とも呼ばれる)
  - 頻度に小さな値(δ)を加える

$$p(w_1 \cdots w_n) = \frac{C(w_1 \cdots w_n) + \delta}{N + \delta V}$$

  - δ=1/2の時、予期尤度推定法(expected likelihood estimation)、あるいはジェフリース・パークス法を呼ばれる

ヘルドアウト推定法

- 訓練データを二分割
  - 訓練データ
  - ヘルドアウトデータ(C^hをヘルドアウトデータ中の出現回数とする)
- 訓練データでの出現回数をヘルドアウトデータでの出現回数で置き換える

$$N_r = \sum_{w_1 \cdots w_n : C(w_1 \cdots w_n) = r} 1$$

$$C_r = \sum_{w_1 \cdots w_n : C(w_1 \cdots w_n) = r} C^h(w_1 \cdots w_n)$$

$$p(w_1 \cdots w_n) = \frac{C_r}{N_r N} \text{(ただし、} r = C(w_1 \cdots w_n))$$

削除推定法(deleted estimation)

- ヘルドアウト推定法のクロスバリデーション版
  - 訓練データとヘルドアウトデータの役割をさらに交換すれば2倍データが増える

グッド・チューリング推定法 (Good-Turing estimation)

- 出現回数の補正値として次のr*を用いる

$$N_r = \sum_{w_1 \cdots w_n : C(w_1 \cdots w_n) = r} 1$$

$$r^* = (r+1)\frac{N_{r+1}}{N_r}$$

- 出現確率

$$p(w_1 \cdots w_n) = \begin{cases} \dfrac{r^*}{N} & C(w_1 \cdots w_n) > 0 \\ \dfrac{N_1}{N_0 N} & C(w_1 \cdots w_n) = 0 \end{cases}$$

各種推定法による比較

- APコーパス中の2単語組の出現回数の推定

最尤推定	ラプラス法	ヘルドアウト法	削除推定法	グッド・チューリング法
0	0.000137	0.0000270	0.0000374	0.0000270
1	0.000274	0.448	0.396	0.446
2	0.000411	1.25	1.24	1.26
3	0.000548	2.24	2.23	2.24
4	0.000685	3.23	3.22	3.24
5	0.000822	4.21	4.22	4.22
6	0.000959	5.23	5.20	5.19
7	0.001096	6.21	6.21	6.21
8	0.001233	7.21	7.18	7.24
9	0.001370	8.26	8.18	8.25

まとめ

- PCFGとEMアルゴリズム
  - EMアルゴリズム
  - ディスカウンティング