ノンパラメトリックベイズ入門
～ベイジアン HMM の実装まで～

Graham Neubig

２０１０年１１月８日

発表を聞いて分かること

- 実際にベイズ学習を使ったプログラムを実装するにはどうすれば良いか

○なぜベイズ学習が必要なのか

△ 確率過程やギブスサンプリングの基礎知識

○ ベイズ学習を使った教師なし品詞推定

× ベイズ学習の細かい理論

ベイズ法の基礎知識

雨について

- 外国へ留学して、４日が経ったところで毎日雨が降っている
- 毎日、雨が降る確率はどれぐらい？

観測データ　X = 雨、雨、雨、雨
パラメータ　θ = P(雨) = ???

最尤推定の答え

- 雨が降った日数を全体の日数で割る

$$P(\text{雨}) = \theta = 4/4 = 100\%$$

- 「今まで雨が毎日降っているので、これからも毎日降るでしょう！」
- そんなバカな…
- 実際のシステムでも問題になる（スムージングなどで対応）

最大事後確率（MAP）推定

- 日本で雨が降る確率は 25% なので、外国もそれぐらい（事前知識＝パラメータの事前分布 P(θ)）
- ただ、たくさん降っているので日本より多いかもしれない（観測データの尤度 P(X|θ)）
- これらを組み合わせると事後確率が得られる：

$$P(\theta|X) \propto P(X|\theta)P(\theta)$$

- 事後確率が最も高くなる θ を利用する：

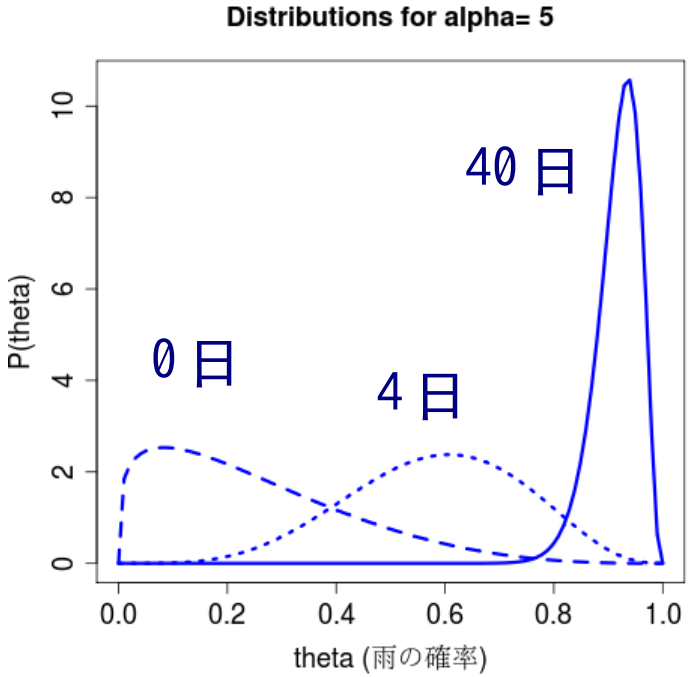
$$P(\text{雨}) = \hat{\theta} = \underset{\theta}{\operatorname{argmax}} P(X|\theta)P(\theta)$$

ベイズ推定

- MAP 推定は θ を一定に決める
  - 「明日雨が降る確率は絶対これだ！」
  - 砂漠でもたまたま 4 日間雨が降ることもある
- ベイズ学習は「θ の可能な値を全部考慮する」

$$P(\text{雨}) = \int_\theta P(X|\theta) P(\theta) d\theta$$

- 4 日降るより 40 日降った方が確信が持てる！



HMM モデル

今日の目的：教師なし品詞推定

- 入力：単語列 X

行 ごと に 処理 を 行 う

- 出力：品詞にマッチするクラスタ列 Y

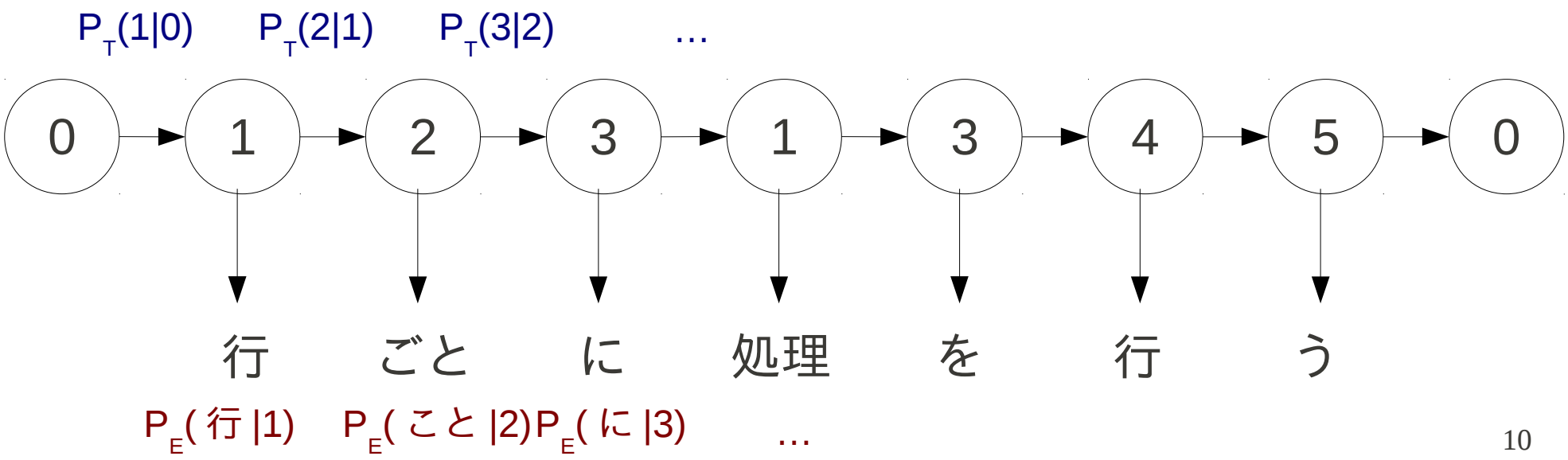
1 2 3 1 3 4 5

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

使用するモデル： HMM

- 品詞は隠れている状態
  - 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 $P_E(x_i|y_i)=\theta_{E,y_i,x_i}$



教師ありマルコフモデル学習

- コーパスを使って品詞遷移や単語 / 品詞組みを数える

行	ごと	に	処理	を	行	う
名詞	接尾辞	助詞	名詞	助詞	動詞	語尾

c(<s> 名詞) = 1 c(名詞 接尾辞) = 1 …

c(名詞→行) = 1 c(接尾辞→ごと) = 1 …

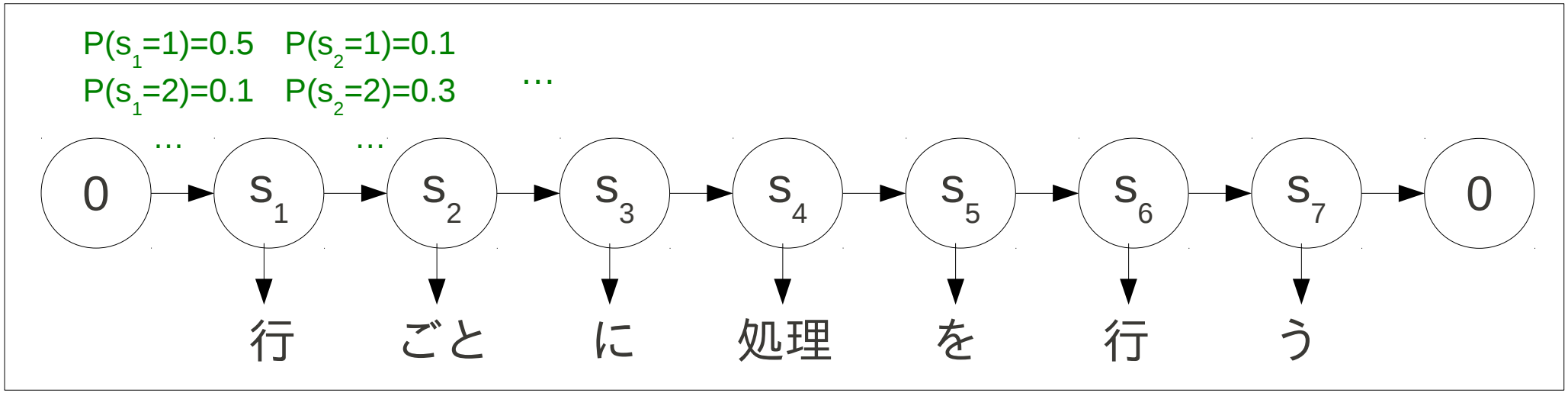
- 最尤推定で遷移確率、生成確率を計算

P_T(接尾辞 | 名詞) = c(名詞 接尾辞)/c(名詞) = 1/2 = 0.5

P_T(行 | 名詞) = c(名詞→行)/c(名詞) = 1/2 = 0.5

教師なし HMM 学習（最尤推定）

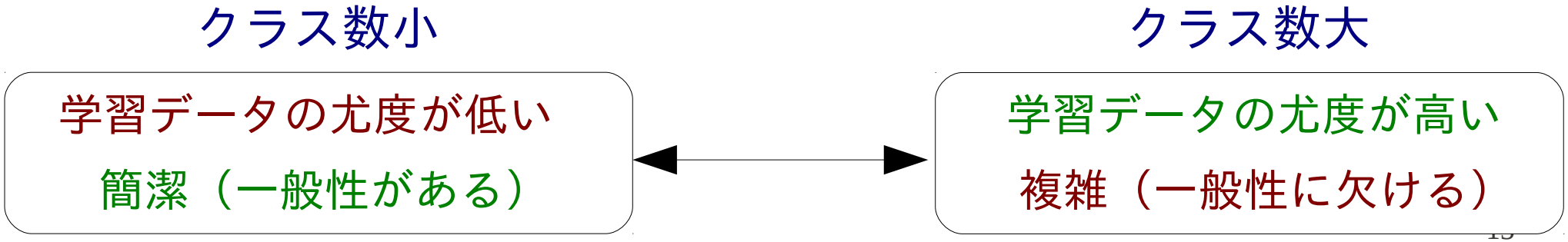
- モデルの遷移・生成確率をランダムに初期化
- EM アルゴリズムで学習
  - E-step: 現在のモデルで、各文に対して品詞列の期待値を計算する



  - M-step: E-step で計算された期待値を用いて、最尤推定でモデルの確率を更新

HMMの最尤推定の問題点

- 局所解に陥りやすく、一回なったら脱出できない
- 初期化に敏感
  - 例えば、全部の単語が同じクラスと初期化したら、ずっとそのまま
- クラス数を予め選ばなければならない
  - クラス数を自由にすると、全ての単語は個別のクラスに振られる
  - これは教師なし学習でよくある問題



ベイズ法の利点

- 最尤推定より極小解や初期値に頑健
  - 陥ることもあるが、イタレーションを十分繰り返したら脱出することができる
- クラス数を予め決めなくても良い
  - ノンパラメトリックベイズという手法を利用することで、モデルの大きさとバランスが取れる
- 実装は意外と簡単！

ベイジアン HMM

確率のスムージング

- 最尤推定で大きな問題になるのはゼロの確率

$c(y_{i-1}=N\ y_i=P) = 30$　$c(y_{i-1}=N\ y_i=V) = 20$

$y_i=$	P	V	N	ADJ	…
$P_{ML}(y_i \mid y_{i-1}=N)=$	0.6	0.4	0	0	

- よくある手法として確率がゼロにならないようにスムージングを行う（例えば：加算スムージング）

$$P_{smooth}(y_i \mid y_{i-1}) = \frac{c(y_{i-1} y_i) + \gamma}{c(y_{i-1}) + \gamma * S}$$

$c(y_i=N\ y_{i-1}=P)=30+1$　$c(y_i=N\ y_{i-1}=V)=20+1$

$c(y_i=N\ y_{i-1}=N)=0+1$　$(\gamma = 1)$

S= 品詞の数　　γ= 定数

$y_i=$	P	V	N	…
$P_{ML}(y_i \mid y_{i-1}=N)=$	0.52	0.36	0.02	

スムージングとベイズ

- 実は、この加算スムージングをベイズ学習の枠組みで「ディリクレ過程による事前分布」と等価である！

加算スムージング　　ディリクレ過程

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

- ディリクレ過程の理論は結構難しい…
  - ここでは実装に重要な部分だけを紹介
  - 参考：
    Teh, “Hierarchical Dirichlet Processes”
    Ghosh+ “Bayesian Nonparametrics”

ディリクレ過程の式

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

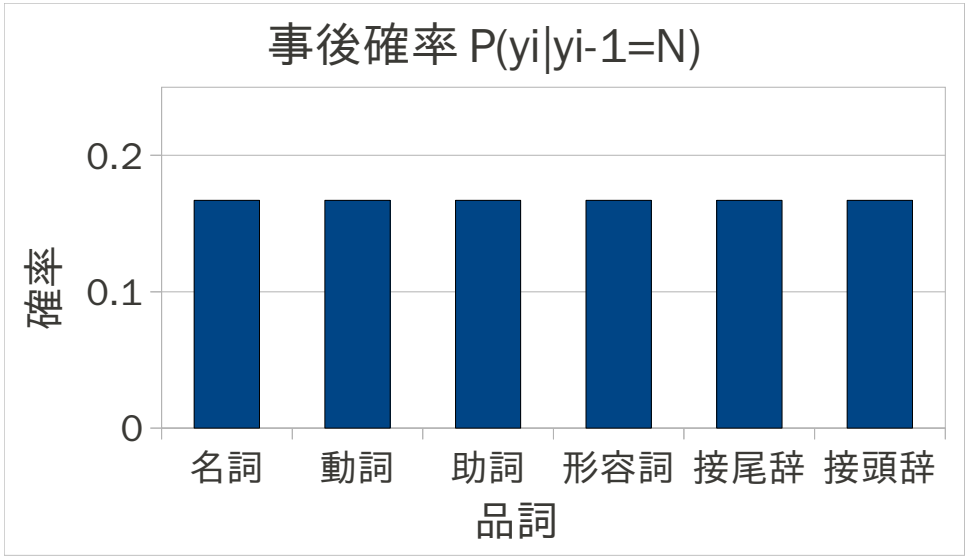
- $P_{base}(y_i)$ はディリクレ過程の基底測度（ここでは $P_{base}(y_i)=1/S$）
- α はディリクレ過程のハイパーパラメータ（ここでは、 $\alpha=\gamma*S$ ）
- $P(y_i|y_{i-1})$ が上記の式に従うことを以下の式で表すこともある

$$P(y_i|y_{i-1})\sim DP(\alpha, P_{base,y_{i-1}=N})$$

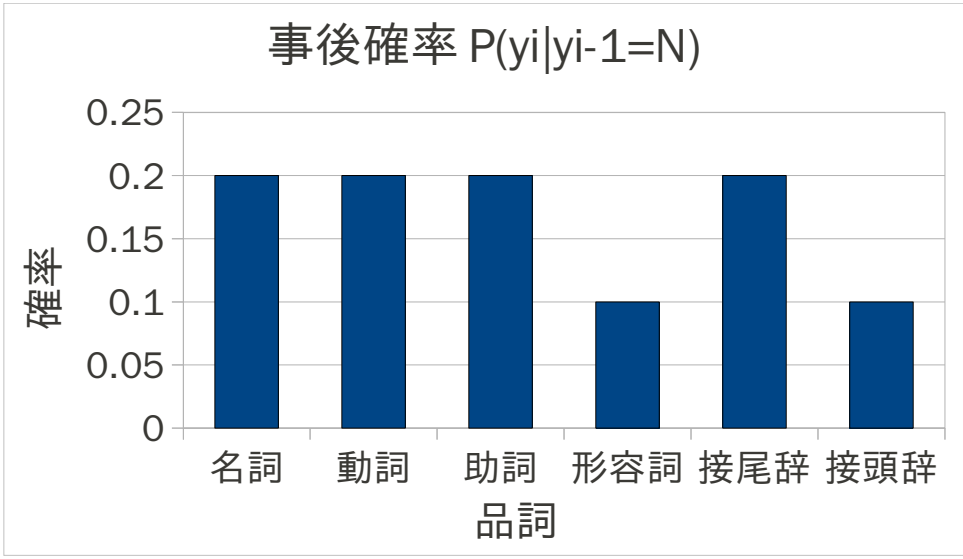
基底測度 (base measure) とは

- データを見る前の確率の期待値
- 例えば、 $P(y_i \mid y_{i-1}=N)$ の期待値は？

データを見る前に何も言えない！（一様分布）



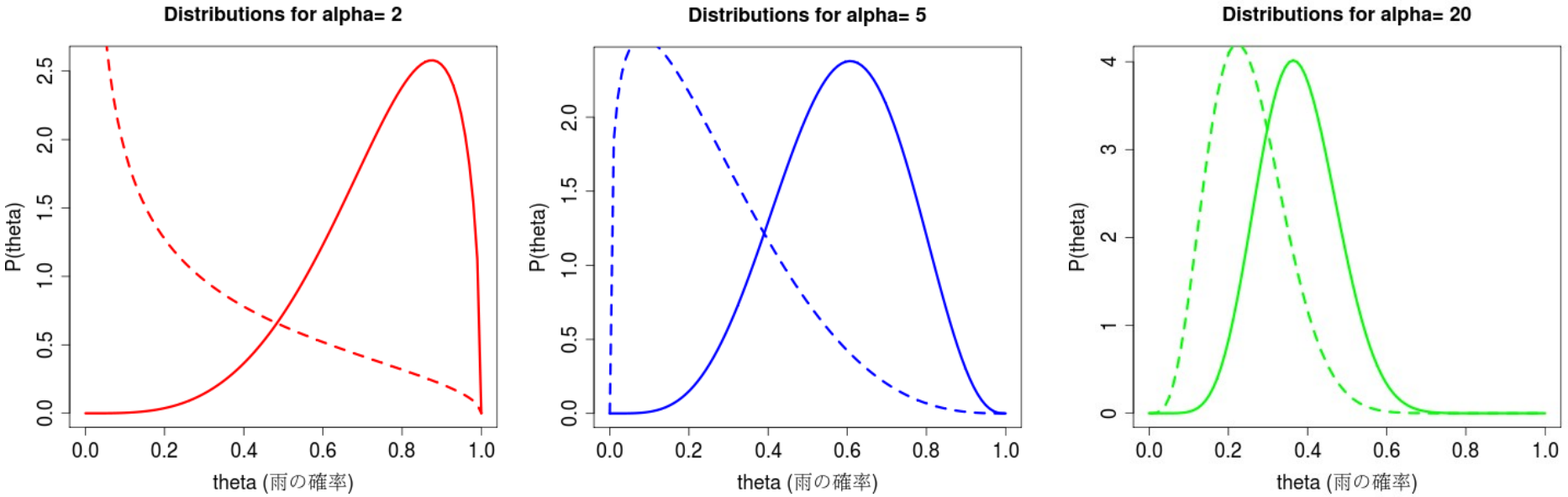
分からないけど、形容詞や接頭辞の確率は小さいはず！



- 基底測度を使って事前知識を組み込める

ハイパーパラメータ

- α を選ぶことで、事前分布の影響力が調整できる
  - 低いほどデータに影響される（0= 最尤推定、∞ = どれだけデータがあっても事前分布から動かない）



点線 = 事前確率　　実線 = 事後確率

ベイジアン HMM の構築

- 状態遷移確率と生成確率はディリクレ過程によるもの

$$P_E(x_i|y_i) \sim DP(\alpha_E, P_{base,E}) \qquad P_T(y_i|y_{i-1}) \sim DP(\alpha_T, P_{base,T})$$

α_E と α_T を適当に選ぶ（実験して一番いいものを利用）

$P_{base,T}$ は一様分布？

$P_{base,E}$ は単語ユニグラム確率？

Bayesian HMM の学習
～サンプリング～

ベイジアン HMM の確率推定

- パラメータの事後確率の分布を計算したい

$$P(\theta|X) \propto P(X|\theta)P(\theta)$$

- 実はディリクレ過程の場合は式にして解ける
  - 「変分ベイズ」という方法でこのようにしている
  - 積分を解くのがちょっと面倒
  - より複雑なモデルになるとできなくなる
- 代わりにサンプリングという方法を利用する
  - 変分ベイズより遅いが、実装が簡単＋より多くのモデルで使える

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

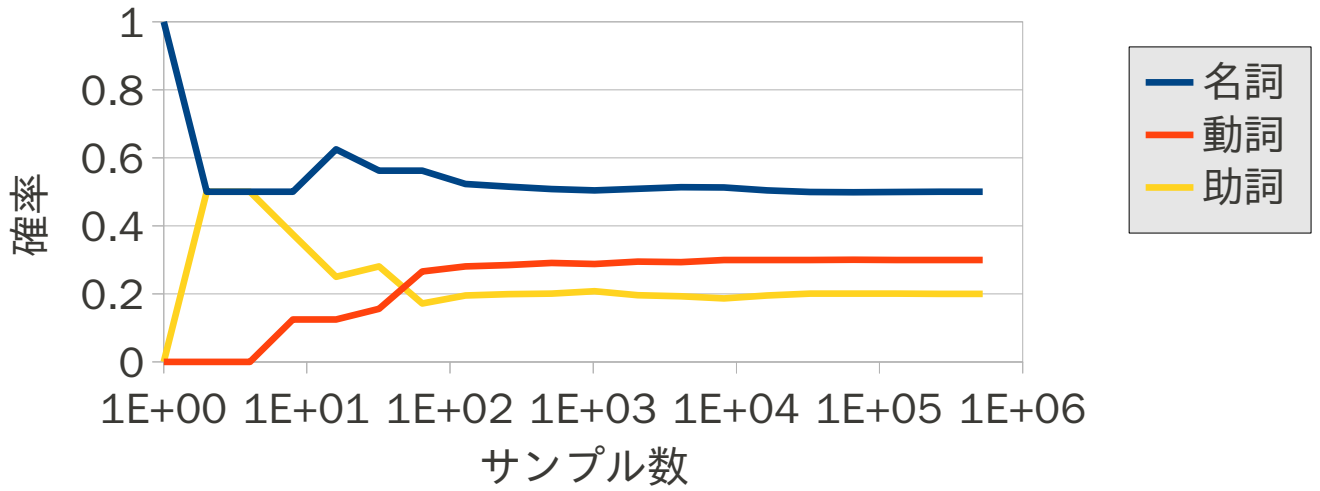
実際の分布： P(名詞)=0.5　P(動詞)=0.3　P(助詞)=0.2

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
```

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

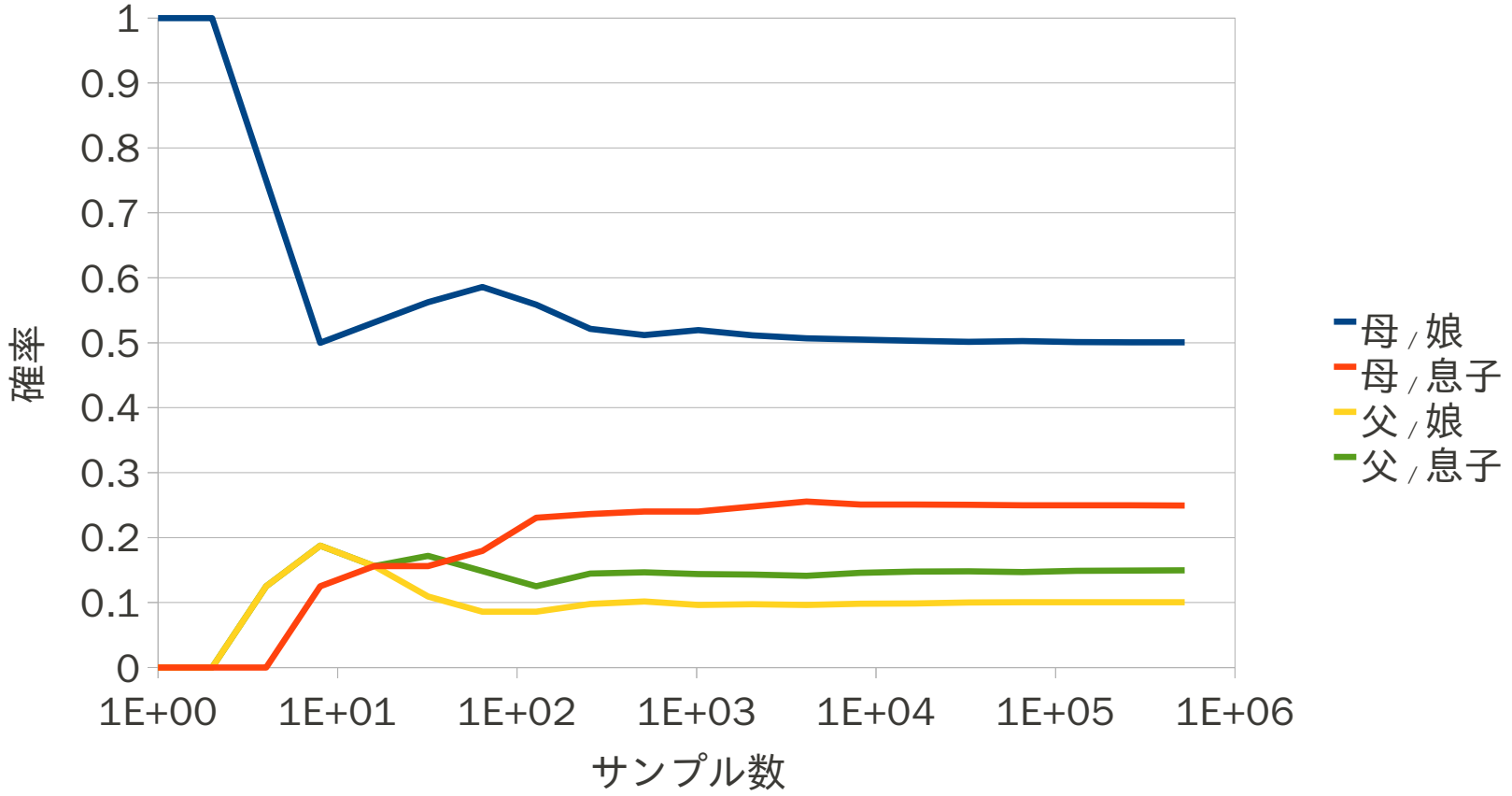
- ２つの変数 A,B があり P(A,B) をサンプリングしたい
  - が…、 P(A,B) 自体からサンプルできない
  - ただし、 P(A|B) と P(B|A) からサンプルできる
- ギブスサンプリングでは、変数を一個ずつサンプルできる
- 毎回：
  A を固定して、 P(B|A) から B をサンプルする
  B を固定して、 P(A|B) から A をサンプルする

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)++

…

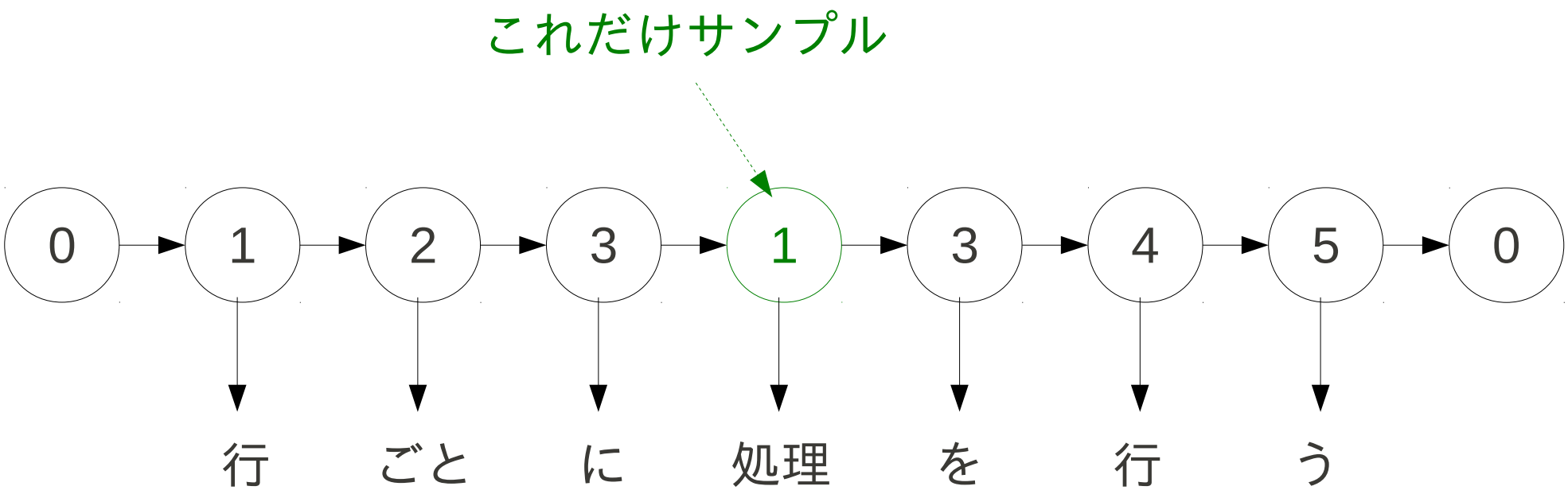
実際にやってみると



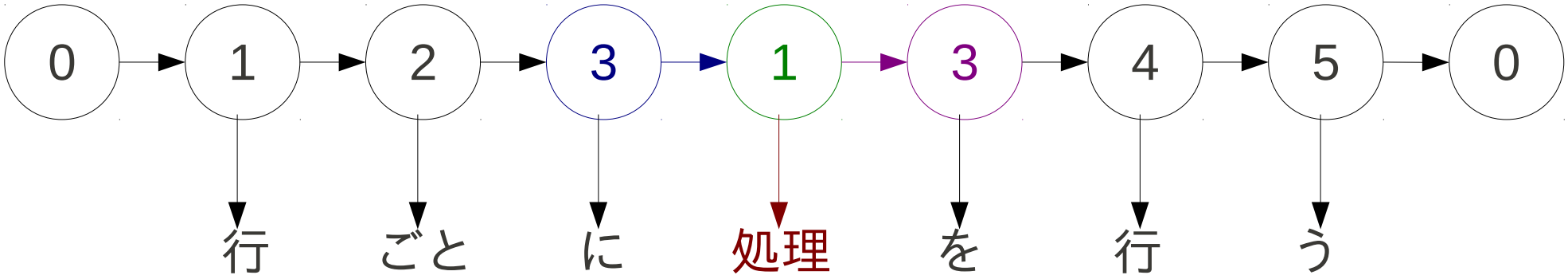
- 同時確率の式を手で解いてこの結果を確認できる

HMM におけるギブスサンプリング

- HMM は品詞列 Y は観測されていない変数とすると、一個ずつサンプルできるということになる



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ルしたら、教師なしで HMM が学習できる

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)

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

可能なタグの確率を計算（ディリクレ過程の式で , p. 17 ）

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

終わり！

- 意外と簡単！
  - 前向き後ろ向きアルゴリズム、 EM などしなくても良い
  - ただし、前向き後ろ向きアルゴリズムを利用するとさらに収束が早くなる（上級編）
- ただ、いくつかの難しいところがある
  - 品詞の数をどうやって選ぶ？
  - ハイパーパラメータ α はどうやって選ぶ？
  - 最尤推定による EM などと違って、尤度が単調増加する保証はない→デバッグがちょっと難しい

品詞の数

- 今回の課題で手で選ぶ（学習データにある品詞の数）
  - 課題のデータでは２１個
- ノンパラメトリックベイズの魅力の１つは予め品詞の数を選ばなくても良いこと
  - 次回の発表で扱う

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（上級編）

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
- 小さいデータでテストする
- 乱数生成器のシードを一定の値に設定する (srand)

課題と評価方法

課題

- ベイジアン HMM による品詞推定の教師なし学習を実装
- 学習データと評価スクリプトをお送りします

教師なし品詞推定の評価

- 教師なし学習はクラスに名前を付けることができない

教師あり：

行	ごと	に	処理	を	行	う
名詞	接尾辞	助詞	名詞	助詞	動詞	語尾

教師なし：

行	ごと	に	処理	を	行	う
1	2	3	1	3	4	5

- どうやって評価する？

品詞のマッピング

- ある教師なしクラス（1,2,3...,21) を、誤り率が最小となるようにマッピングする
- 例えば、クラス 0 が割り当てられた単語の実際の品詞が以下の通りであれば

  動詞 =3600 語尾 =2679 助動詞 =1581 名詞 =932 …
- 一番頻度が大きい「動詞」のクラスにマッピングする
- このような評価をしてくれるスクリプトもデータと一緒に送ります
  - 目安として、 50% を超えると動いていると言える

ノンパラメトリックとは？

- 「パラメータの数が決まっていない」
  - ここでは、パラメータの数は
    - 遷移確率： S*S
    - 生成確率： S*W
  - 実はノンパラメトリックじゃない！
- ただ、品詞の数を決めなければノンパラメトリック
  - 今日の課題が実装できたら後一歩
  - 教師なし単語分割も
- 次回で触れる予定