



# N-gramモデル

東京大学 情報基盤センター  
(総合文化研究科、情報学府 兼  
担)

中川裕志

## 文字列の統計的モデル

- 1次元文字列  $c_1 c_2 c_3, \dots, c_n = C_1^n$
- 1次元単語列  $w_1 w_2 w_3, \dots, w_n = w_1^n$ 
  - とりあえず単語列の場合として話を進める
- 単語列  $w_1^n$  の生起確率を各単語の条件付確率でモデル化

$$P(w_1^n) = P(w_1)P(w_2 | w_1)P(w_3 | w_1 w_2), \dots, P(w_n | w_1^{n-1})$$

- 各項の  $w_n$  の条件付確率が文字列のモデル。これが直前の  $N$  単語に依存するモデルを  $N$ -gramモデルという

$$P(w_n | w_1^{n-1}) = P(w_n | w_{n-N+1}^{n-1})$$

## Bag of Wordsモデル

- 1-gramモデルだと、言語モデルは $P(w_n)$ すなわち、各単語の(コーパスにおける)生起確率だけで決まる。
- 大胆な近似だが、計算が容易で、計算量も少ない。
- 情報検索では基本的モデルとして使われる

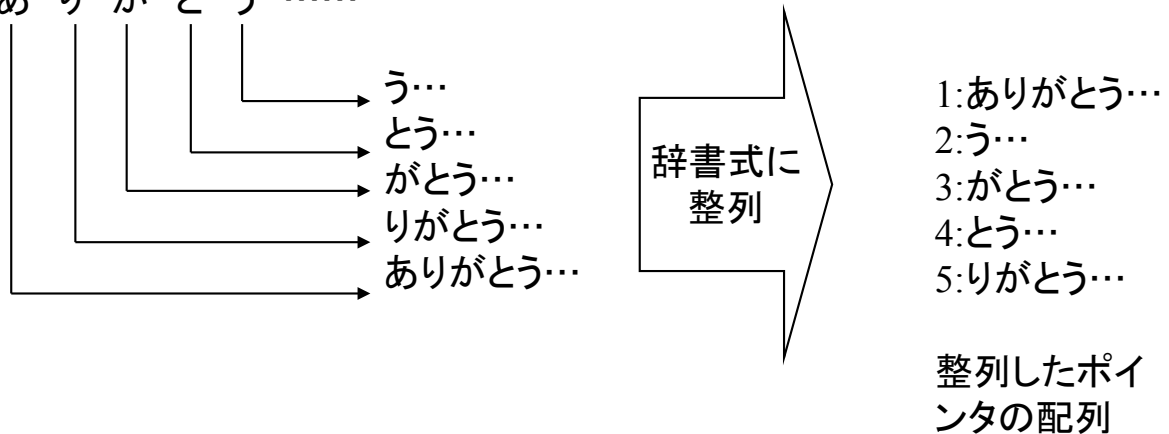
## Nグラム

- ◆ Nグラムとは言語を特徴つける簡単な方法(言語モデル)
- ◆ ある言語単位(音素、文字、単語、品詞など)を選ぶ。その言語単位のN個連続をマルコフモデルで表したものをNグラム(N-gram)という。特に言語単位を陽に指定する場合、「言語単位名Nグラム」(例えば、単語2グラム)という。
- ◆ 単独の言語単位のモデルを unigram、2個の連続を bigram、3個の連続を trigram という。(zero-gram とは、全ての単語が等確率で生起するモデル)
- ◆ 異なり数を計算してみよう。
  - ◆ (1) 英語の文字2グラムの総数
  - ◆ (2) 日本語のモーラ2グラムの総数。
  - ◆ **モーラ(拍)**とは、ひらがな1文字同じ長さの音の単位。「ん」「っ」「ー」は1モーラ。
  - ◆ なお、**音節(syllable)**とは、「(子音)(半母音)母音(モーラ音素)」
  - ◆ (3) 日本語の文字2グラムの総数
  - ◆ (4) 日本語の単語2グラムの総数
  - ◆ (5) 日本語の品詞2グラムの総数

## Nグラム の 計算

- ある言語におけるNグラムの種類の総数はとても大きすぎて計算できない場合が多い。実際のテキストにおいて出現したNグラムによって言語(の部分集合)を特徴つける。そこで、テキストにおけるNグラムの計算法が必要。

- あ り が と う ……



- 整列したポインタの配列をサフィックスアレイという。先頭部分に同じ文字列を持つものが隣接ないし近接する。

- 近所を見回せば、同じNグラムが何個あるかという統計を簡単に計算できる。

## KWIC ( Key Word In Context )

- ある言語表現がどのような文脈に現れるかを、与えられたコーパスにおいて列挙したもの。
- 辞書式に整列したテキストへのポインタの配列 (Nグラム の計算に利用するもの) を使えば、容易に抽出できる。
- Nグラム の計算 のページの「Nグラム」に対するKWICは以下の通り。

前の文脈	Key Word	後の文脈
ある言語における ストにおいて出現した テキストにおける	Nグラム Nグラム Nグラム	の総数はとても大きすぎて によって言語 (の部分集合) の計算法が必要。

- Key Word がどのような単語や表現と共起するかという情報を得られる。共起情報は自然言語処理において必須の情報。

## Nグラム の 確率モデル

- ◆ NグラムはN言語単位の連鎖のモデルだが、言語単位としては、文字、単語、品詞などなんでも採用できる。
- ◆ まず、N言語単位の連鎖は、 $C(w_1 w_2 \dots w_n)$ 、ただしCはコーパス中の頻度。
- ◆ コーパスの文を文字のN重マルコフ過程つまり直前のN文字から次に現れる文字を予測するモデルにしたい。一般にN重マルコフ過程とは、現在の状態がN個前の入力に依存してきまる確率プロセス
- ◆ つまり、である。

$$p(w_i | w_{i-n} \dots w_{i-1})$$

- ◆ これは条件つき確率で

$$p(w_i | w_{i-n} \dots w_{i-1}) = \frac{C(w_{i-n} \dots w_{i-1} w_i)}{C(w_{i-n} \dots w_{i-1})}$$

## N グラムの生起確率を求める その 1

### □ 最尤推定法

文字のN-1重マルコフ過程  $p(w_n | w_1 \dots w_{n-1}) = \frac{C(w_1 \dots w_{n-1} w_n)}{C(w_1 \dots w_{n-1})}$

相対頻度CからNグラムが生起確率を推定

### □ Nが大きいと信頼性の高いNグラム推定ができない。

### □ 相対頻度が0のNグラムがたくさん現れる。(データスパースネス問題)

### □ 加算法: 単に分母分子に適当な数を足す。

$$P(w_n | w_1 \dots w_{n-1}) = \frac{C(w_1 \dots w_{n-1} w_n) + \delta}{N + V\delta} = \lambda \frac{C(w_1 \dots w_{n-1} w_n)}{N} + (1 - \lambda) \frac{1}{V}$$

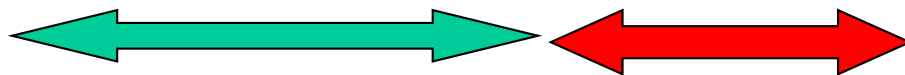
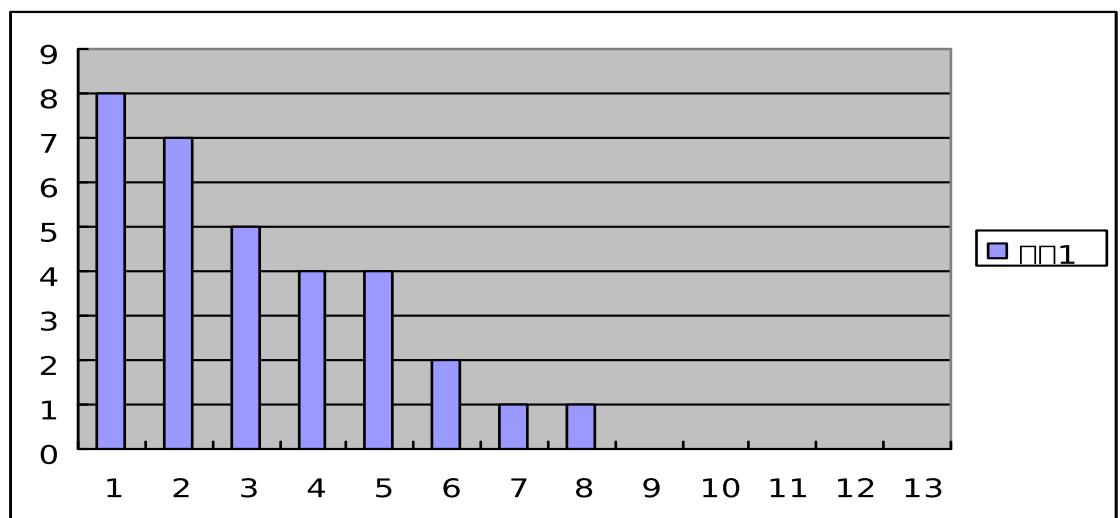
$$\text{where } \lambda = \frac{N}{N + \delta V}$$

分子が0の場合は単に $\delta$ を分子とする。簡単だがあまり精度がよくない。

Vはコーパス中の異なり語数



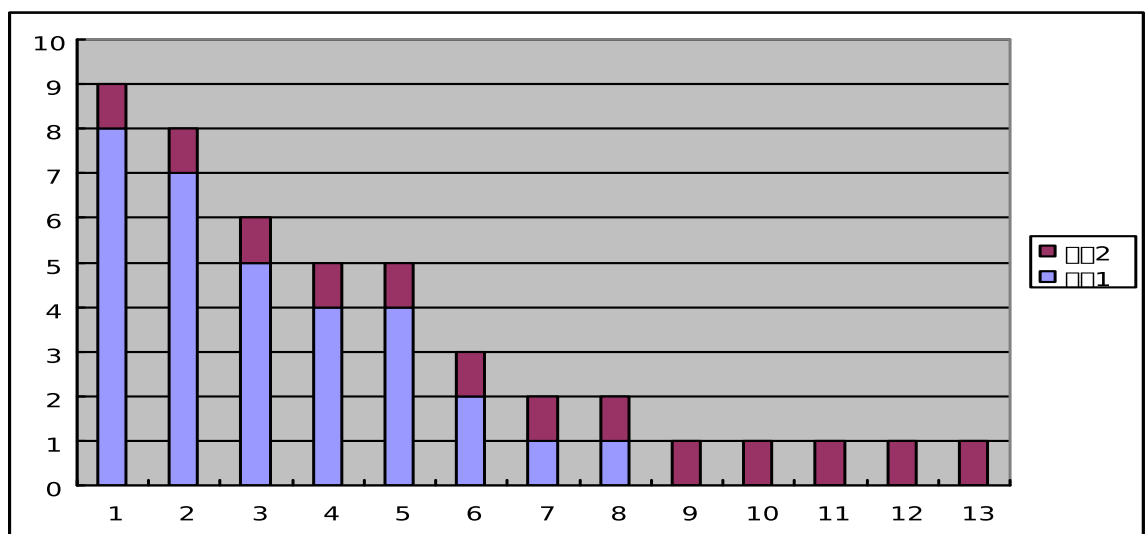
## Back-off smoothing (元データの頻度)



実際に出現した単語(8個)

出現していないが、これから出現する可能性がある単語(5個)

# 各単語の頻度に $\delta (=1)$ を加算



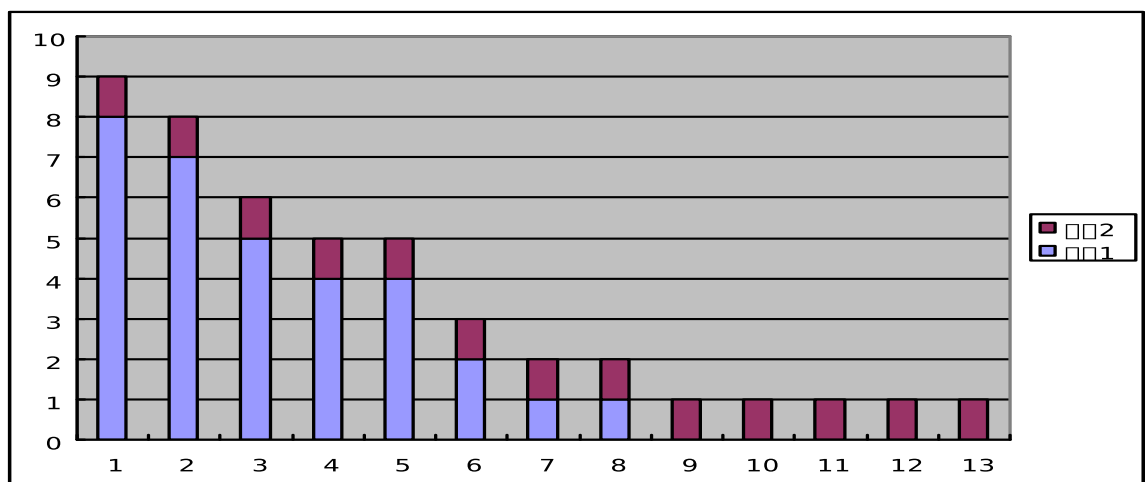
実際に出現した単語(8個)



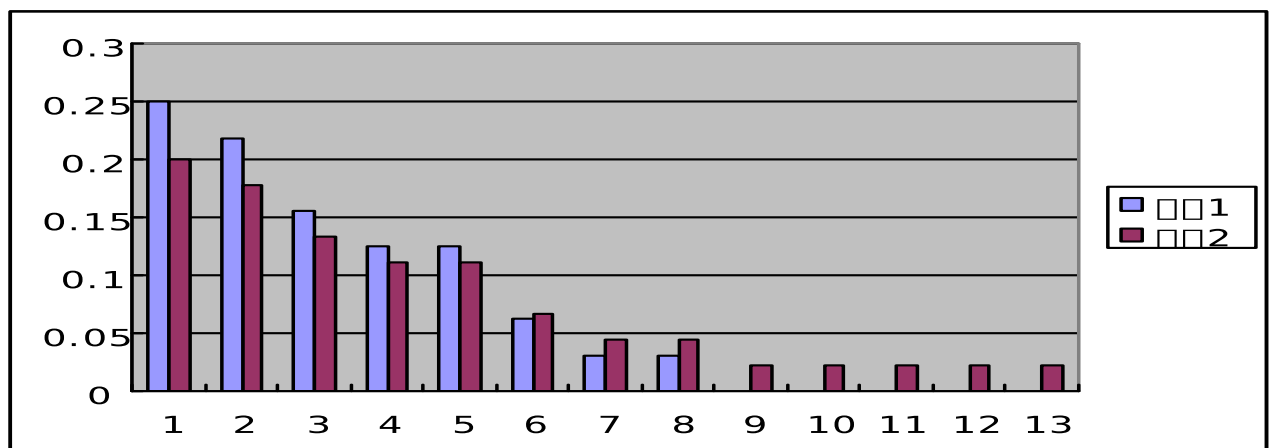
出現していないが、これから出現する可能性がある単語(5個)

## Back-off smoothing (確率を計算しなおす)

原データ



確率



## Nグラムが生起確率を求める その2 Good-Turingの推定

### □ Good-Turingの推定

語数 $N$ のコーパス中で $r$ 回出現する異なり単語数を $n_r$ とする。すると

$$N = \sum_{r>0} r n_r = n_1 + 2n_2 + 3n_3 + \dots$$

ここでコーパスに $r$ 回出現する単語 $w$ の頻度を次の式で推定するのがGood-Turingの推定

$$r^* = (r+1) \frac{n_{r+1}}{n_r}$$

□ Good-Turingの推定

語数Nのコーパス中でr回出現する単語の数を $n_r$ とする。すると

$$N = \sum_{r>0} r n_r = n_1 + 2n_2 + 3n_3 + \dots$$

ここでコーパスにr回出現する単語wの頻度を次の式で推定するのがGood-Turingの推定

$$r^* = (r+1) \frac{n_{r+1}}{n_r}$$

□ ここで0回出現した単語の出現頻度の期待値 $0^*$ は  $n_1$

$$0^* = \frac{n_1}{n_0} = \frac{\text{□□□□}}{\text{□□□□□□□□□□□□□□}} = \frac{\text{□□}}{\text{□□}}$$

□ 一方、1回以上出現した単語の相対頻度の総和を求めると

$$\sum_{r>0} \frac{n_r r^*}{N} = 1 - \frac{n_1}{N}$$

つまり、 $\frac{n_1}{N}$  がコーパスに出現しない全単語の頻度の合計の推定確率

□ なお、 $d = \frac{r^*}{r}$  をディスカウント係数という。

## Good-Turingの推定の導出

- 母集団における異なり単語数をMとする
- 母集団における単語 $w_i$ の出現確率を $P(w_i)$
- $w_i$  が語数(サイズ)Nのコーパス中で出現する回数を $C(w_i)$  当然  $\sum_{i=1}^M C(w_i) = N$
- 単語 $w$ がコーパス中に $r$ 回出現したとき、 $w$ の母集団での生起確率および出現回数の期待値は

$$E[P(w) | C(w) = r] = \sum_{i=1}^M P(w = w_i | C(w) = r) P(w_i) \quad - (1)$$

$$r^* = E[r | C(w) = r] = E[P(w) | C(w) = r] N \quad - (2)$$

- サイズ $N$ のコーパスにおける単語の出現確率分布を2項分布とすると

$$\begin{aligned}
 P(w = w_i \mid C(w) = r) &= \frac{P(C(w_i) = r)}{\sum_{i=1}^M P(C(w_i) = r)} \\
 &= \frac{{}_N C_r P(w_i)^r (1 - P(w_i))^{N-r}}{\sum_{i=1}^M {}_N C_r P(w_i)^r (1 - P(w_i))^{N-r}} \quad - (3)
 \end{aligned}$$

この結果を(1)に代入すると

$$E[P(w) \mid C(w) = r] = \frac{\sum_{i=1}^M {}_N C_r P(w_i)^{r+1} (1 - P(w_i))^{N-r}}{\sum_{i=1}^M {}_N C_r P(w_i)^r (1 - P(w_i))^{N-r}} \quad - (4)$$

サイズNのコーパス中にr回出現する単語の総数の期待値

$$E_N[N_r] = \sum_{i=1}^M P(C(w_i) = r) = \sum_{i=1}^M {}_N C_r P(w_i)^r (1 - P(w_i))^{N-r}$$

$$\square\square\square \quad (4)\square \quad {}_N C_r = \frac{r+1}{N+1} {}_{N+1} C_{r+1} \quad \square\square\square\square\square\square\square\square\square \quad \square\square\square\square\square$$

$$E[P(w) | C(w) = r] = \frac{r+1}{N+1} \frac{E_{N+1}(N_{r+1})}{E_N(N_r)} \quad - (5)$$

この結果を使って(2)式の $r^*$ を求めると

$$r^* = E[P(w) | C(w) = r]N = N \frac{r+1}{N+1} \frac{E_{N+1}(N_{r+1})}{E_N(N_r)} \quad - (6)$$

ここでNが十分大きく、 $E_N(N_r)$ をコーパス中出现頻度 $N_r$ で近似すると

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



## Nグラムが生起確率を求める その4 (バックオフ スムースジング)

- Good-Turingの推定を基礎にした頻度=0のbi-gramの頻度推定

$$p(w_2|w_1) = \frac{C^*(w_1, w_2)}{C(w_1)} = \frac{C^*(w_1, w_2)}{C(w_1, w_2)} \times \frac{C(w_1, w_2)}{C(w_1)} = d_{C(w_1, w_2)} \frac{C(w_1, w_2)}{C(w_1)}$$

- この計算によればコーパス中出现する bi-gram の確率の和は1より小さい

$$\beta(w_1) = 1 - \sum_{w_2: C(w_1, w_2) > 0} p(w_2|w_1)$$

- この  $\beta(w_1)$  は  $w_1$  に対して  $C(w_1, w_2) = 0$  となる全  $w_2$  の条件つき確率の和。
- これを  $C(w_1, w_2) = 0$  なる単語列の確率に分配して  $p(w_2|w_1)$  を求める

$$p(w_2|w_1) = \frac{\beta(w_1)p(w_2)}{\sum_{w_2: C(w_1, w_2)=0} p(w_2)} = \frac{1 - \sum_{w_2: C(w_1, w_2) > 0} p(w_2|w_1)}{1 - \sum_{w_2: C(w_1, w_2) > 0} p(w_2)} \times p(w_2)$$

## Nグラムの拡張 その1 クラスモデル

- Nグラム クラスモデル：例えば品詞のような単語のクラスから次にくる単語を予測するモデル。単語を $w$ 、品詞を $c$ とする。すると、1重マルコフモデルで品詞-単語の bi-gramクラスモデルは、

$$\begin{aligned} p(w_n | w_{n-1}) &= p(w_n | c_n) p(c_n | c_{n-1}) p(c_{n-1} | w_{n-1}) \\ &= p(w_n | c_n) p(c_n | c_{n-1}) \\ \square p(c_{n-1} | w_{n-1}) &= 1 \end{aligned}$$

ただし、 $c_i$  は単語  $w_i$  の属するクラス(例えば品詞)  
一般には単語は複数の品詞を持つ。よって、次のように書くべき。

$$p(w_n | w_{n-1}) = \sum_{c_n} p(w_n | c_n) p(c_n | c_{n-1})$$

例えば、クラスとして品詞を使うと品詞の異なり数は単語の異なり数よりはるかに少ないので、 $N$ の大きいNグラムも計算できる。

## Nグラムの拡張 その2 キャッシュモデル

- 直前の  $n$  語の中に、現在の単語と同じ単語が何回現れるか。
- 問題:このモデルを式で書けますか？
-