

# NLP Programming Tutorial 7 - トピックモデル

Graham Neubig  
奈良先端科学技術大学院大学 (NAIST)

# 文章のトピック

- 文章には様々なトピックが存在する

Cuomo to Push for Broader  
Ban on Assault Weapons

...  
...  
...  
...

2012 Was Hottest  
Year in U.S. History

...  
...  
...  
...

# 文章のトピック

- 文章には様々なトピックが存在する

Cuomo to Push for Broader  
Ban on Assault Weapons

...  
...  
...  
...

ニューヨーク  
政治  
武器  
犯罪

2012 Was Hottest  
Year in U.S. History

...  
...  
...  
...

天気  
気候  
統計  
アメリカ

# トピックモデル

- トピックモデルでは文章  $X$  に対してトピック  $Y$  を発見



- 構造予測の一種

## 確率的生成モデル

- 文章  $X$  とトピック  $Y$  が何かの過程によって同時に生成されたとする

$$P(Y, X)$$

- 同時確率が高ければ、条件付き確率も高い：

$$\operatorname{argmax}_Y P(Y|X) = \operatorname{argmax}_Y P(Y, X)$$

# トピックを考慮した文の生成モデル

- 単語列  $X$  とトピック列  $Y$ :

$X =$  Cuomo to Push for Broader Ban on Assault Weapons

$Y =$  NY 機能 政治 機能 政治 政治 機能 犯罪 犯罪

- まずトピックを独立に生成:

$$P(\mathbf{Y}) = \prod_{i=1}^I P(y_i)$$

- その次、各単語をトピックに基づいて生成:

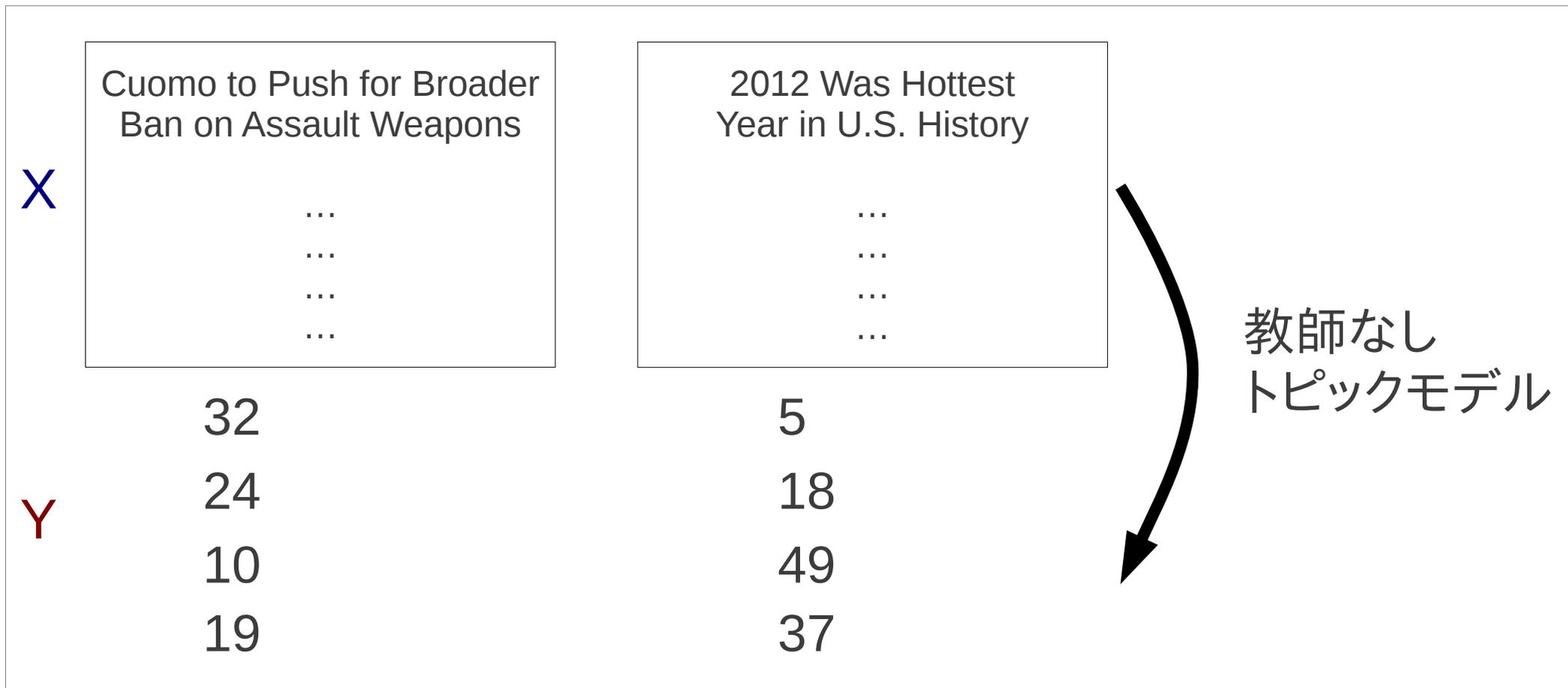
$$P(\mathbf{X}|\mathbf{Y}) = \prod_{i=1}^I P(x_i|y_i)$$

---


$$P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{X}|\mathbf{Y})P(\mathbf{Y}) = \prod_{i=1}^I P(x_i|y_i)P(y_i)$$

# 教師なしトピックモデル

- 文章  $X$  のみからトピックらしいクラス  $Y$  を発見



- 前と違って  $Y$  の記された学習データがない！

# 潜在的ディリクレ配分法 (Latent Dirichlet Allocation: LDA)

- トピックモデルの中で最も一般的
- まずモデルのパラメータ  $\theta$  を生成： $P(\theta)$
- 各文章に対して  $X$ :
  - 文章のトピック分布  $T_i$  を生成： $P(T_i|\theta)$
  - $X_i$  の各単語  $x_{i,j}$  に対して：
    - トピック  $y_{i,j}$  を生成： $P(y_{i,j}|T_i, \theta)$
    - 単語  $x_{i,j}$  を生成： $P(x_{i,j}|y_{i,j}, \theta)$

$$P(X, Y) = \int_{\theta} P(\theta) \prod_i P(T_i|\theta) \prod_j P(y_{i,j}|T_i, \theta) P(x_{i,j}|y_{i,j}, \theta)$$

# 最尤推定

- 単語  $X$  とトピック  $Y$  が与えられたとしたら：

$X_1$	=	Cuomo	to	Push	for	Broader	Ban	on	Assault	Weapons
$Y_1$	=	32	7	24	7	24	24	7	10	10

- 各文章のトピック分布を決定：

$$P(y|Y_i) = c(y, Y_i) / |Y_i| \quad \text{e.g.: } P(y=24|Y_1) = 2/9$$

- 各トピックの単語分布を決定：

$$P(x|y) = c(x, y) / c(y) \quad \text{e.g.: } P(x=\text{assault}|y=10) = 1/2$$

# 隠れ変数

- 問題：  $y_{ij}$  の値は与えられていない
- 解決策： 教師なし学習を利用
- 教師なし学習の手法例：
  - EM アルゴリズム
  - 変分ベイズ
  - サンプリング

# サンプリングの例

- ある分布に従ってサンプルを生成：

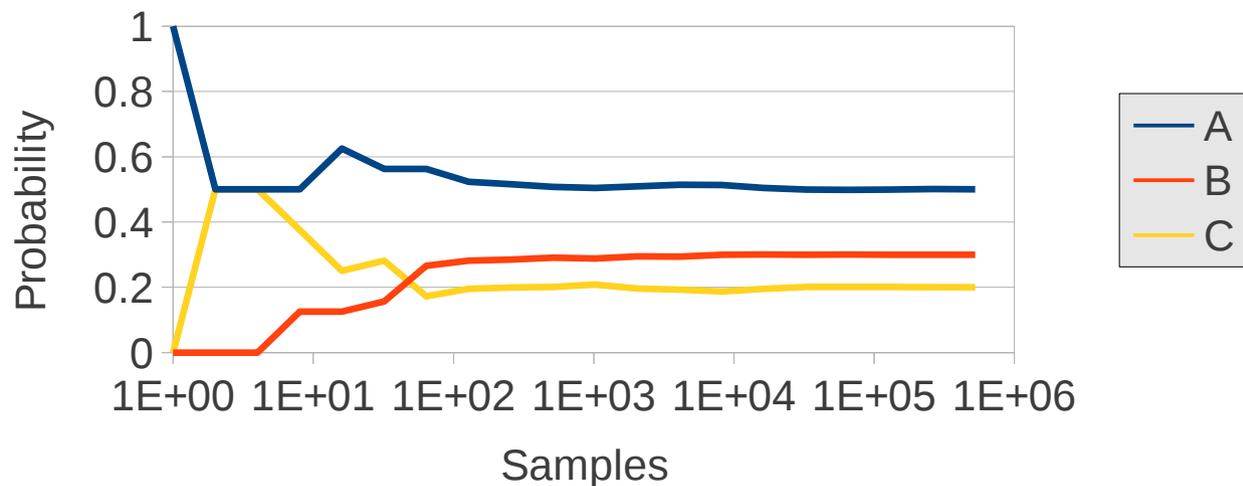
分布：  $P(A)=0.5$   $P(B)=0.3$   $P(C)=0.2$

サンプル： B B C A A C A B B A ...

- サンプルを数え上げて割ったら確率が近似可能：

$$P(A) = 4/10 = 0.4, \quad P(B) = 4/10 = 0.4, \quad P(C) = 2/10 = 0.2$$

- サンプルが増えれば近似の精度も増える：



# アルゴリズム

```
SAMPLEONE(probs[])
```

```
z = SUM(probs)
```

確率の和（正規化項）を計算

```
remaining = RAND(z)
```

[0,z) の乱数を一様分布によって生成

```
for each i in 0 .. probs.size-1
```

probs の各項目を検証

```
    remaining -= probs[i]
```

現在の確率を引く

```
    if remaining <= 0
```

0 より小さい場合、返す

```
        return i
```

全ての確率が終わっても返さない場合はバグでエラー終了！

# ギブスサンプリング

- 2つの変数を分布  $P(X, Y)$  からサンプルしたい
  - ...  $P(X, Y)$  からサンプリングすることが不可
  - ...  $P(X|Y)$  と  $P(Y|X)$  からサンプリングすることが可
- ギブスサンプリングでは、変数を1個ずつサンプリングする
- 各イタレーション：
  - Xを固定し、Yを  $P(Y|X)$  に従ってサンプリング
  - Yを固定し、Xを  $P(X|Y)$  に従ってサンプリング

## ギブスサンプリングの例

- 親 A と子 B は買い物している、それぞれの性別は？  
 $P(\text{母} | \text{娘}) = 5/6 = 0.833$     $P(\text{母} | \text{息子}) = 5/8 = 0.625$   
 $P(\text{娘} | \text{母}) = 2/3 = 0.667$     $P(\text{娘} | \text{父}) = 2/5 = 0.4$

- 

初期状態：母 / 娘

A をサンプル：  $P(\text{母} | \text{娘}) = 0.833$ , 母を選んだ！

B をサンプル：  $P(\text{娘} | \text{母}) = 0.667$ , 息子を選んだ！

$c(\text{母}, \text{息子})++$

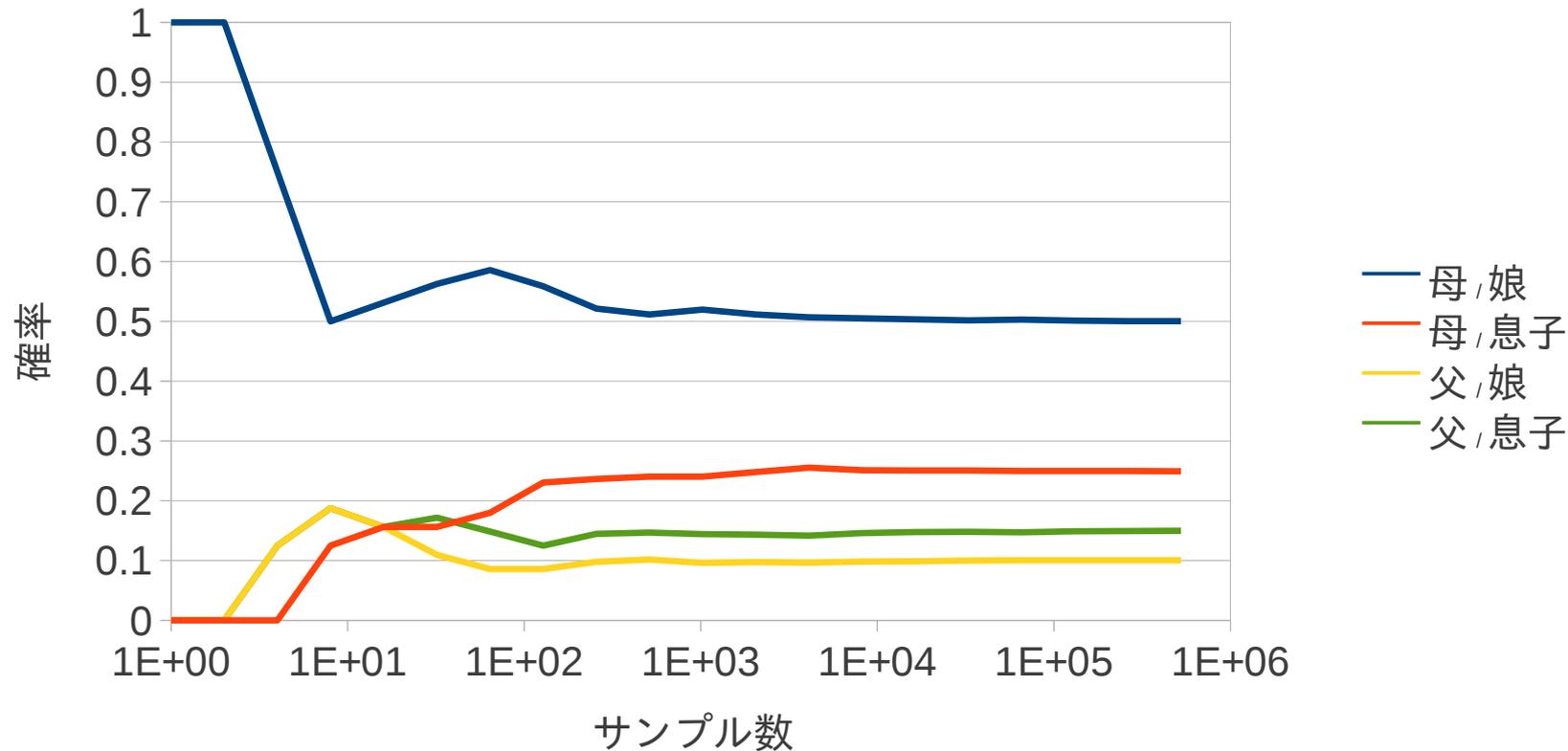
A をサンプル：  $P(\text{母} | \text{息子}) = 0.625$ , 母を選んだ！

B をサンプル：  $P(\text{娘} | \text{母}) = 0.667$ , 娘を選んだ！

$c(\text{母}, \text{娘})++$

...

# 実際にやってみると



- 同時確率の式を手で解いてこの結果を確認できる

# トピックモデルのサンプリング (1)

- $y_{i,j}$  を1つずつ :

$X_1 =$  Cuomo to Push for Broader Ban on Assault Weapons  
 $Y_1 =$  5 7 4 7 3 4 7 6 6

- まず  $y_{i,j}$  をカウントから削除、確率を再計算

{0, 0, 1/9, 2/9, 1/9, 2/9, 3/9, 0}

↓  
{0, 0, 1/8, 2/8, 1/8, 2/8, 2/8, 0}

## トピックモデルのサンプリング (2)

- $y_{i,j}$  を 1 つずつ :

$X_1 =$  Cuomo to Push for Broader Ban on Assault Weapons  
 $Y_1 =$  5 7 4 ??? 3 4 7 6 6

- トピック確率と単語確率を掛け合わせる :

$$\begin{aligned}
 P(y_{i,j} | T_i) &= \{ 0, 0, 0.125, 0.25, 0.125, 0.25, 0.25, 0 \} \\
 &\quad \text{コーパス全体から計算} \\
 &\quad * \\
 P(x_{i,j} | y_{i,j}, \theta) &= \{ 0.01, 0.02, 0.01, 0.10, 0.08, 0.07, 0.70, 0.01 \} \\
 &\quad = \\
 P(x_{i,j} y_{i,j} | T_i, \theta) &= \{ 0, 0, 0.00125, 0.01, 0.01, 0.00875, 0.175, 0 \} / Z
 \end{aligned}$$

正規化係数  $\nearrow^{17}$

## トピックモデルのサンプリング (3)

- 確率分布から1つの値をサンプリング：

$$P(x_{ij}, y_{ij} | T_i, \theta) = \{ 0, 0, 0.00125, 0.01, 0.01, 0.00875, 0.175, 0 \} / Z$$

- トピックを更新：

$$\begin{array}{l} X_1 = \text{Cuomo to Push for Broader Ban on Assault Weapons} \\ Y_1 = \quad 5 \quad 7 \quad 4 \quad 6 \quad 3 \quad 4 \quad 7 \quad 6 \quad 6 \end{array}$$

- カウントと確率を更新：

$$\{0, 0, 1/8, 2/8, 1/8, 2/8, 2/8, 0\}$$



$$\{0, 0, 1/9, 2/9, 1/9, 3/9, 2/9, 0\}$$

# ディリクレ分布による平滑化：

- 問題：多くのカウントが0 → 多くの確率が0  
→ 局所解に陥る
- 解決策：確率の平滑化

平滑化なし

$$P(x_{i,j}|x_{i,j}) = \frac{c(x_{i,j}, y_{i,j})}{c(y_{i,j})}$$

→

平滑化有り

$$P(x_{i,j}|y_{i,j}) = \frac{c(x_{i,j}, y_{i,j}) + \alpha}{c(y_{i,j}) + \alpha * N_x}$$

$$P(y_{i,j}|Y_i) = \frac{c(y_{i,j}, Y_i)}{c(Y_i)}$$

→

$$P(y_{i,j}|Y_i) = \frac{c(y_{i,j}, Y_i) + \beta}{c(Y_i) + \beta * N_y}$$

- $N_x$  と  $N_y$  はそれぞれ単語とトピックの異なり数
- 確率に対してディリクレ分布に基づく事前分布の利用と等しい（LDAの論文を参照）

## 実装：初期化

```
make vectors xcorpus, ycorpus # 各 x, y を格納
make map xcounts, ycounts # カウントの格納
for line in file
  docid = size of xcorpus # この文章の ID を獲得
  split line into words
  make vector topics # 単語のトピックをランダム初期化
  for word in words
    topic = RAND(NUM_TOPICS) # [0,NUM_TOP) の間
    append topic to topics
    ADDCOUNTS(word, topic, docid, 1) # カウントを追加
  append words (vector) to xcorpus
  append topics (vector) to ycorpus
```

# 実装：カウントの追加

**ADDCOUNTS**(*word, topic, docid, amount*)

*xcounts*["topic"] += *amount*  
*xcounts*["word|topic"] += *amount*

$$P(x_{i,j}|y_{i,j}) = \frac{c(x_{i,j}, y_{i,j}) + \alpha}{c(y_{i,j}) + \alpha * N_x}$$

*ycounts*["docid"] += *amount*  
*ycounts*["topic|docid"] += *amount*

$$P(y_{i,j}|Y_i) = \frac{c(y_{i,j}, Y_i) + \beta}{c(Y_i) + \beta * N_y}$$

バグチェック

< 0 の場合はエラー終了

# 実装：サンプリング

**for** many iterations:

**for**  $i$  in  $0:\text{SIZE}(x\text{corpus})$ :

**for**  $j$  in  $0:\text{SIZE}(x\text{corpus}[i])$ :

$x = x\text{corpus}[i][j]$

$y = y\text{corpus}[i][j]$

**ADDCOUNTS**( $x, y, i, -1$ ) # 各カウントの減算 (-1)

**make vector**  $probs$

**for**  $k$  in  $0 .. \text{NUM\_TOPICS}-1$ :

**append**  $P(x|k) * P(k|Y)$  **to**  $probs$  # トピック  $k$  の確率

$new\_y = \text{SAMPLEONE}(probs)$

// +=  $\log(probs[new\_y])$  # 対数尤度の計

**ADDCOUNTS**( $x, new\_y, i, 1$ ) # 各カウントの加算

$y\text{corpus}[i][j] = new\_y$

**print** //

**print out**  $wcounts$  and  $tcunts$

# 演習課題

# Exercise

- **実装** learn-lda
- **テスト** (NUM\_TOPICS=2)
  - **入力**: `test/07-train.txt`
  - **正解**:
    - 正解はない！ (サンプリングはランダムなので)
    - しかし、「abcd」と「efgh」に分かれる確率が高い
- **学習** `data/wiki-en-documents.word` を使って
- **検証** 発見されたトピックは直感に沿うのか？ (機能語を削除して、各トピックで頻度の高い内容語を見ると良い)
- **チャレンジ** トピック数を事前に決めなくても良いようにモデルを変更 (ノンパラメトリックベイズで検索)

Thank You!