ノンパラメトリックベイズ法

Graham Neubig
2011 年 5 月 10 日
@NAIST

概要

- ノンパラメトリックベイズ法について
  - ベイズ法の基礎理論
  - サンプリングによる推論
  - サンプリングを利用した HMM の学習
  - 有限 HMM から無限 HMM へ
  - 近年の展開（サンプリング法、モデル化法）
  - 音声処理・言語処理のおける応用
- 基本は離散分布の教師なし学習

Non-parametric
ノンパラメトリック

パラメータの数は
予め決まっていない
（無限）

Bayes
ベイズ

パラメータに
事前分布をかけ、
パラメータの
分布を扱う

確率モデルの分類

	パラメータの事前分布	パラメータ数（クラス数）	離散分布の代表	連続分布の代表
最尤推定	無	有限	多項式分布	ガウス分布
ベイズ推定	有	有限	多項式＋ディリクレ分布	ガウス分布＋ガウス分布
ノンパラベイズ	有	無限	多項式＋ディリクレ過程	ガウス過程

今日の話

ベイズ法の基礎

最尤推定

- ある観測データがある（幼稚園児の年齢？）

  X = 1 2 4 5 2 1 4 4 1 4

- 頻度を数える $c(x)=\{3,2,0,4,1\}$
- 頻度を全体の頻度でわることで、確率を得る

$$P(x=i)=\frac{c(x=i)}{c(\cdot)}$$

多項式分布

$$P(x)=\vec{\theta}=\{0.3,0.2,0,0.4,0.1\}$$

ベイズ推定

- 最尤推定はスパースなデータに弱い
- 実際のパラメータは知らない

$c(x)=\{3,2,0,4,1\}$ なら

$\vec{\theta}=\{0.3,0.2,0,0.4,0.1\}$ かもしれないし

$\vec{\theta}=\{0.35,0.05,0.05,0.35,0.2\}$ かもしれない

- ベイズ推定で一意に決めずに、パラメータにも分布を持たせて、パラメータの期待値で確率を計算

$$P(x=i)=\int \theta_i P(\vec{\theta}|X)d\vec{\theta}$$

パラメータの分布をどうやって計算？

- ベイズ則で分解

$$P(\theta|X) = \frac{\overbrace{P(X|\theta)}^{\text{尤度}}\overbrace{P(\theta)}^{\text{事前分布}}}{\underbrace{\int P(X|\theta)P(\theta)d\theta}_{\text{正則化定数}}}$$

- 尤度は一般的にモデルの定義に基づいて簡単に計算
- 事前分布は適当に決める
- 正則化定数の計算は積分処理が必要で難しい…
  - … が、共役事前分布を利用すると簡単！

共役事前分布

- 定義：尤度関数と事前分布の掛け算の結果は事前分布と同じ形である

多項式尤度 ＊ ディリクレ事前分布 ＝ ディリクレ分布

ガウス尤度 ＊ ガウス事前分布 ＝ ガウス分布

同じ

- ディリクレ分布とガウス分布の正則化定数は既知であるため、積分をしなくても求まる

ディリクレ分布・過程

- ディリクレ分布は多項式分布に生成確率を与える

例えば：

$$P(\{0.3, 0.2, 0.01, 0.4, 0.09\}) = 0.000512$$
$$P(\{0.35, 0.05, 0.05, 0.35, 0.2\}) = 0.0000963$$

- 確率として成り立つ実数の集合 $\{\theta_1, \ldots, \theta_n\}$ にだけ確率を与える

$$\forall_{\theta_i} 0 \leqslant \theta_i \leqslant 1 \qquad \sum_{i=1}^{n} \theta_i = 1$$

- ディリクレ過程はディリクレ分布の一般化
  - 無限の集合でも扱える

ディリクレ過程

資料で欠落

- 式：

$$P(\vec{\theta};\alpha,P_{base})=\frac{1}{Z}\prod_{i=1}^{n}\theta_i^{\alpha P_{base}(x=i)-1}$$

  - α は「集中パラメータ」、大きければ大きいほど確率のピークが尖っている
  - P_{base} は「基底測度」、θ の期待値を表す（後述）

ディリクレ分布における書き方

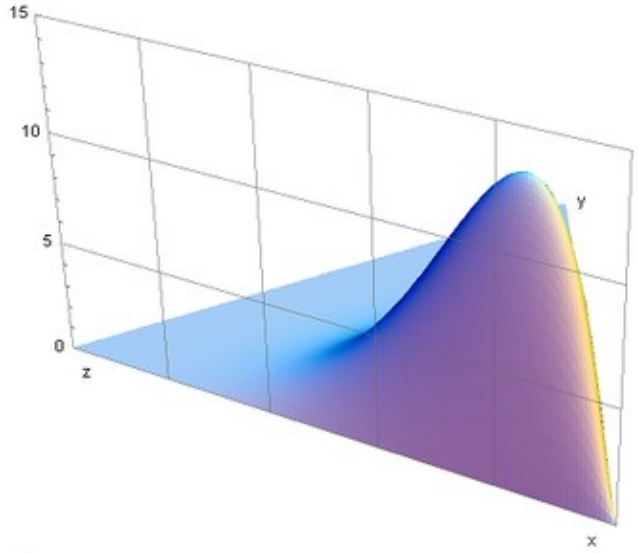
$$\alpha_i=\alpha P_{base}(x=i)$$

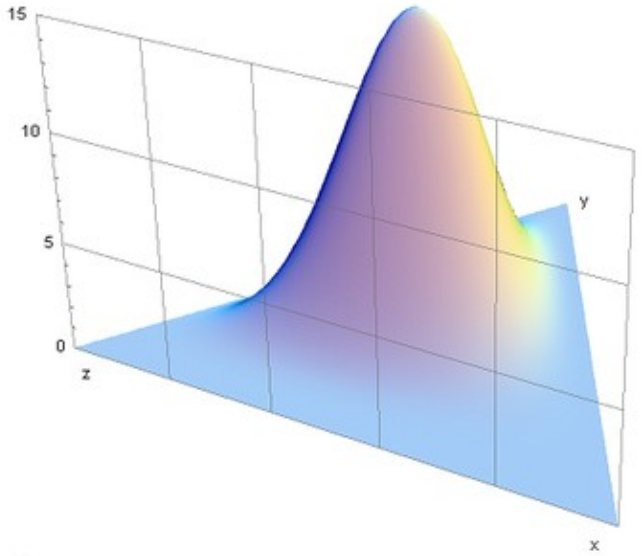
ディリクレ過程における書き方

  - 正則化定数は計算可：$Z=\frac{\prod_{i=1}^{n}\Gamma(\alpha P_{base}(x=i))}{\Gamma(\sum_{i=1}^{n}\alpha P_{base}(x=i))}$

確率密度の例

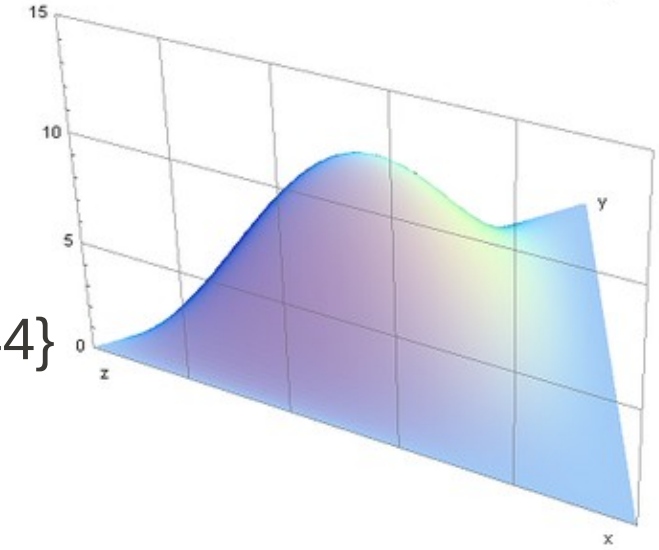
α = 10
P_{base} = {0.6,0.2,0.2}

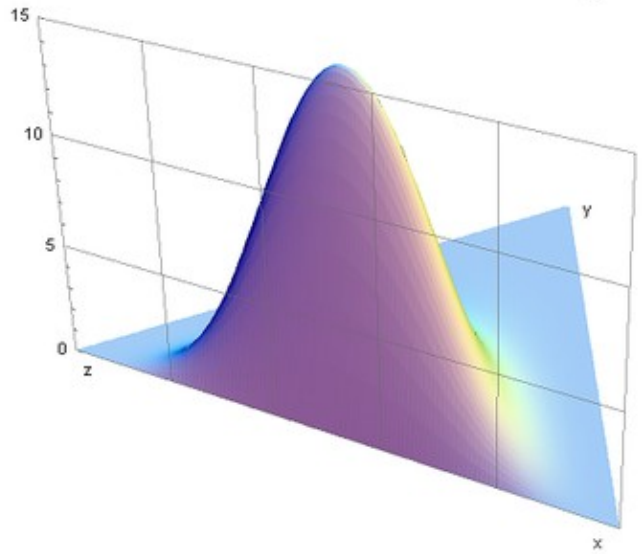




α = 15
P_{base} = {0.2,0.47,0.33}

α = 9
P_{base} = {0.22,0.33,0.44}





α = 14
P_{base} = {0.43,0.14,0.43}

図： Wikipedia

なぜ共役なのか？

- 尤度は多項分布から生成された事象の確率の積

データ： $x_1=1, x_2=5, x_3=2, x_4=5$

$$P(X|\theta)=p(x=1|\theta)p(x=5|\theta)p(x=2|\theta)p(x=5|\theta)=\theta_1\theta_5\theta_2\theta_5$$

- 頻度を利用し同一の事象を一緒にすると

$$c(x=i)=\{1,1,0,0,2\}$$

$$P(X|\theta)=\theta_1\theta_2\theta_5^2=\prod_{i=1}^{n}\theta_i^{c(x=i)}$$

- ディリクレ事前分布との積を取ると

$$\prod_{i=1}^{n}\theta_i^{c(x=i)} * \frac{1}{Z_{prior}}\prod_{i=1}^{n}\theta_i^{\alpha_i-1} \rightarrow \frac{1}{Z_{post}}\prod_{i=1}^{n}\theta_i^{c(x=i)+\alpha_i-1}$$

ディリクレ分布におけるθの期待値

- N=2 の場合

$$E[\theta_1]=\int_0^1 \theta_1 \frac{1}{Z}\theta_1^{\alpha_1-1}\theta_2^{\alpha_2-1}d\theta_1$$

$$=\frac{1}{Z}\int_0^1 \theta_1^{\alpha_1}(1-\theta_1)^{\alpha_2-1}d\theta_1$$

部分積分で

$$u=\theta_1^{\alpha_1} \qquad du=\alpha_1\theta_1^{\alpha_1-1}d\theta_1$$

$$dv=(1-\theta_1)^{\alpha_2-1}d\theta_1$$

$$v=-(1-\theta_1)^{\alpha_2}/\alpha_2$$

$$\int u\,dv=uv-\int v\,du$$

$$=\frac{1}{Z}[-\theta_1^{\alpha_1}(1-\theta_1)^{\alpha_2}/\alpha_2]_0^1-$$

$$\frac{1}{Z}\int_0^1 -(1-\theta_1)^{\alpha_2}/\alpha_2*\alpha_1\theta_1^{\alpha_1-1}d\theta_1$$

$$=0+\frac{\alpha_1}{\alpha_2}\frac{1}{Z}\int_0^1 \theta_1^{\alpha_1-1}(1-\theta_1)^{\alpha_2}d\theta_1$$

$$=\frac{\alpha_1}{\alpha_2}E[\theta_2]=\frac{\alpha_1}{\alpha_2}(1-E[\theta_1])$$

$$E[\theta_1]=\frac{\alpha_1}{\alpha_1+\alpha_2}$$

一般化すると

$$E[\theta_i] = \frac{\alpha_i}{\sum_{i=1}^{n} \alpha_i} = \frac{\alpha P_{base}(x=i)}{\alpha} = P_{base}(x=i)$$

- ディリクレ過程の事後確率に当てはめると

$$P(x=i) = \int_0^1 \theta_i \frac{1}{Z_{post}} \prod_{i=1}^{n} \theta_i^{c(x=i)+\alpha_i-1}$$

観測頻度

$$= \frac{c(x=i)+\alpha * P_{base}(x=i)}{c(\cdot)+\alpha}$$

基底測度

集中パラメータ

- 加算スムージングと同じ

ディリクレ過程の周辺確率

- 連鎖法則を使って、コーパス全体の周辺確率を計算

X = 1 2 1 3 1 α=1 $P_{base}(x=1,2,3,4) = .25$ $P(x_i) = \frac{c(x_i) + \alpha * P_{base}(x_i)}{c(\cdot) + \alpha}$

c = { 0, 0, 0, 0 }

$P(x_1 = 1) = \frac{0 + 1 * .25}{0 + 1} = .25$

c = { 1, 0, 0, 0 }

$P(x_2 = 2 | x_1) = \frac{0 + 1 * .25}{1 + 1} = .125$

c = { 1, 1, 0, 0 }

$P(x_3 = 1 | x_{1,2}) = \frac{1 + 1 * .25}{2 + 1} = .417$

c = { 2, 1, 0, 0 }

$P(x_4 = 3 | x_{1,2,3}) = \frac{0 + 1 * .25}{3 + 1} = .063$

c = { 2, 1, 1, 0 }

$P(x_5 = 1 | x_{1,2,3,4}) = \frac{2 + 1 * .25}{4 + 1} = .45$

全体の周辺確率
P(X) = .25*.125*.417*.063*.45

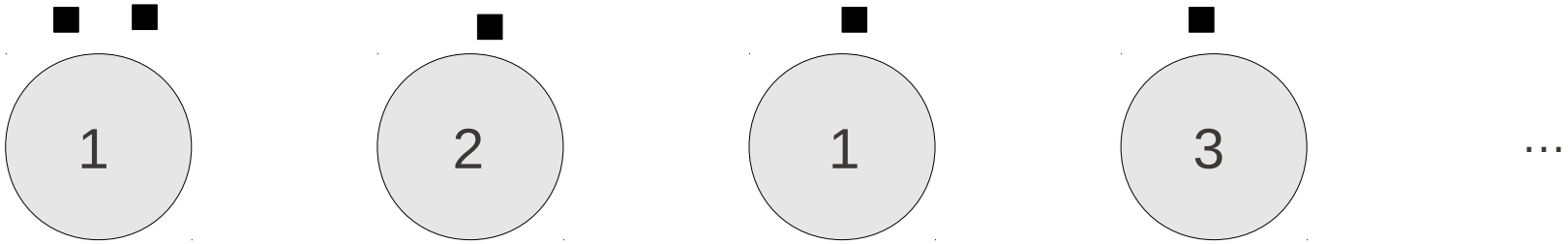
中華料理店過程

- ディリクレ過程などの表現法
- 中華料理店があり、無限のテーブルがある
- 客が１人ずつ入ってきて、以下の行動を取る：

$$p(\text{テーブル}\,i\,\text{に座る}) \propto c(i)$$
$$p(\text{新しいテーブルに座る}) \propto \alpha$$

- テーブルに初めての客が座る時に、そのテーブルに盛ってある料理を P_{base} の確率で選ぶ

X = 1 2 1 3 1 α=1 N=4



サンプリングの基礎

サンプリングの基本

- ある確率分布にしたがって、サンプルを生成する

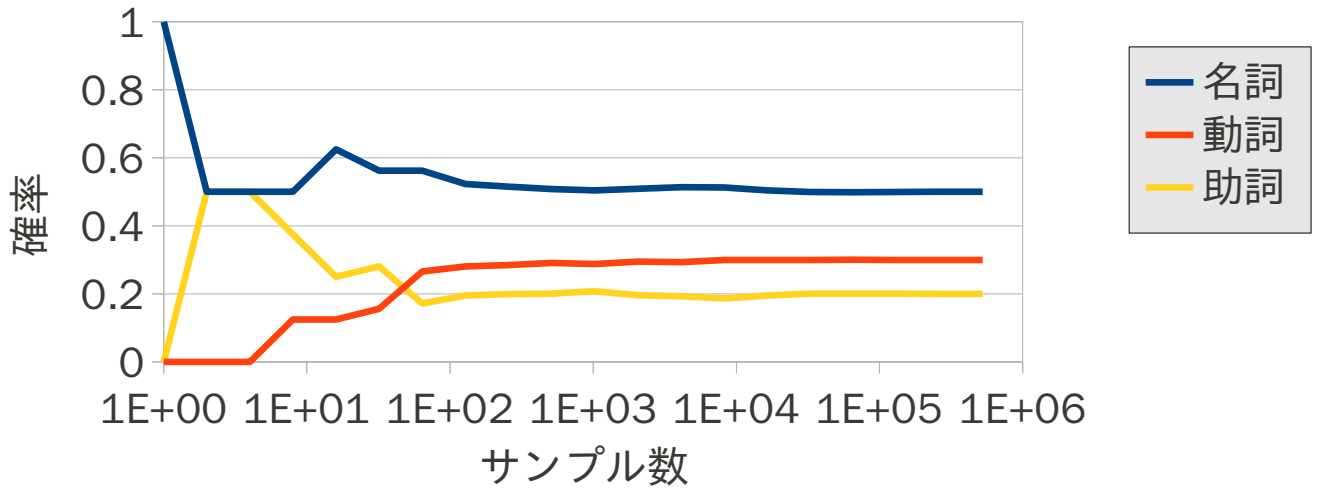
実際の分布： P(名詞)=0.5　P(動詞)=0.3　P(助詞)=0.2

サンプル： 動詞 動詞 助詞 名詞 名詞 助詞 名詞 動詞 動詞 名詞 …

- 各品詞のサンプル数を数えて、確率を近似

P(名詞)= 4/10 = 0.4, P(動詞)= 4/10 = 0.4, P(助詞) = 2/10 = 0.2

- サンプルの数を増やせば実際の分布に収束



具体的なアルゴリズム

```
SampleOne(probs[])
    z = sum(probs)
    remaining = rand(z)
    for each i in 1:probs.size
        remaining -= probs[i]
        if remaining <= 0
            return i
```

バグチェック！

確率の和を計算

[0,z) の一様分布に
従って乱数を生成

可能な確率をすべて考慮

現在の仮説の確率を引いて

ゼロより小さくなった
なら、サンプルの ID
を返す

ギブスサンプリング

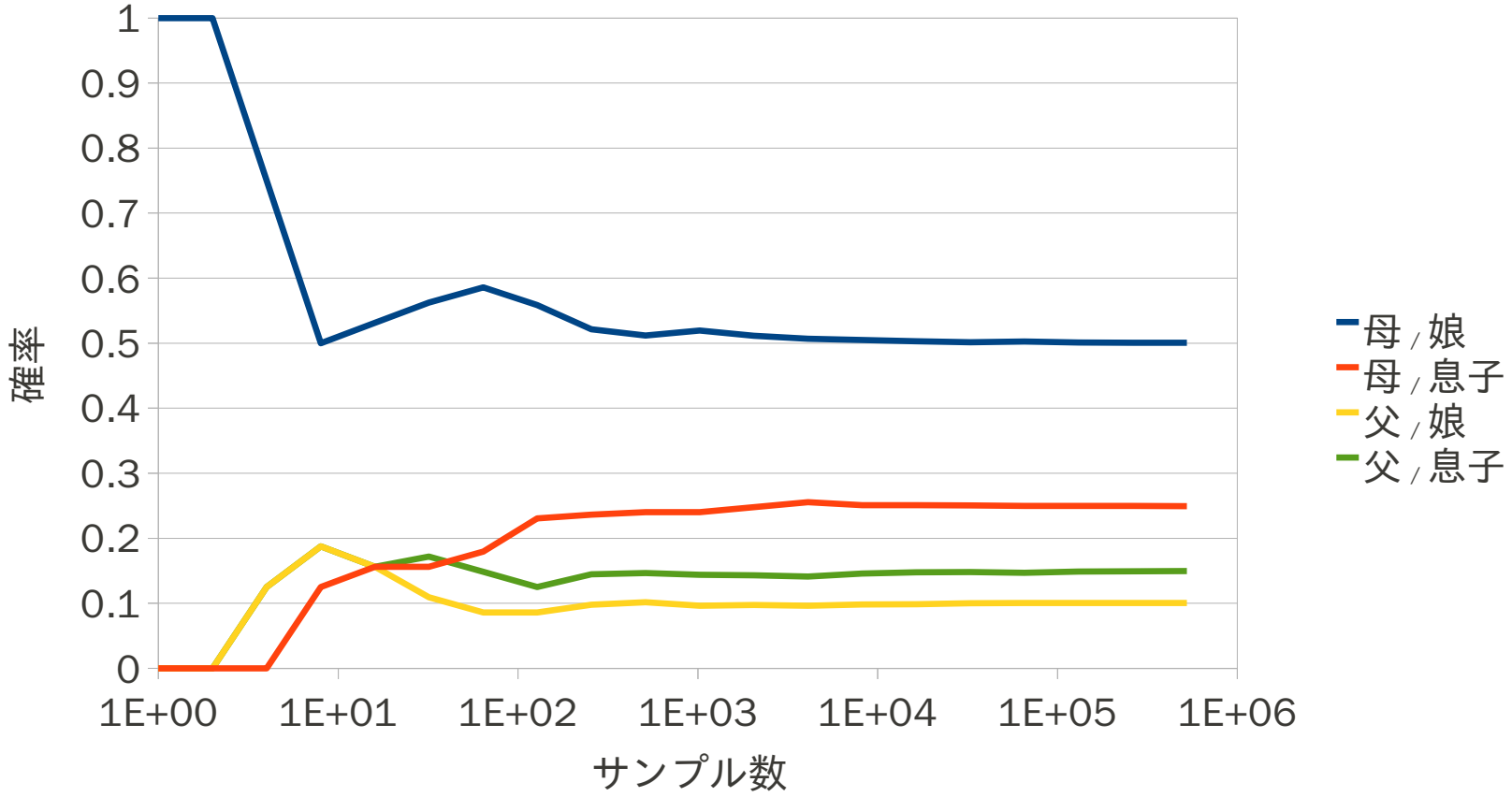
- ２つの変数 A,B があり P(A,B) をサンプリングしたい
  - が…、 P(A,B) 自体からサンプルできない
  - ただし、 P(A|B) と P(B|A) からサンプルできる
- ギブスサンプリングでは、変数を一個ずつサンプルできる
- 毎回：
  A を固定して、 P(B|A) から B をサンプルする
  B を固定して、 P(A|B) から A をサンプルする

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(母 | 娘) = 5/6 = 0.833　P(母 | 息子) = 5/8 = 0.625
  P(娘 | 母) = 2/3 = 0.667　P(娘 | 父) = 2/5 = 0.4
- 初期状態：母 / 娘
  A をサンプル：P(母 | 娘)=0.833，母を選んだ！
  B をサンプル：P(娘 | 母)=0.667，息子を選んだ！
  c(母 , 息子)++
  A をサンプル：P(母 | 息子)=0.625，母を選んだ！
  B をサンプル：P(娘 | 母)=0.667，娘を選んだ！
  c(母 , 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

サンプリングを用いた HMM の学習

教師なし学習の設定

- 観測された学習データ X
  - HMM の場合：自然言語のコーパス
- 潜在変数 Y
  - HMM の場合：コーパスの単語の品詞タグ
- 未観測のパラメータ θ
  - 主に確率

題材とするタスク：教師なし品詞推定

- 入力：単語列の集合 X

行 ごと に 処理 を 行 う

- 出力：クラスタ列のコーパス Y

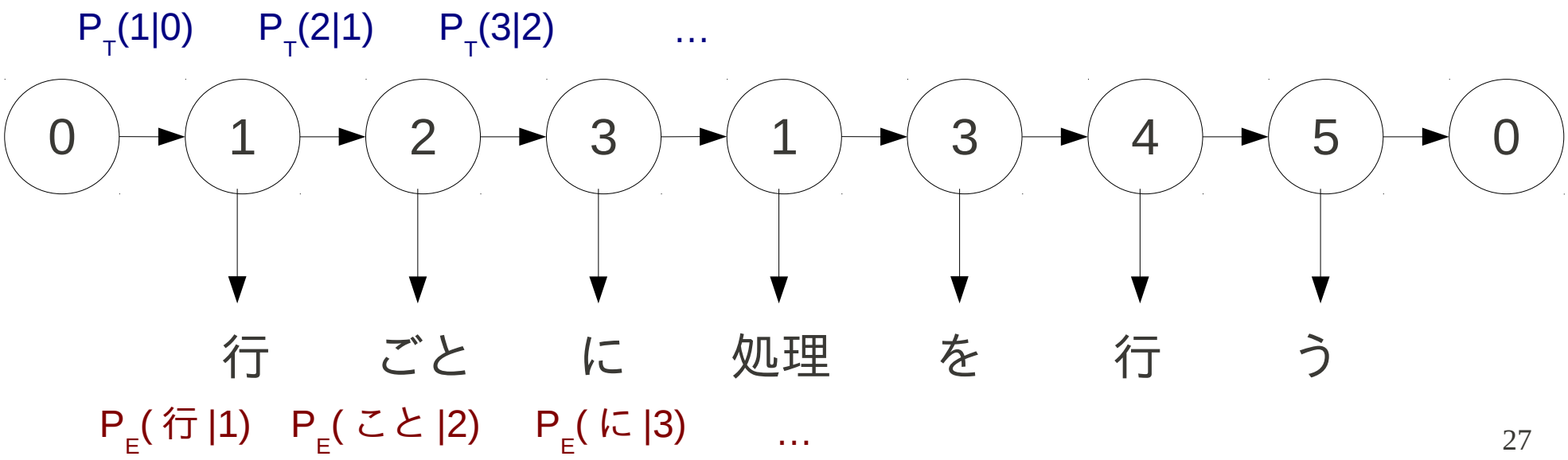
1 2 3 1 3 4 5

1→ 名詞 2→ 接尾辞 3→ 助詞 4→ 動詞 5→ 語尾

行	ごと	に	処理	を	行	う
名詞	接尾辞	助詞	名詞	品詞	動詞	語尾

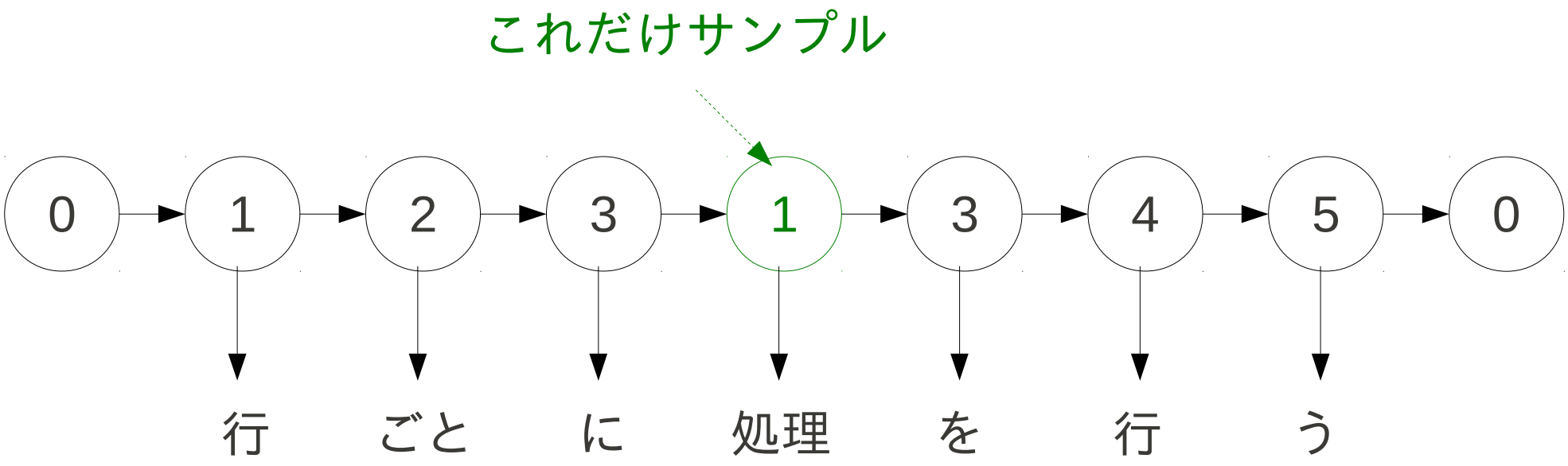
題材とするモデル：HMM

- 品詞 Y は隠れている状態
  - 状態の遷移確率は $P_T(y_i|y_{i-1})=\theta_{T,y_i,y_{i-1}}$
- 各状態から単語を生成する
  - 状態が与えられた場合の生成確率は $P_E(x_i|y_i)=\theta_{E,y_i,x_i}$

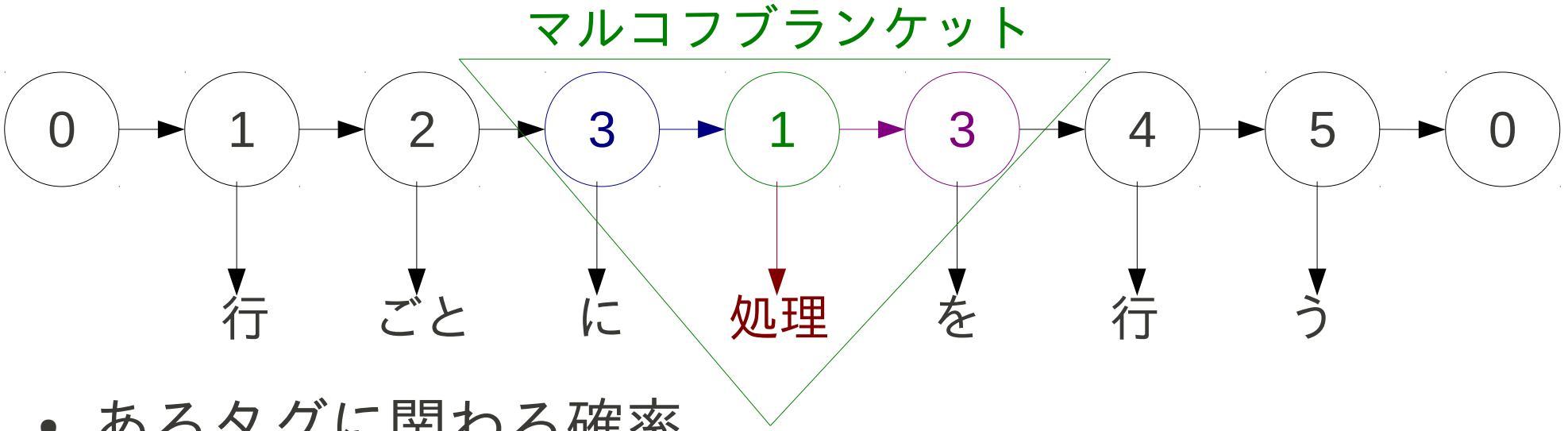


HMM におけるギブスサンプリング

- まず、Ｙをランダムに初期化
- Ｙを一個ずつギブスサンプリングでサンプルする



HMM におけるギブスサンプリング



- あるタグに関わる確率
  - 前のタグから遷移する確率： $P_T(y_i|y_{i-1})$
  - 次のタグへ遷移する確率： $P_T(y_{i+1}|y_i)$
  - タグが単語を生成する確率： $P_E(x_i|y_i)$
- この確率にしたがって各タグを順にサンプリングする
- 確率を決める周りの変数は「マルコフブランケット」

ディリクレ過程で HMM 確率を計算

- 遷移確率

$$P_T(y_i|y_{i-1}) = \frac{c(y_{i-1}y_i) + \alpha_T * P_{baseT}(y_i)}{c(y_{i-1}) + \alpha_T}$$

- 生成確率

$$P_E(x_i|y_i) = \frac{c(y_i, x_i) + \alpha_E * P_{baseE}(x_i)}{c(y_i) + \alpha_E}$$

1つのタグをサンプルするアルゴリズム

SampleTag(y_i)

$c(y_{i-1}\ y_i)$--; $c(y_i\ y_{i+1})$--; $c(y_i \rightarrow x_i)$--

いまのタグのカウントを削除する

for each *tag* in S (品詞の集合)

可能なタグの確率を計算

$p[tag]=P_E(tag|y_{i-1})*P_E(y_{i+1}|tag)*P_T(x_i|tag)$

y_i = **SampleOne**(p)

新しいタグを選ぶ

$c(y_{i-1}\ y_i)$++; $c(y_i\ y_{i+1})$++; $c(y_i \rightarrow x_i)$++

そのタグを追加する

全てのタグを サンプルするアルゴリズム

SampleCorpus()

initialize Y randomly	タグをランダムに初期化する
for N iterations	N イタレーション繰り返す
for each y_i in the corpus	全てのタグをサンプルする
SampleTag(y_i)	
save parameters	θ のサンプルを保存する
average parameters	θ のサンプルの平均を取る

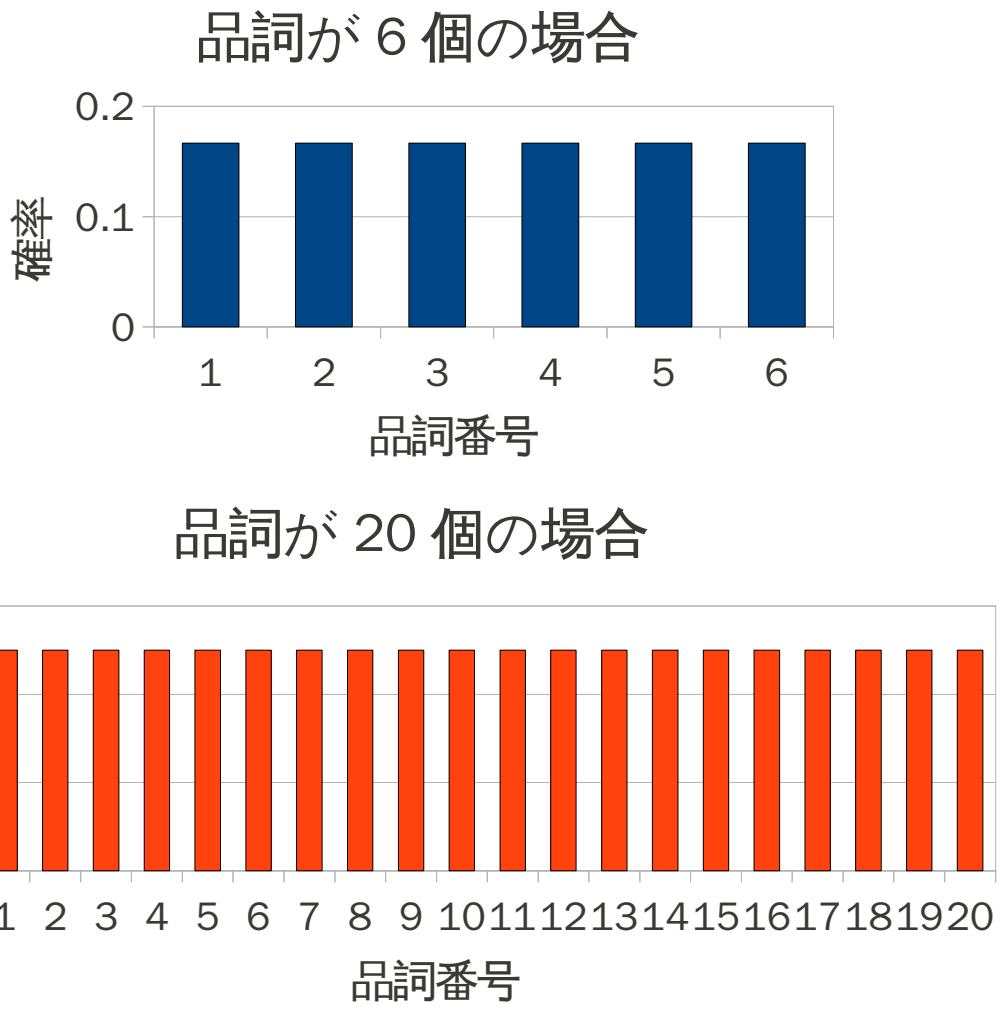
ハイパーパラメータの選び方

- α を選ぶ時に、 α の効果を考えなければならない
  - 小さい α(<0.1) を選べば、よりスパースな分布ができ上がる
    - 例えば、１つの単語がなるべく１つの品詞になるように生成確率 P_E の α_E を小さくする
  - 自然言語処理で多くの分布はスパースなので、小さい α が基本
- 実験で確認するのがベスト
- また、ハイパーパラメータ自体に事前分布をかけて自動的に調整する手法もある

有限 HMM から無限 HMM へ

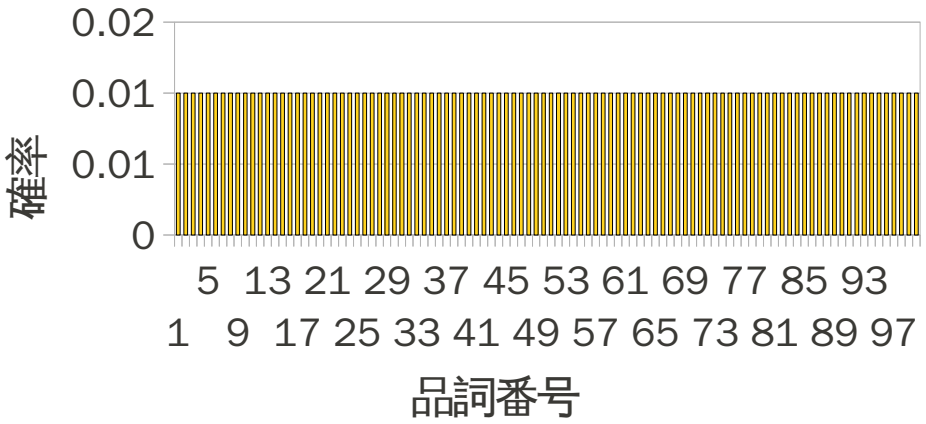
基底測度の次元数

- 一様分布の基底測度を利用すると

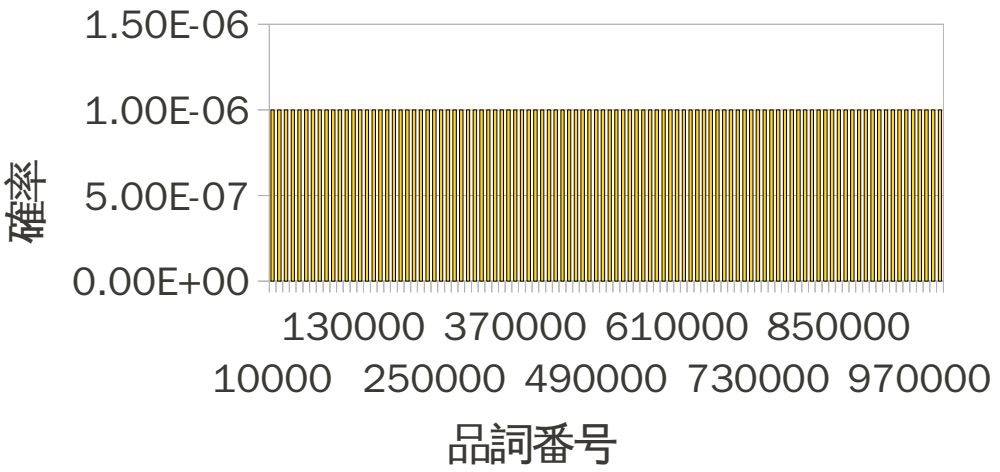


さらに伸ばすと

品詞が 100 個の場合



品詞が 100 万個の場合



- 品詞の数を極大まで増やしていくと
  - それぞれの品詞の P_{base} がゼロに近づく
  - しかし、 P_{base} 全体から品詞を生成する確率はそのまま

$$P(y_i|y_{i-1}) = \frac{c(y_{i-1}y_i) + \alpha * P_{base}(y_i)}{c(y_{i-1}) + \alpha}$$

N= 品詞の数

$$\lim_{N\to\infty} \sum_{i=1}^{N} \frac{1}{N} = 1$$

有限 HMM と無限 HMM

- 有限 HMM
  存在する品詞 y_i を (y_{i-1} の後に) 生成する確率

$$P(y_i|y_{i-1}) = \frac{c(y_{i-1}y_i) + \alpha * P_{base}(y_i)}{c(y_{i-1}) + \alpha}$$

- 無限 HMM
  存在する品詞 y_i を
  (y_{i-1} の後に) 生成する確率

$$P(y_i|y_{i-1}) = \frac{c(y_{i-1}y_i)}{c(y_{i-1}) + \alpha}$$

  新しい品詞を
  (y_{i-1} の後に) 生成する確率

$$P(y_i = new|y_{i-1}) = \frac{\alpha}{c(y_{i-1}) + \alpha}$$

例えば

- $c(y_{i-1}=1\ y_i=1)=1$ $c(y_{i-1}=1\ y_i=2)=1$ としよう

可能な品詞が 2 個 (y_1,y_2) の場合

$$P(y_i=1|y_{i-1}=1)=\frac{1+\alpha*1/2}{2+\alpha} \quad P(y_i=2|y_{i-1}=1)=\frac{1+\alpha*1/2}{2+\alpha}$$

$$P(y_i=1,2\text{以外}|y_{i-1}=1)=\frac{\alpha*0}{2+\alpha}$$

可能な品詞が 20 個 (y_1,y_{20}) の場合

$$P(y_i=1|y_{i-1}=1)=\frac{1+\alpha*1/20}{2+\alpha} \quad P(y_i=2|y_{i-1}=1)=\frac{1+\alpha*1/20}{2+\alpha}$$

$$P(y_i=1,2\text{以外}|y_{i-1}=1)=\frac{\alpha*18/20}{2+\alpha}$$

可能な品詞が∞個 (y_1,y_∞) の場合

$$P(y_i=1|y_{i-1}=1)=\frac{1+\alpha*1/\infty}{2+\alpha} \quad P(y_i=2|y_{i-1}=1)=\frac{1+\alpha*1/\infty}{2+\alpha}$$

$$P(y_i=1,2\text{以外}|y_{i-1}=1)=\frac{\alpha*1}{2+\alpha}$$

サンプリングアルゴリズム

SampleTag(y_i)

$c(y_{i-1}\ y_i)$--; $c(y_i\ y_{i+1})$--; $c(y_i \rightarrow x_i)$--

今のタグのカウントを削除する

for each *tag* in S (品詞の集合)

$p[tag]=P_E(tag|y_{i-1})*P_E(y_{i+1}|tag)*P_T(x_i|tag)$

存在するタグの確率を計算（ディリクレ過程の式で）

$p[|S|+1]=P_E(new|y_{i-1})*P_E(y_{i+1}|new)*P_T(x_i|new)$

新しいタグの確率を計算

y_i = **SampleOne**(p)

y_i の値を選ぶ

$c(y_{i-1}\ y_i)$++; $c(y_i\ y_{i+1})$++; $c(y_i \rightarrow x_i)$++

そのタグを追加する

一様でない基底測度

- 今までの展開は一様分布を前提としたが、一様でない基底測度も考えられる
- 代表的な一例：言語モデルの未知語モデル

$$P(\text{単語}) = \frac{c(\text{単語}) + \alpha * P_{base}(\text{単語})}{c(\text{単語}) + \alpha}$$

- 単語を文字に分解し、すべての可能な文字列に確率が与えられるようにする：

$$P_{base}(\text{単語}) = P_{len}(2) P_{char}(\text{単}) P_{char}(\text{語})$$

- 確率は一定ではないが、無限の集合に対して確率を与えている＝ノンパラメトリック

実装のいろいろ

- 普通は 0 頻度のクラスは残る→クラス数が上がる一方
  - 新しいクラスを作る時に 0 頻度のクラス番号を使い回す

$c(y_1)=5\ c(y_2)=0\ c(y_3)=1$ →
単純： $c(y_1)=5\ c(y_2)=0\ c(y_3)=1\ c(y_4)=1$
賢い： $c(y_1)=5\ c(y_2)=1\ c(y_3)=1$

  - $c(y_1)=0$ になると、 y_1 が復活する確率が 0 になる
- このモデルだと新しい品詞に弱い
  - 新しい品詞は 1 つの品詞の後にしか来ない
  - 階層モデル（後述）で解決

遷移確率 → $P_T(y_i|y_{i-1})=DP(\alpha,P_T(y_i))$

品詞確率 → $P_T(y_i)=DP(\alpha,P_{base}(y_i))$

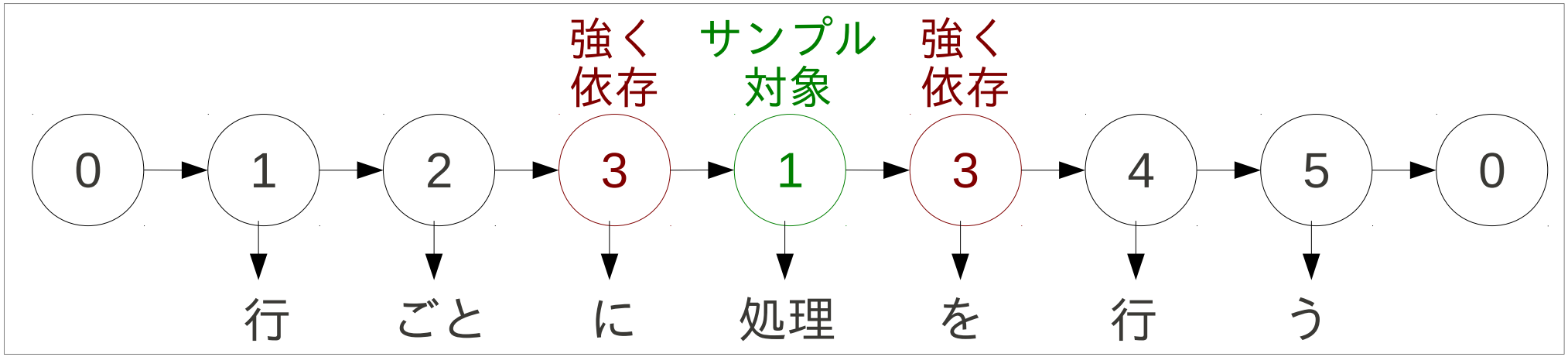
バグ探し

- 頻度の引き算や足し算を行う関数を作り、負になった場合に即時終了する
- プログラム終了時に全てのサンプルを削除し、全ての頻度がゼロになることを確認する
- 尤度は必ずしも単調上昇しないが、一旦上がってから単調減少すれば必ずバグが入っている
- 小さいデータで単体テストを作る
- 乱数生成器のシードを一定の値に設定する (srand)

近年の進展

サンプリング：ブロックサンプリング

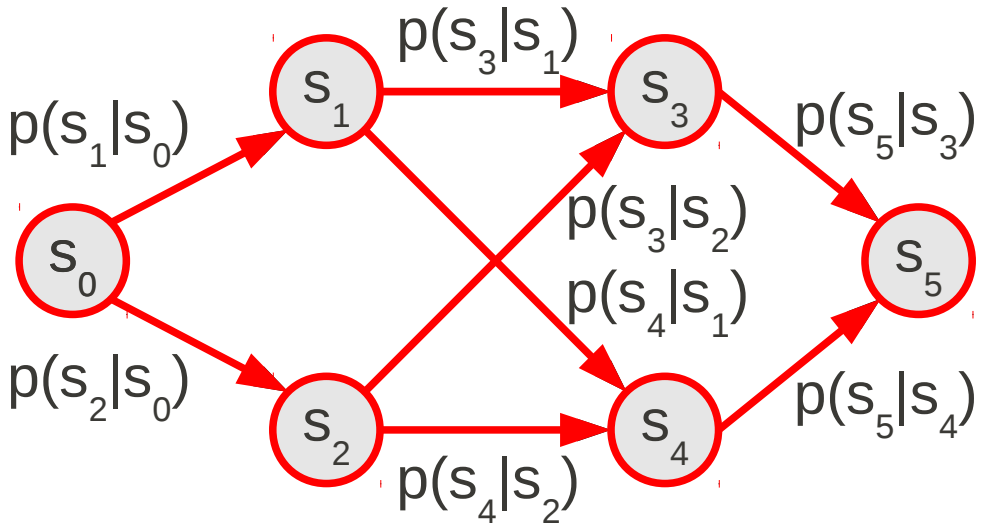
- 多くの場合、潜在変数は強い依存関係にある



- 例えば言語処理なら、文内の変数は強く依存
- ブロックサンプリングで、複数の潜在変数を同時にサンプリングする（依存関係も考慮）
- HMM なら forward filtering/backward sampling
  - PCFG なども可能

forward-filtering

- forward-filtering は前向き後ろ向きアルゴリズムの前向きステップと同等



forward filtering
前向き確率を順番に計算

$f(s_0) = 1$

$f(s_1) = p(s_1|s_0)*f(s_0)$

$f(s_2) = p(s_2|s_0)*f(s_0)$

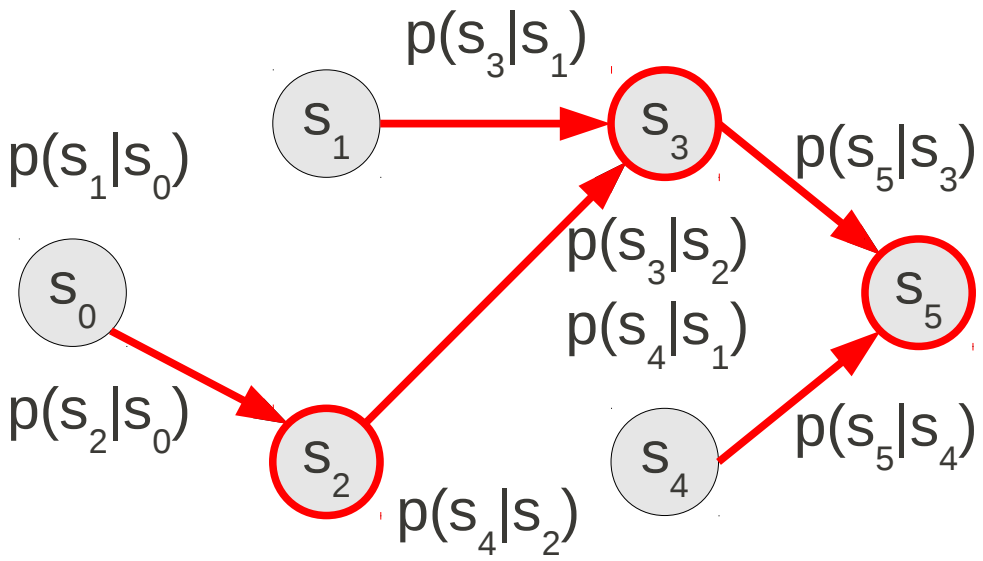
$f(s_3) = p(s_3|s_1)*f(s_1) + p(s_3|s_2)*f(s_2)$

$f(s_4) = p(s_4|s_1)*f(s_1) + p(s_4|s_2)*f(s_2)$

$f(s_5) = p(s_5|s_3)*f(s_3) + p(s_5|s_4)*f(s_4)$

backward-sampling

- backward-sampling は、受理状態から後ろ向きにエッジをサンプリングして行く



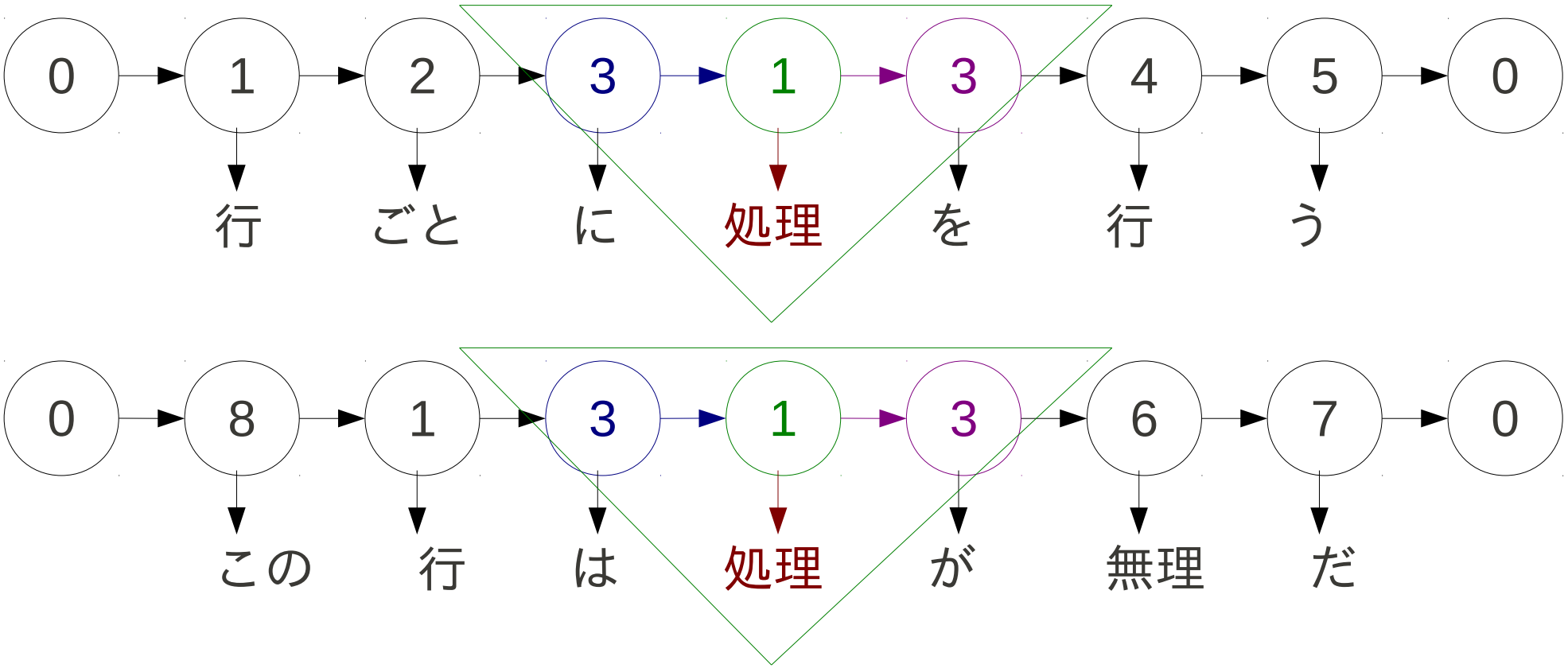
backward sampling
後ろからパスをサンプリング

$e(s_5 \rightarrow x)$
$p(x=s_3) \propto p(s_5|s_3)*f(s_3)$
$p(x=s_4) \propto p(s_5|s_4)*f(s_4)$

$e(s_3 \rightarrow x)$
$p(x=s_1) \propto p(s_3|s_1)*f(s_1)$
$p(x=s_2) \propto p(s_3|s_2)*f(s_2)$

サンプリング：タイプサンプリング

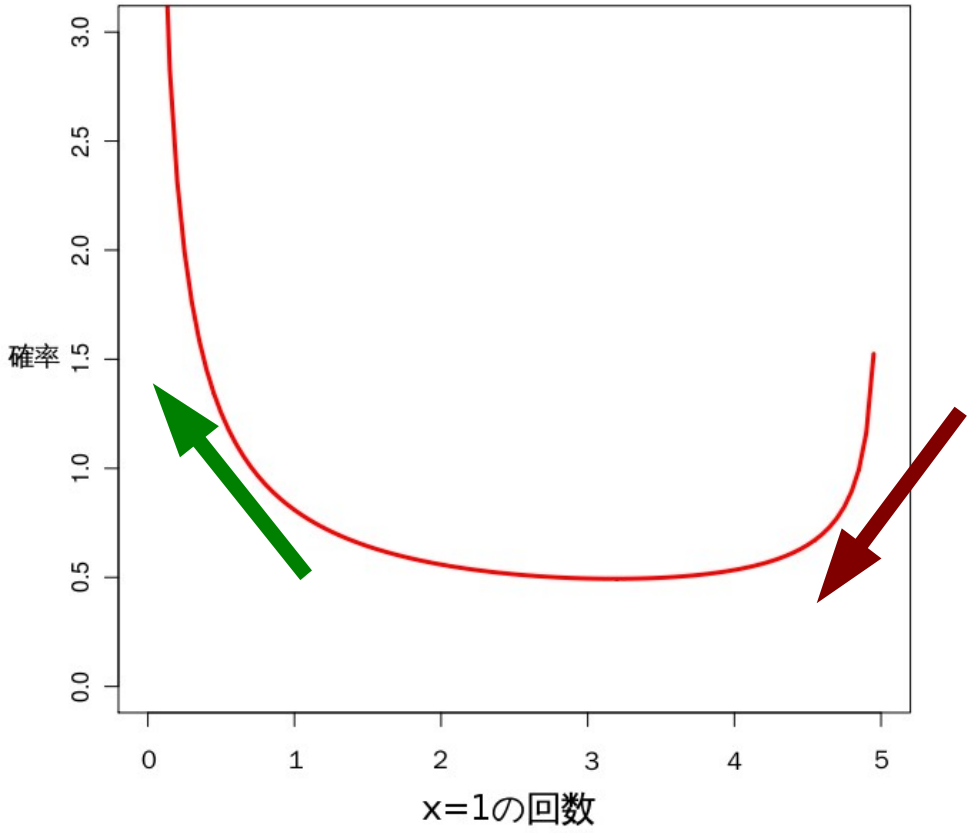
- 同じマルコフブランケットを持つ変数を同時にサンプリング



- 「3, 処理 ,3」の場合、 x=1 か x=2 かを考える

タイプサンプリング

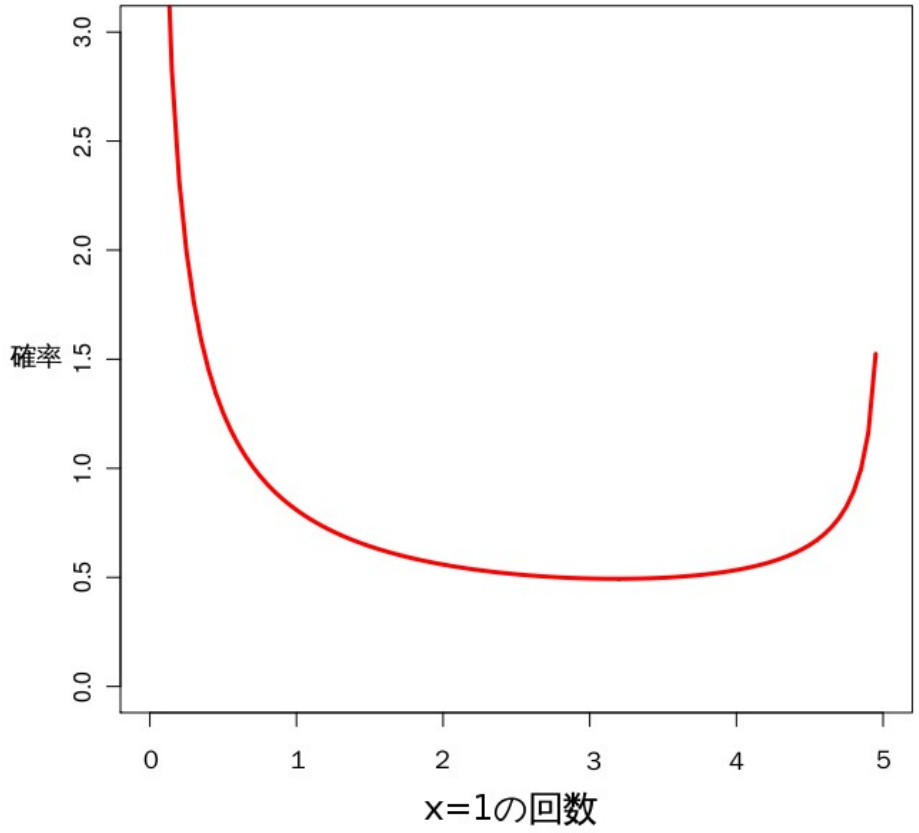
- ディリクレ過程などで、同じ状況のものに同じタグを振る傾向（rich-gets-richer 効果）がある
  - モデルとしては良い：スパースで簡潔なモデルを学習
  - 推論は難しい：これによって確率の「谷」ができる



- 現在の状況が右側
- より高い確率の領域に行く前に、低い確率の領域を通る
- サンプリングで脱出可能であるが、非常に時間がかかる

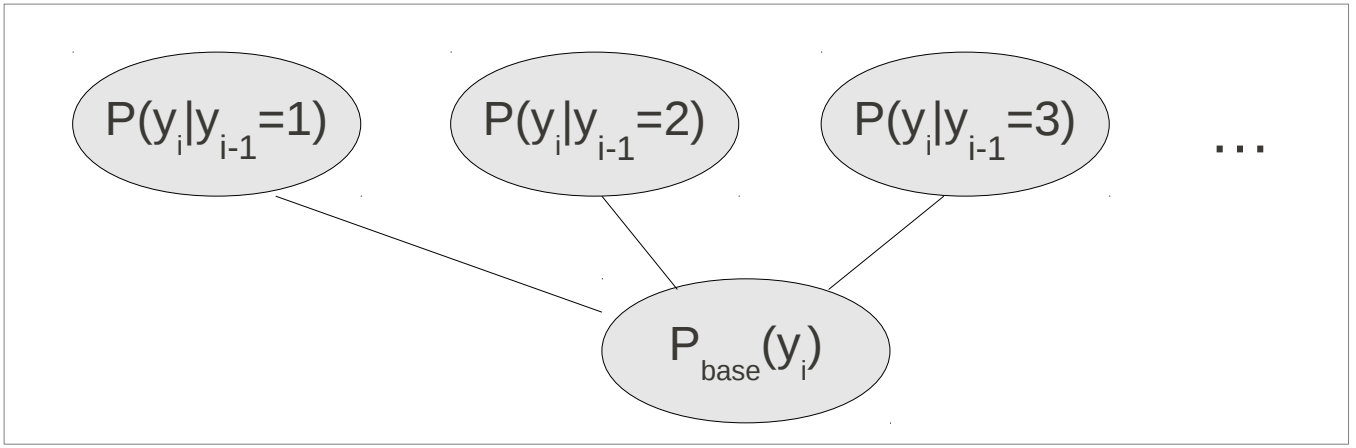
タイプサンプリング

- タイプをまとめて「x=1 は何個あるか」を解析的にサンプリングする
  - 「x=1 は 1 回」が選べれた
- マルコフブランケットは同じなので、確率は全て同等
  - 必要な個数をランダムに割り当てる
  - 5 個中、ランダムに 1 個だけ 1 にし、それ以外を 2 にする



モデル：階層的モデル

- 階層的ディリクレ過程でモデルによる多層化



遷移確率：

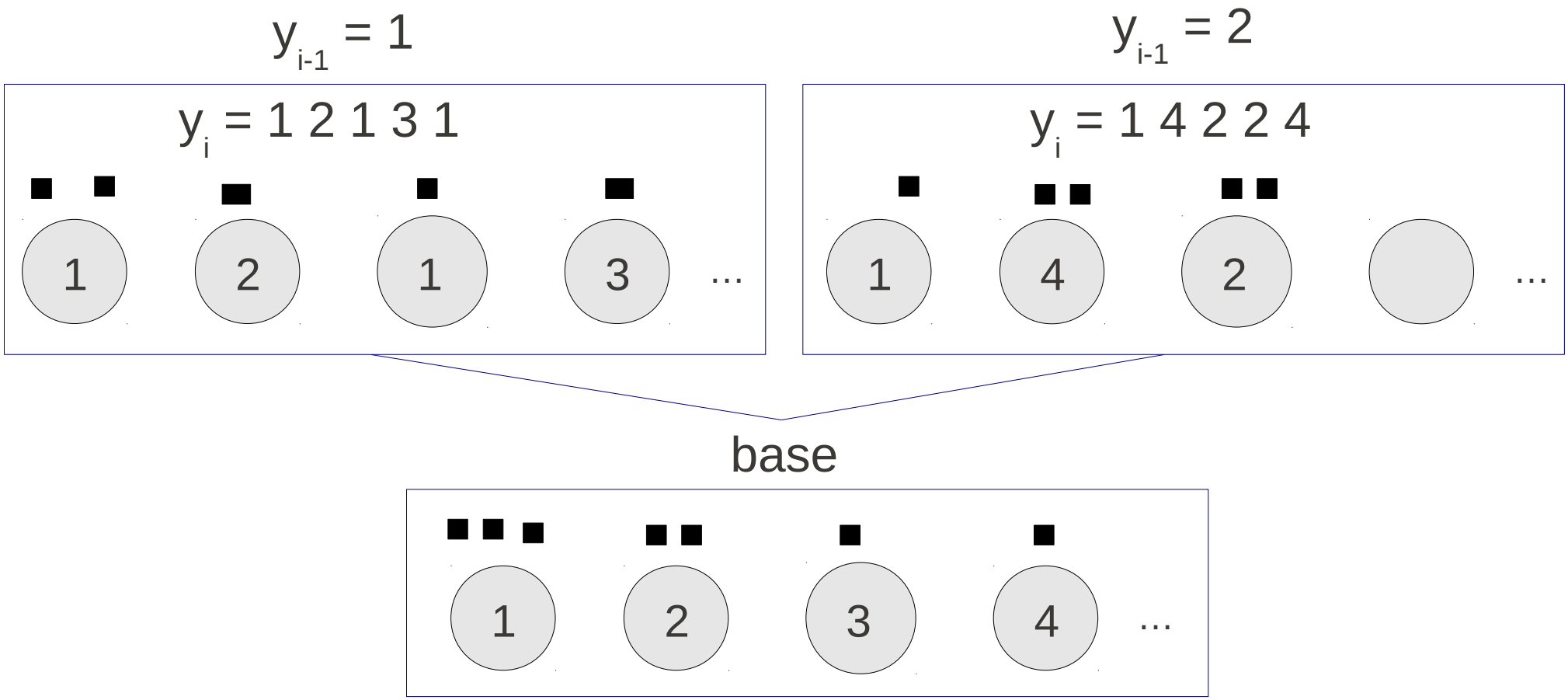
$$P(y_i|y_{i-1}) = \frac{c(y_{i-1}y_i) + \alpha * P_{base}(y_i)}{c(y_{i-1}) + \alpha}$$

共通の基底測度：

$$P_{base}(y_i) = \frac{c_{base}(y_i) + \alpha * 1/N}{c_{base}(\cdot) + \alpha}$$

c_{base} の数え方

- 中華料理店過程を利用

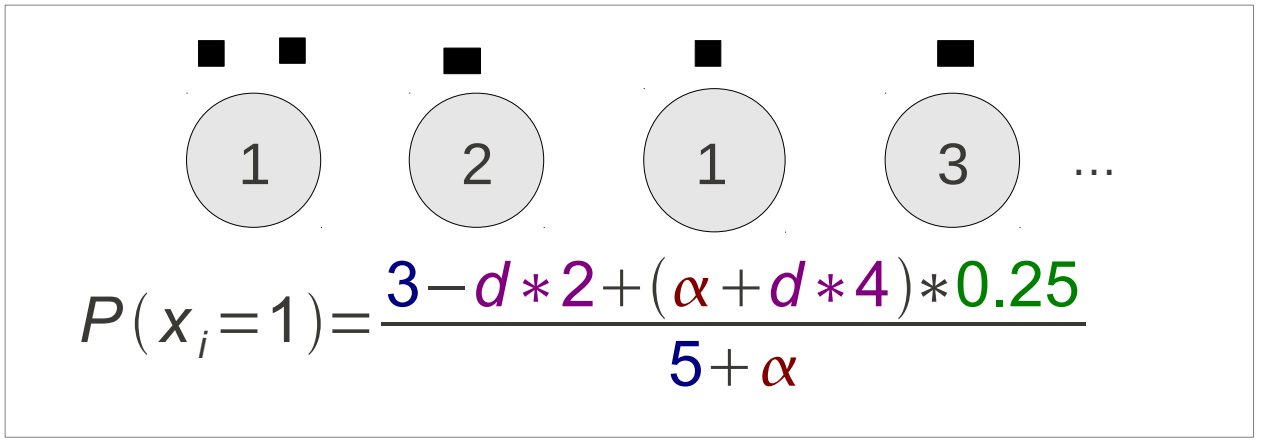


- 上の分布にテーブルが追加される場合のみ、下の分布に客を追加（Kneser-Ney の頻度の数え方と類似）

モデル： Pitman-Yor 過程

- ディリクレ過程の頻度に、テーブルごとの割引 d を追加

$$P(x_i)=\frac{c(x_i)-d*t(x_i)+(\alpha+d*t(\cdot))*P_{base}(x_i)}{c(\cdot)+\alpha}$$



$$P(x_i=1)=\frac{3-d*2+(\alpha+d*4)*0.25}{5+\alpha}$$

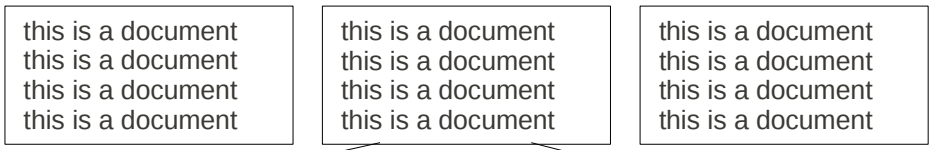
- Kneser-Ney などで利用する割引スムージング法と類似している形
- 言語処理等で現れるべき乗則 (power-law) 分布を表すのに適している

言語・音声処理における実用例

トピックモデル

- Latent Dirichlet Allocation (LDA) [Blei+ 03]

文章の
集合があり：



文章ごとにトピック
多項式分布を生成
（ディリクレ事前分布）

政治　芸能　スポ　経済　社会　科学
{ 0.4,　0.05,　0.3,　0.2,　0.01,　0.04}

各単語のトピックを
トピック分布から生成

1　1　4　3　3　3

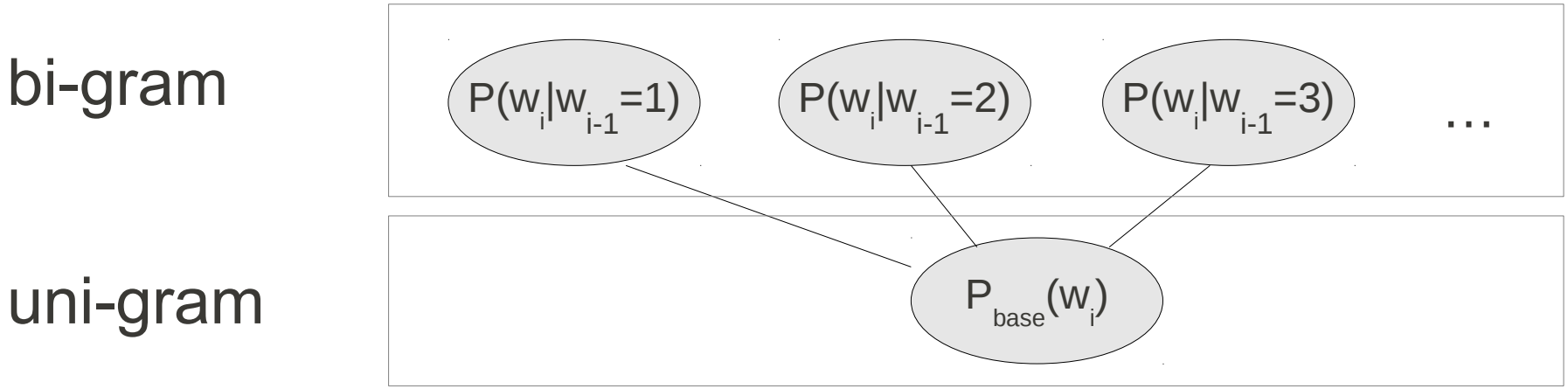
トピックの単語分布
から単語を生成

Bill Clinton buys the Detroit Tigers

- 階層モデルを使った無限トピックモデル [Teh+ 06]
- コンピュータービジョン、文書分類、音声認識の言語モデル等で利用（例：[Heidel+ 07]）

言語モデル

- 階層的 Pitman-Yor 言語モデル [Teh 06]



  - ディリクレ過程ではなく Pitman-Yor 過程を利用することで性能アップ
  - Kneser-Ney とほぼ同等の精度
- 音声認識で利用 [Huang&Renals 07]

教師なし単語分割

- 階層的ディリクレ過程を利用した単語分割 [Goldwater+ 09]

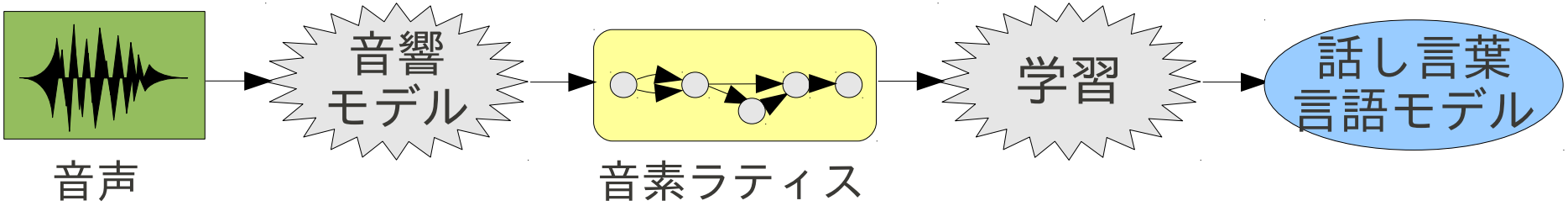
サンプリング

これ は 単語 で す	P(単語)
or	or
これ は 単 語 で す	P(単)P(語)

- ブロックサンプリングと Pitman-Yor 言語モデルへの展開 [Mochihashi+ 09]

音声からの言語モデル学習

- Pitman-Yor 言語モデルを使ってテキストではなく音声から直接言葉モデルと単語辞書を作る [Neubig+ 10]



- forward filtering-backward sampling を音素ラティスに対して行う
- 色々な応用が可能：
  - 資源の少ない言語の言語モデル構築
  - 話し言葉の口語的表現や発音変動に忠実なモデルが学習可能

様々な言語的情報の学習

- 品詞推定とその無限版 [Beal+ 02]
- 文脈自由文法 [Johnson+ 07] とその無限版 [Liang+ 07]
- 機械翻訳の単語やフレーズアライメント [DeNero+ 08, Blunsom+ 09, Neubig+ 11]
- 教師なし意味解析 [Poon+ 09] をノンパラメトリックベイズで実現した研究 [Titov+ 11]

参考文献

- M.J. Beal, Z. Ghahramani, and C.E. Rasmussen. 2002. The infinite hidden Markov model. Proceedings of the 16th Annual Conference on Neural Information Processing Systems, 1:577–584.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. Journal of Machine Learning Research, 3:993–1022.
- Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne. 2009. A Gibbs sampler for phrasal synchronous grammar induction. In Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics, pages 782–790.
- John DeNero, Alex Bouchard-Cˆot´e, and Dan Klein. 2008. Sampling alignment structure under a Bayesian translation model. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 314–323.
- Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2009. A Bayesian framework for word segmentation: Exploring the effects of context. Cognition, 112(1):21–54.
- A. Heidel, H. Chang, and L. Lee. 2007. Language model adaptation using latent Dirichlet allocation and an efficient topic inference algorithm. In Proceedings of the 8th Annual Conference of the International Speech Communication Association (InterSpeech).
- S. Huang and S. Renals. 2007. Hierarchical Pitman-Yor language models for ASR in meetings. In Proceedings of the 2007 IEEE Automatic Speech Recognition and Understanding Workshop, pages 124–129.
- Mark Johnson, Thomas Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov chain Monte Carlo. In Proceedings of the Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 139–146

参考文献

- P. Liang, S. Petrov, M. Jordan, and D. Klein. 2007. The infinite PCFG using hierarchical Dirichlet processes. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 688–697.
- Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. 2009. Bayesian unsupervised word segmentation with nested Pitman-Yor modeling. In Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics.
- Graham Neubig, Masato Mimura, Shinsuke Mori, and Tatsuya Kawahara. 2010. Learning a language model from continuous speech. In Proceedings of the 11th Annual Conference of the International Speech Communication Association (InterSpeech), Makuhari, Japan, 9.
- Graham Neubig, Taro Watanabe, Eiichiro Sumita, Shinsuke Mori, and Tatsuya Kawahara. 2011. An unsupervised model for joint phrase alignment and extraction. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, Oregon, USA, 6.
- H. Poon and P. Domingos. 2009. Unsupervised semantic parsing. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1–10.
- Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. 2006. Hierarchical dirichlet processes. Journal of the American Statistical Association, 101(476):1566–1581.
- Yee Whye Teh. 2006. A Bayesian interpretation of interpolated Kneser-Ney. Technical report, School of Computing, National Univ. of Singapore.
- Ivan Titov and Alexandre Klementiev. 2011. A Bayesian model for unsupervised semantic parsing. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics.