

---

# 無限次元離散分布と 無限木構造隠れMarkovモデル

持橋大地

統計数理研究所 数理・推論研究系

*daichi@ism.ac.jp*

統計サマーセミナー2016  
2016-8-10 (Wed)

# 統計的自然言語処理

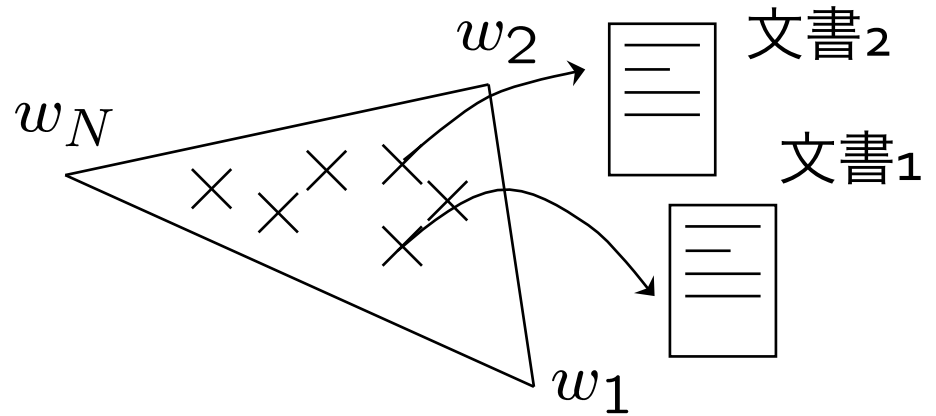
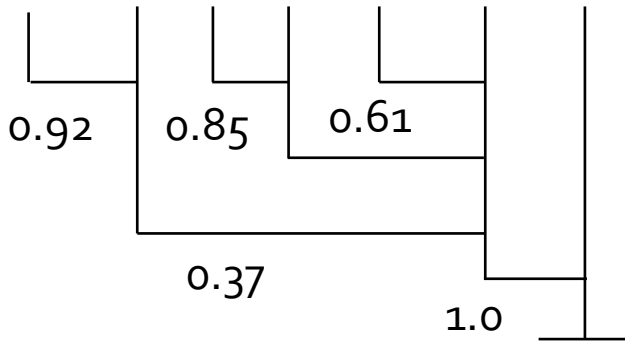
---

- 人間の言葉 (日本語・英語・中国語・サンスクリット…)を計算機で扱う分野
  - 「計算言語学」ともいわれる
- 電子計算機 (ENIAC)の誕生とほぼ同時に発生
  - ～1960年代 : (初歩的な)確率モデル
  - 1970～80年代 : 論理・文法によるモデル
  - 1990年代～ : 統計モデル (統計的機械学習)

# 統計的自然言語処理 (2)

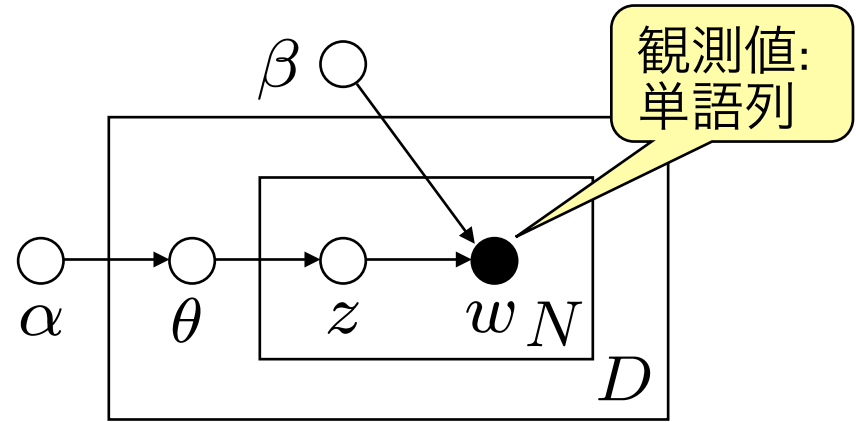
- 構文解析, 文書モデル, 評判分析, 古文書解読, ...

彼女は花を買った。



$$p(t|\mathbf{x}, \Lambda) = \frac{\exp(\sum_i \lambda_i f_i(\mathbf{x}, t))}{\sum_{\mathbf{x}} \exp(\sum_i \lambda_i f_i(\mathbf{x}, t))}$$

ある単語xの品詞  
が形容詞である確率

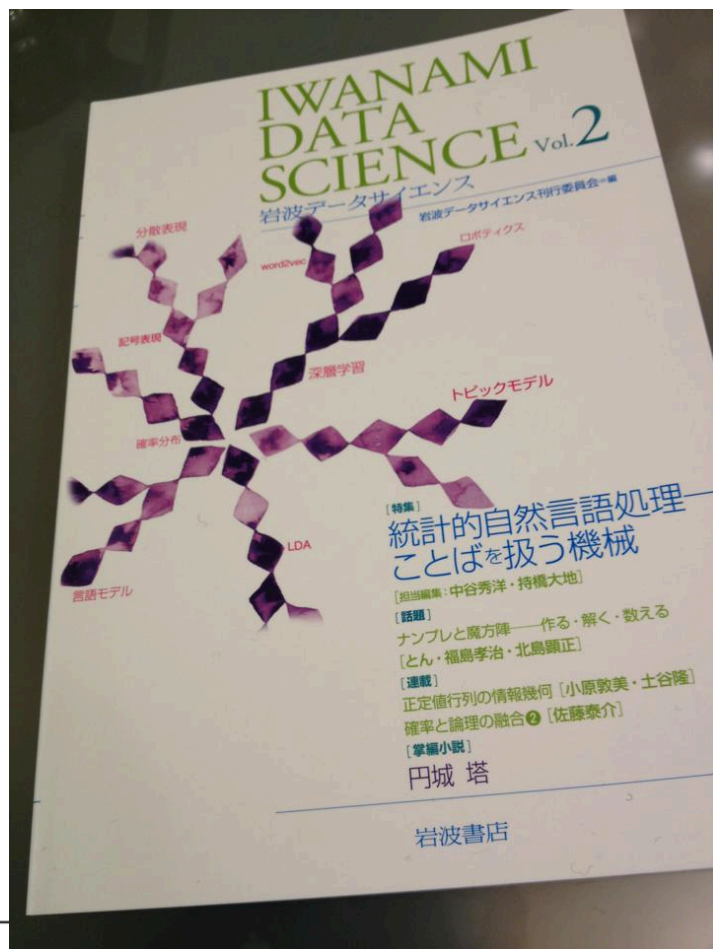


# ちなみに

- 岩波データサイエンス2：「統計的自然言語処理  
—ことばを扱う機械」

編集：持橋 (統数研)・

中谷 (サイボウズラボ)



- 「統計数理」2016年12月号  
も自然言語処理の特集です

# 統計的自然言語処理の特徴

- 観測値が離散・超高次元の時系列  
“国連 安保理 は 経済制裁 を 実行 した”  
↓  
“45701 14332 46 9734 7 2077 672 55 21”
- データ量が膨大
  - 数万～数百万～数億文の学習データ
  - 計算はR/MATLAB等ではほぼ不可能
    - C++の最適化されたコードでも数時間～数日の計算
    - 億単位の学習テキストの場合、数週間計算する場合も

# 統計的自然言語処理の特徴 (2)

- 観測値が離散・超高次元の時系列



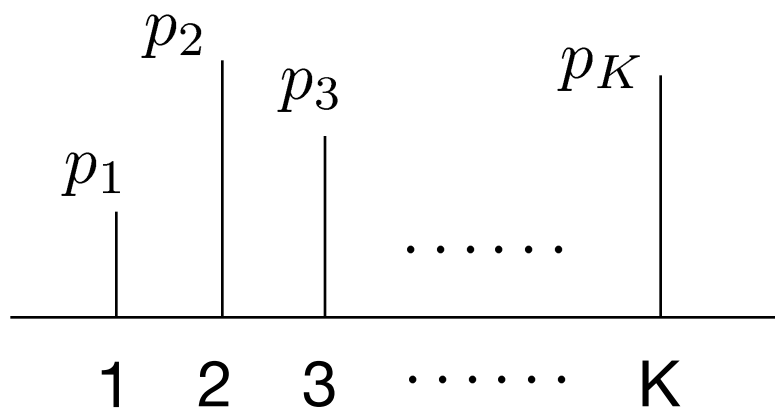
本当は、無限次元

- 可能な単語の種類は無限にある
- “キュラソ星人” “時雨P” “升” “水素水” “今津線”...
- 可能なカテゴリの数も無限
  - 動詞、名詞、名詞-鉄道-阪急、動詞-他動詞-抽象、...



- 無限次元離散分布を統計的にどうやって扱うか?

# 準備: 多項分布 (離散分布)



$$\sum_{k=1}^K p_k = 1,$$

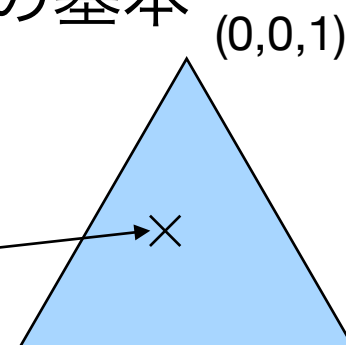
$$\forall k, p_k \geq 0$$

- K種類のアイトムのどれかが出る確率分布

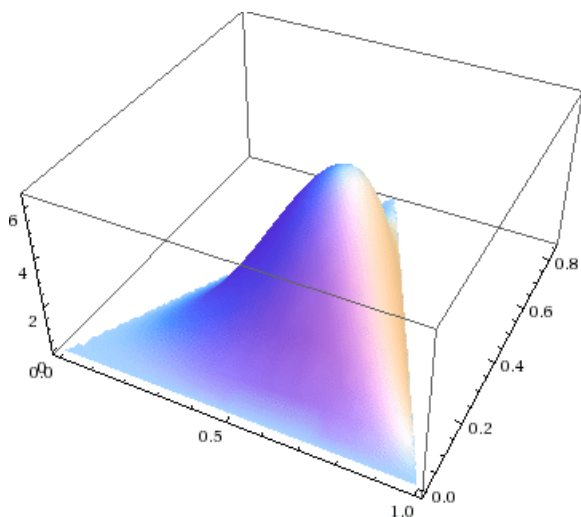
- 離散データの統計モデルの基本中の基本

- $\mathbf{p}$  は  $(K-1)$ 次元の単体(Simplex)の内部に存在

$$\mathbf{p} = (p_1, p_2, \dots, p_K)$$



# ディリクレ分布



$$\text{Dir}(\mathbf{p}|\boldsymbol{\alpha}) \propto \prod_{k=1}^K p_k^{\alpha_k - 1}$$

パラメータ:

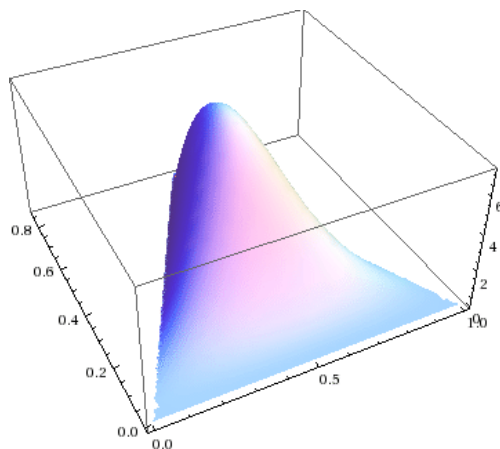
$$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)$$

- ランダムな多項分布を生成する確率分布
- $\alpha_k \equiv 1$  のとき、単体上でUniformな分布
- 「期待値」 :  $\bar{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_K) / \sum_k \alpha_k$
- 「分散」 :  $\alpha = \sum_k \alpha_k$

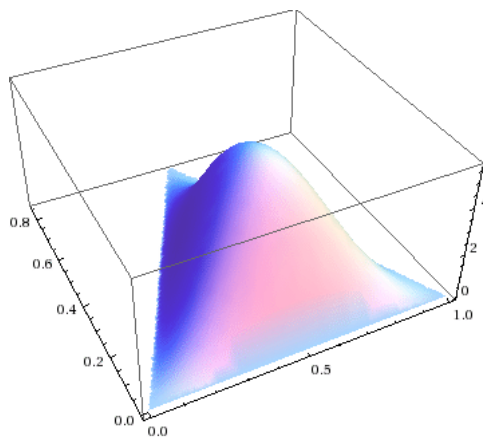




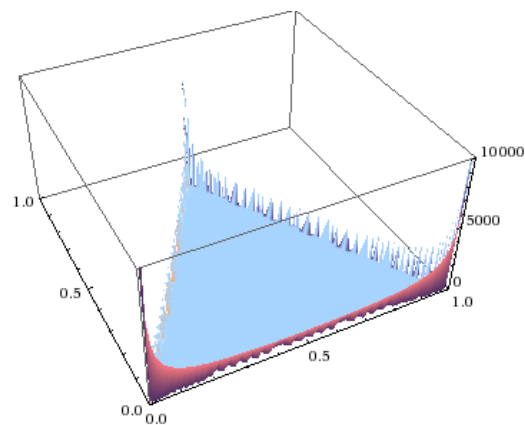
# ディリクレ分布 (2)



$$\alpha = (2, 4, 2)$$



$$\alpha = (2, 2, 2)$$



$$\alpha = (0.5, 0.5, 0.5)$$

- $\alpha_k > 1$  のとき、上に凸
- $\alpha_k < 1$  のとき、下に凸
  - 統計的自然言語処理等では、多くの場合  $\alpha \ll 1$  ( $\alpha = 0.1 \sim 0.0001$  くらい)



# ディリクレ分布に基づく予測

- ゆがんだ三面サイコロを振ったら、結果は  $X = \{1, 2, 2, 3, 2, 3\}$  (1=1回, 2=3回, 3=2回) だった。次の目は?

- ベイズの定理:  $p(\mathbf{p}|X) \propto p(X|\mathbf{p})p(\mathbf{p})$

$$\propto (p_1^1 \cdot p_2^3 \cdot p_3^2) \cdot \left( \prod_{k=1}^3 p_k^{\alpha_k - 1} \right)$$

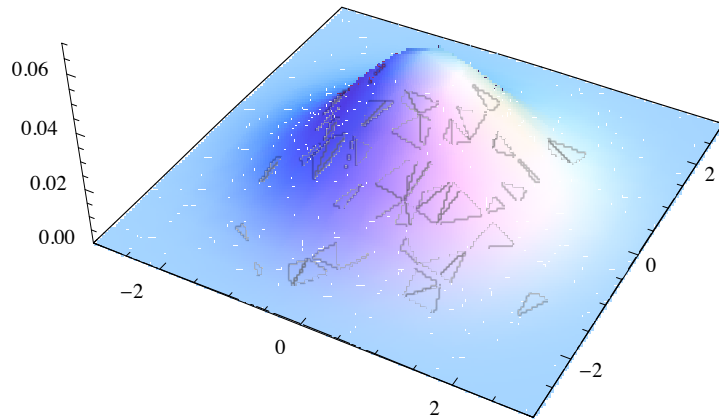
$$= p_1^{\alpha_1 + 1 - 1} \cdot p_2^{\alpha_2 + 3 - 1} \cdot p_3^{\alpha_3 + 2 - 1}$$

$$= \text{Dir}(\alpha_1 + 1, \alpha_2 + 3, \alpha_3 + 2)$$

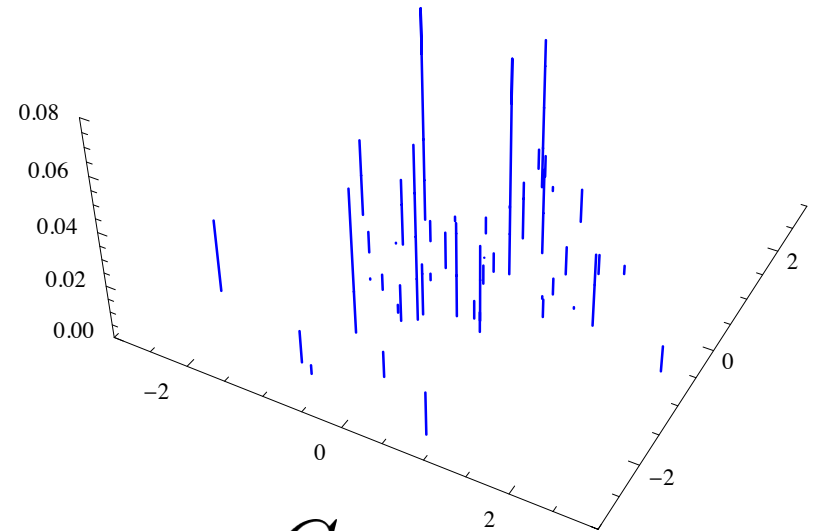
- $\mathbf{p}$  の期待値は、

$$E[\mathbf{p}|X] = \left( \frac{\alpha_1 + 1}{\alpha + 6}, \frac{\alpha_2 + 3}{\alpha + 6}, \frac{\alpha_3 + 2}{\alpha + 6} \right) \quad (\alpha = \sum_k \alpha_k)$$

# ディリクレ過程



$G_0$



$G$

- 基底測度  $G_0$  に似た、無限次元の離散測度 (atomic measure)  $G$  を生成する確率過程

$$G \sim \text{DP}(\alpha, G_0)$$

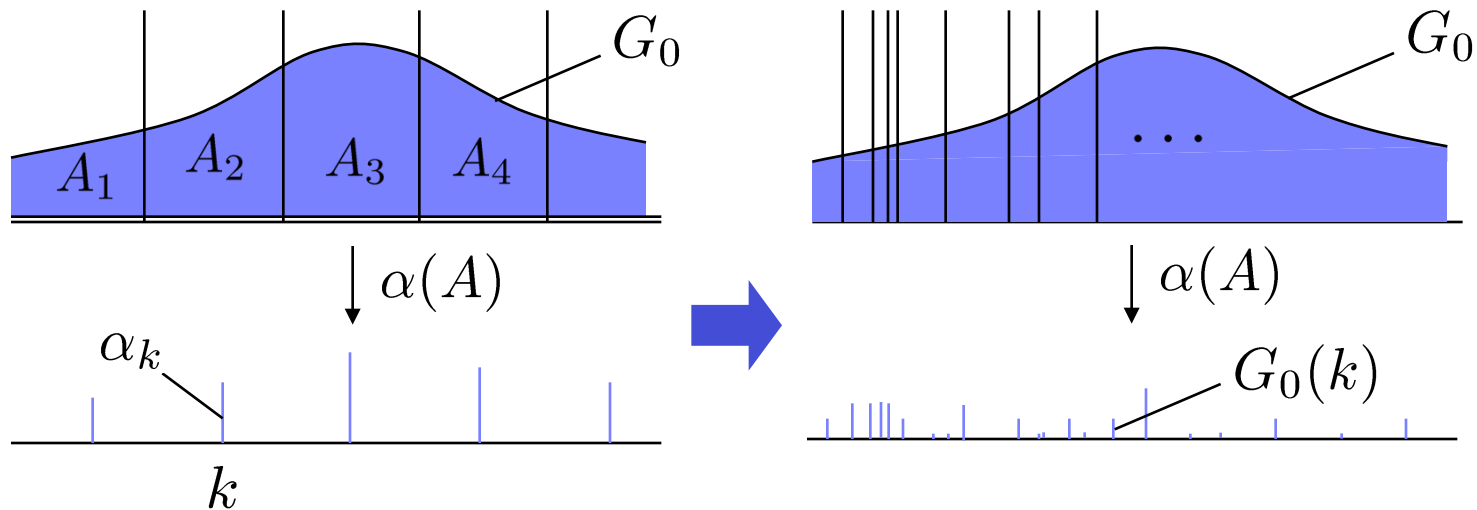
# ディリクレ過程 (2)

- Dirichlet process とは要するに何?  
→ 無限次元ディリクレ分布.
- DP の定義 (Ferguson 1973):

A stochastic process  $P$  is said to be a Dirichlet process on  $(\mathcal{X}, \mathcal{A})$  with parameter  $\alpha$  if for any measurable partition  $(A_1, \dots, A_k)$  of  $\mathcal{X}$ , the random vector  $(P(A_1), \dots, P(A_k))$  has a Dirichlet distribution with parameter  $(\alpha(A_1), \dots, \alpha(A_k))$ .

- つまり..

# ディリクレ過程 (3)



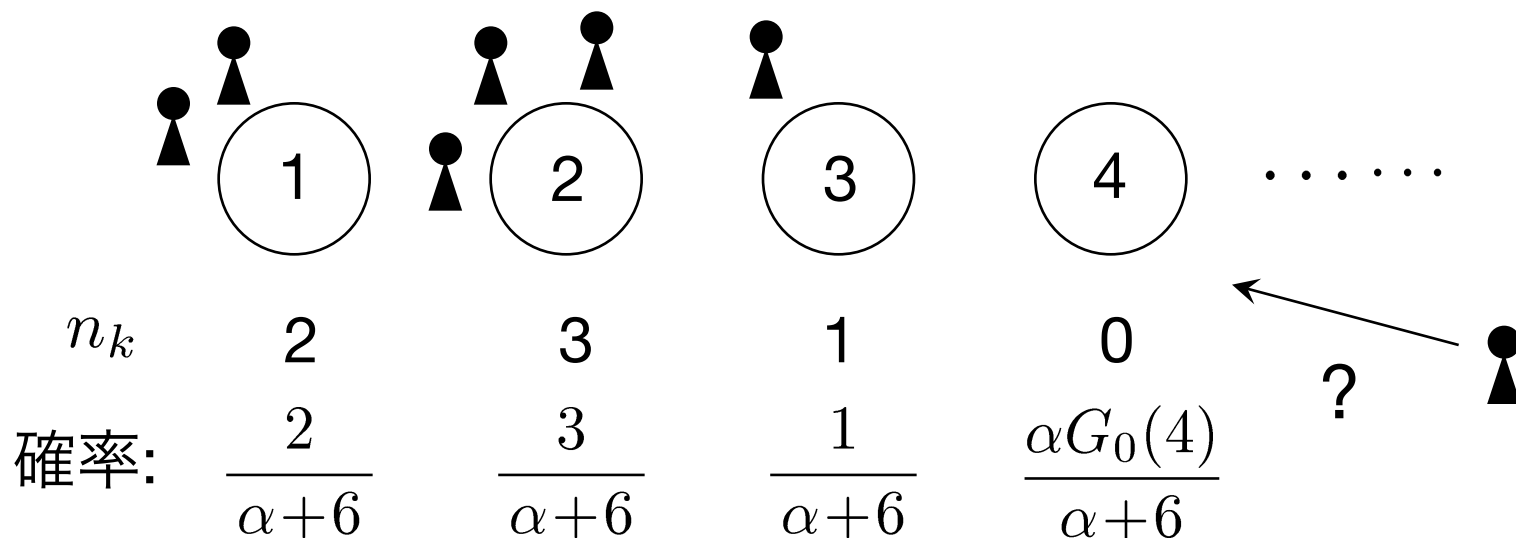
- ディリクレ過程: 任意に細かいPartitionに対して、常にその上で離散分布がディリクレ分布に従う。
- 有限次元に周辺化すれば、普通のディリクレ分布

# Chinese Restaurant Process (CRP)

- 予測確率

$$p(k|X) = \frac{\alpha_k + n_k}{\alpha + n} \quad (\text{Dirichlet}), \quad \frac{\alpha G_0(k) + n_k}{\alpha + n} \quad (\text{DP})$$

- ディリクレ分布/過程に従うと、頻度  $n_k$  の高いものはさらに現れやすくなる (rich-gets-richer) → CRP



# ディリクレ過程と言語モデル

- ディリクレ過程は、語彙が無限の場合の単語の確率分布ともみることができる

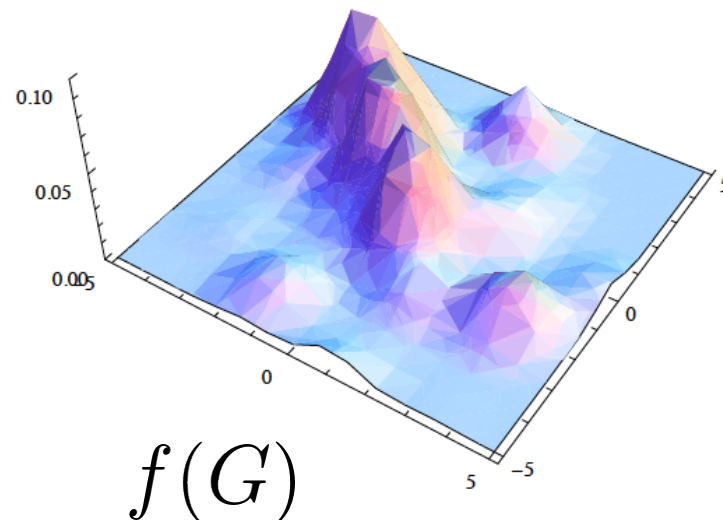
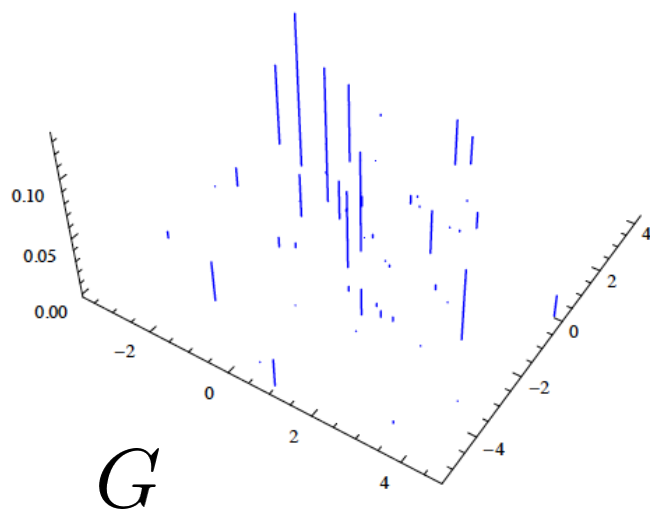
$$p(w|X) = \frac{c(w)}{\alpha+n} + \frac{\alpha}{\alpha+n} G_0(w)$$

- カウント  $c(w)$  が 0 のどんな未知の単語  $w$  でも、 $G_0(w) \cdot \alpha / (\alpha+n)$  の確率を持つ

# ディリクレ過程混合モデル

- 混合モデルのパラメータがディリクレ過程に従うとすると、クラスター数を決めない**無限混合モデル**が可能になる

$$G \sim \text{DP}(\alpha, G_0), \mathbf{x}_i \sim f(G) \text{ i.i.d.}$$





# ディリクレ過程混合モデル (2)

- MCMC推論: 各データ  $\mathbf{x}_i$  に、それを生成したクラスタ番号  $z_i \in \{1 \cdots \infty\}$  を割り当てる
  - ベイズの定理:

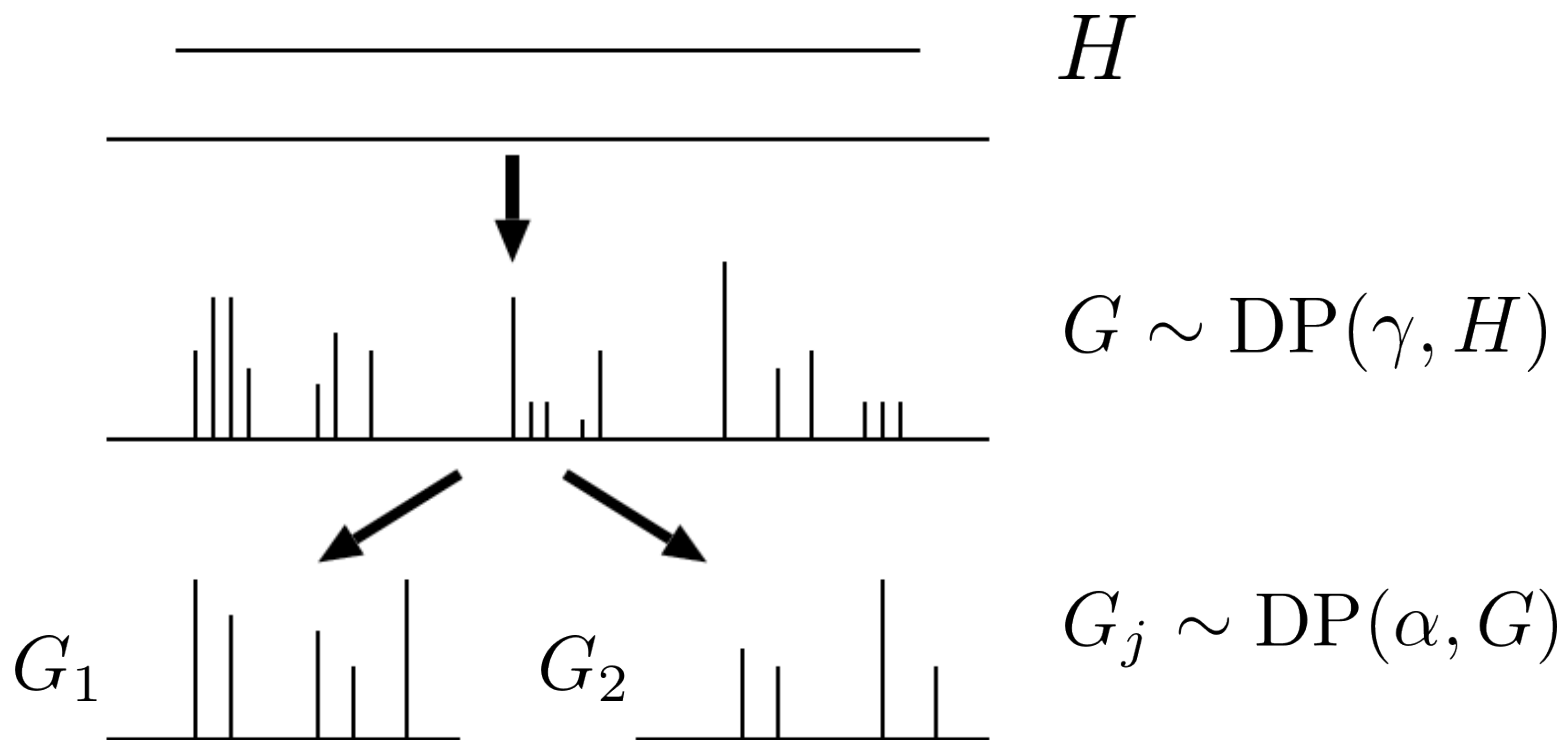
$$p(z_i | \mathbf{x}_i) \propto p(\mathbf{x}_i | z_i) p(z_i)$$

- よって、

$$p(z_i | \mathbf{x}_i) \propto \begin{cases} \frac{n_k}{n + \alpha} p(\mathbf{x}_i | \theta_k) & (k = 1, \dots, K) \\ \frac{\alpha}{n + \alpha} p(\mathbf{x}_i | \theta_{new}) & (k = K + 1) \end{cases}$$

# 階層ディリクレ過程 (HDP) (Teh+ 2006)

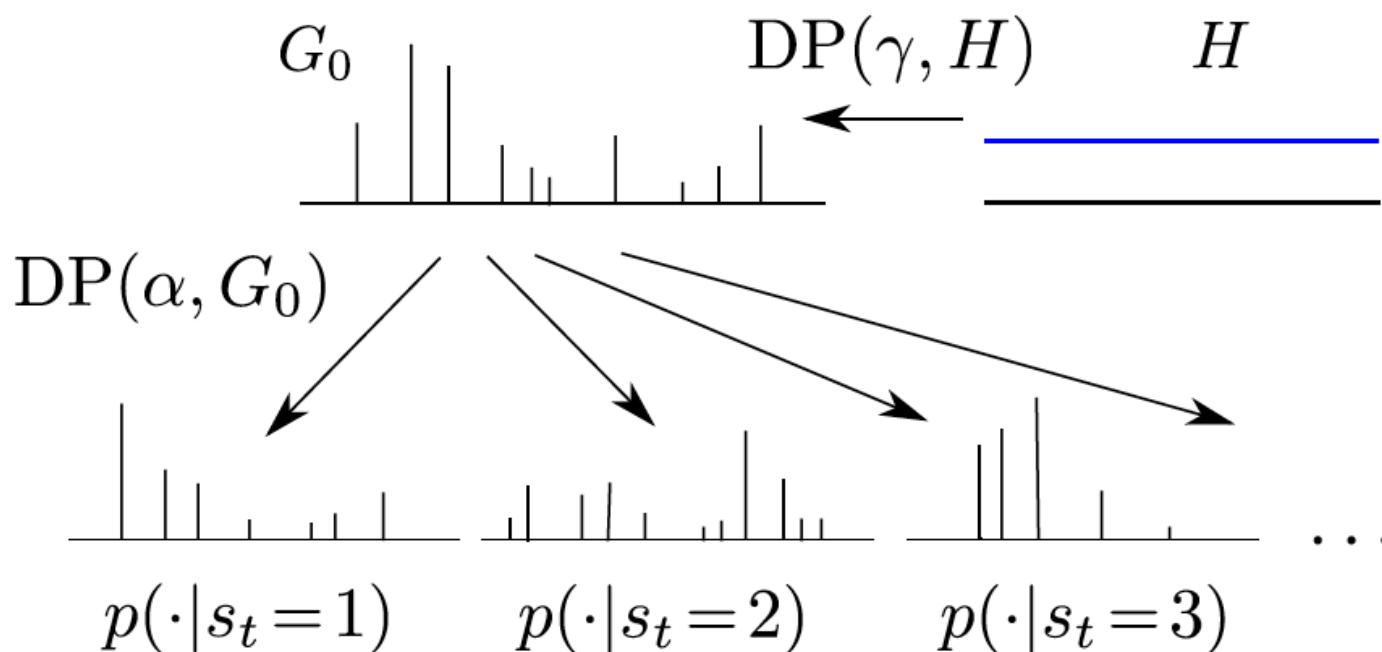
- ディリクレ過程から、さらにディリクレ過程を生成する



# HDP-HMM (無限HMM)

- **なぜHDPが必要?** →

例えば、HMMではHDPを使わず別々に状態遷移分布を生成すると、遷移先がバラバラになってしまう



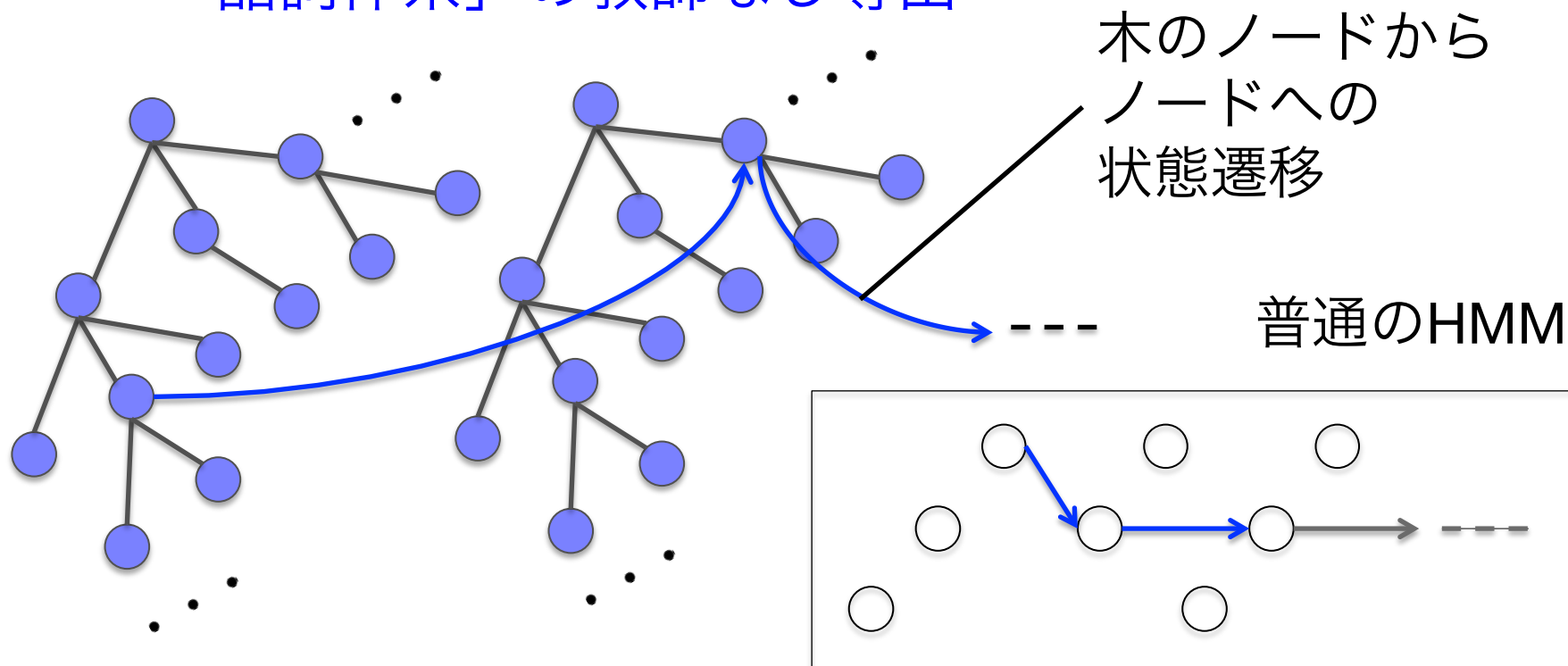
---

# 無限木構造隠れMarkovモデル

(情報処理学会自然言語処理研究会 NL-226, 2016)

# 本研究の概要

- HMMを、無限の木構造上に状態を持つように拡張
  - Infinite HMM (Beal+ 2001; Teh+ 2006)の拡張
  - 「品詞体系」の教師なし導出

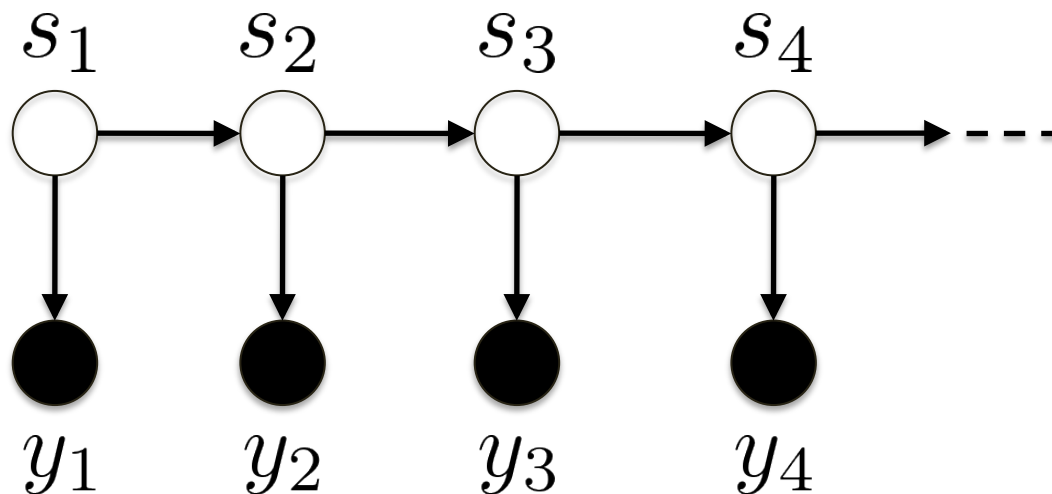


# 発表の流れ

---

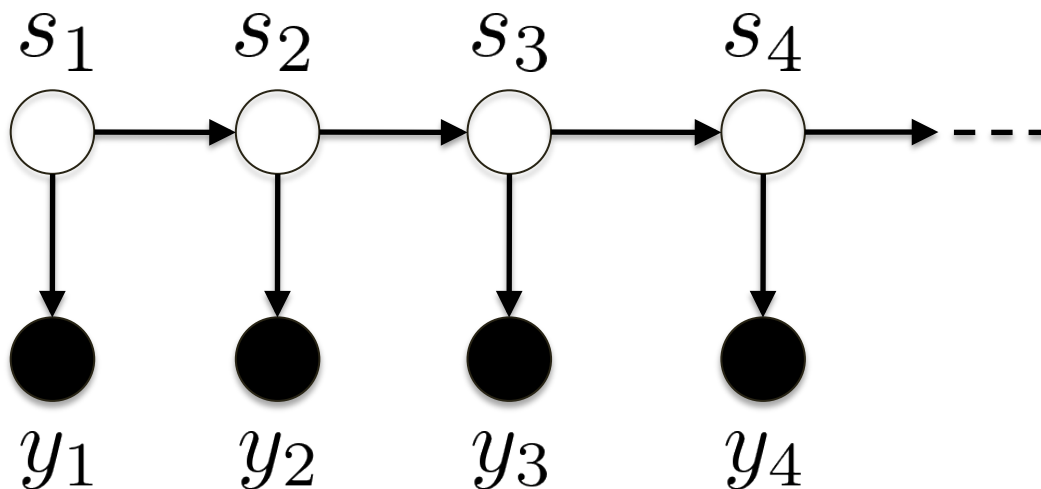
- HMMと自然言語処理
- 品詞の教師なし・半教師あり学習
- 無限隠れMarkovモデル
- 木構造Stick-breaking過程 (Adams+ 2010)
- 階層的木構造Stick-breaking過程
- iTHMMと特別なMCMC法による学習
- 実験 (日本語・英語・クリンゴン語)
- まとめと展望

# 隠れMarkovモデル



- 情報科学に共通する基礎的なモデル
- あらゆる場所で使われている
  - 自然言語処理、音声認識
  - ロボティクス、バイオインフォマティクス、経済..

# 隠れMarkovモデル



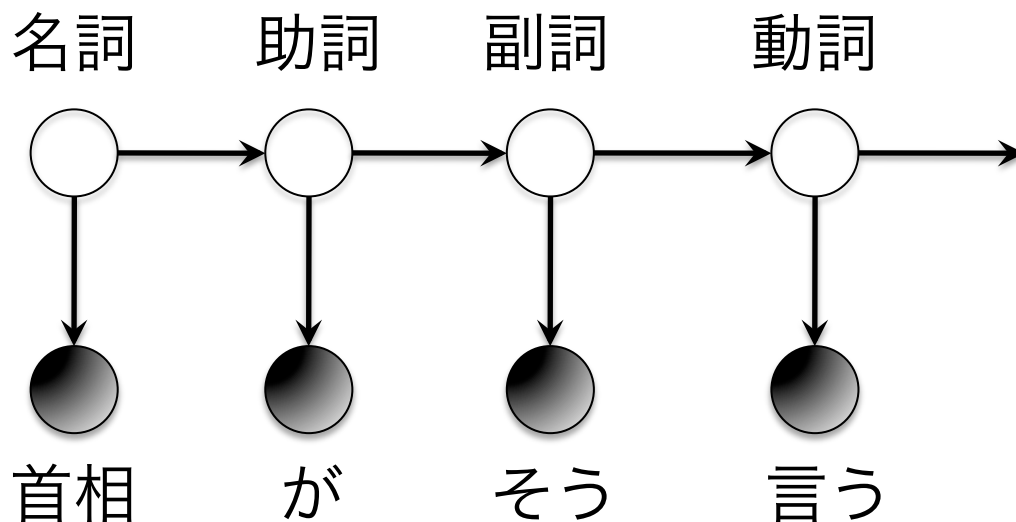
- 観測系列  $y_1 \cdots y_T$  の背後に、隠れ状態の列  $s_1 \cdots s_T$  が存在
- 観測系列の確率を最大化：

$$p(y_1, y_2, \cdots, y_T) = \sum_{s_1 \cdots s_T} \prod_{t=1}^T p(y_t | s_t) p(s_t | s_{t-1})$$



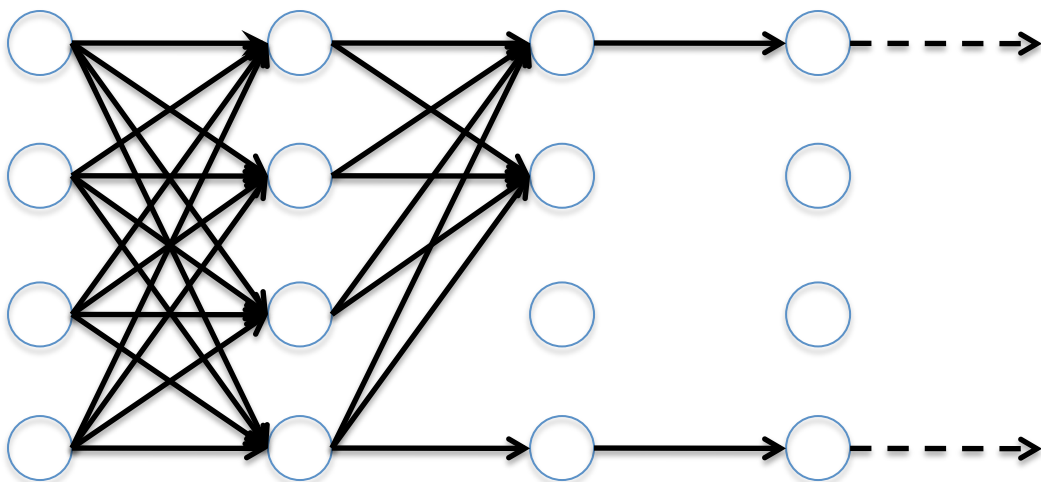
# 統計的自然言語処理でのHMM

- 最もわかりやすい例→品詞学習 (形態素解析)



- 茶釜はHMMの教師あり学習としてモデル化(竹内97)
- 半教師あり学習にも教師なしモデルとして不可欠 (Suzuki+08)

# HMMの学習法: 最尤推定



- 可能なパスは指数的( $K^T$ 個)に存在...動的計画法

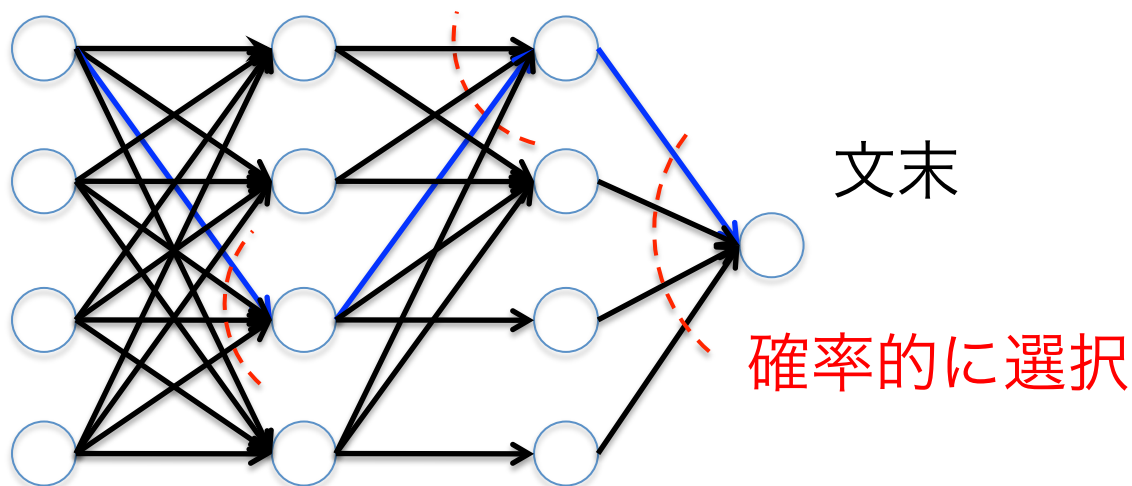
$$\alpha_t(s) = p(y_t = s, x_1 \cdots x_t) \quad (\text{内側確率})$$

$$= \sum_k p(x_t | y_t = s) p(y_t = s | y_{t-1} = k) \alpha_{t-1}(k)$$

- デコード時には、確率最大のパスを1つだけ、動的計画法で求める (Viterbiパス)

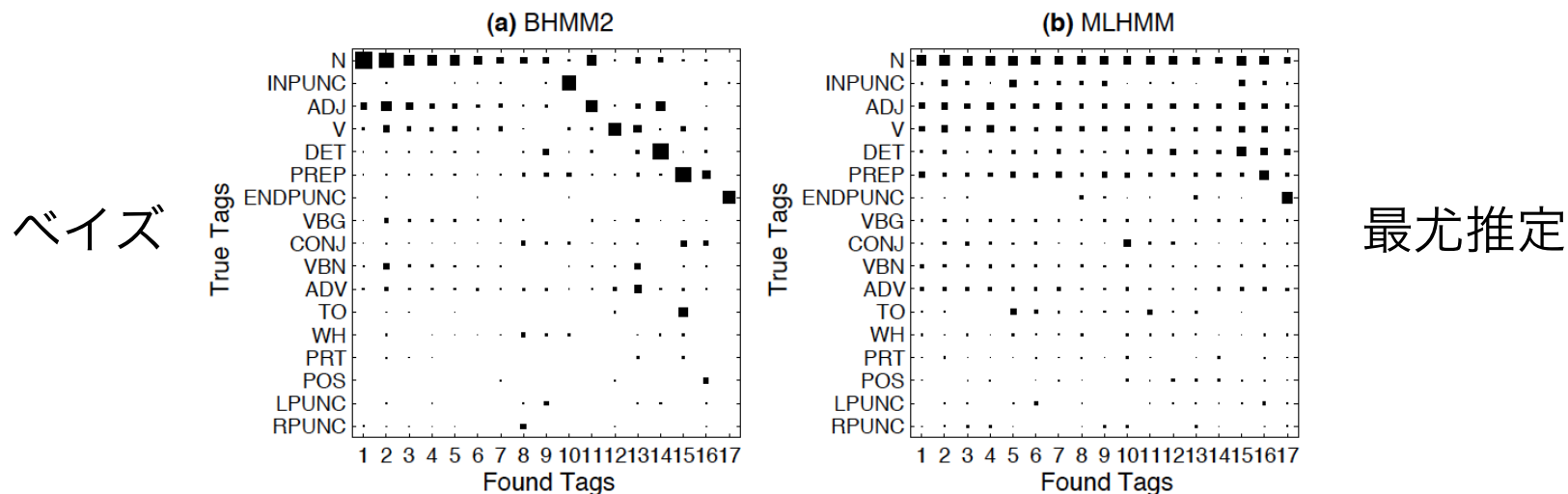
# HMMの学習法: ベイズ推定

- MCMC: 各データの持つ状態系列を実際にサンプリング
- Forward Filtering-Backward Sampling (Scott 2002)



- 内側確率を計算しておいて、文末から**確率的に選択** (確率的 Viterbi)

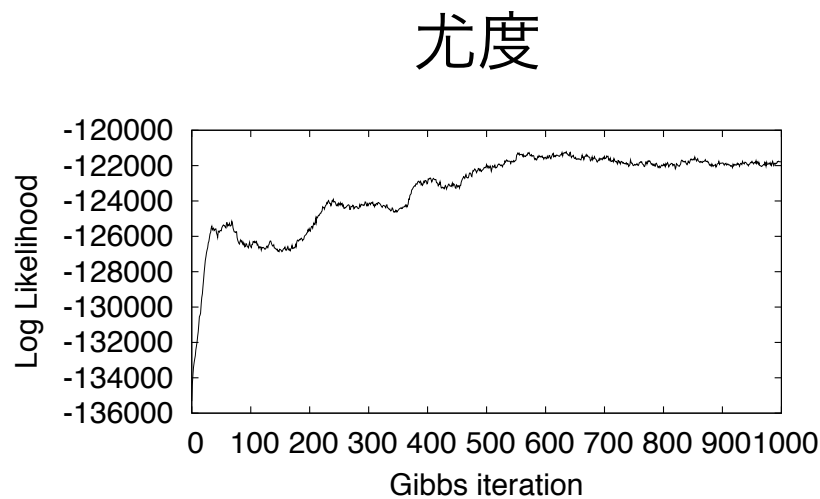
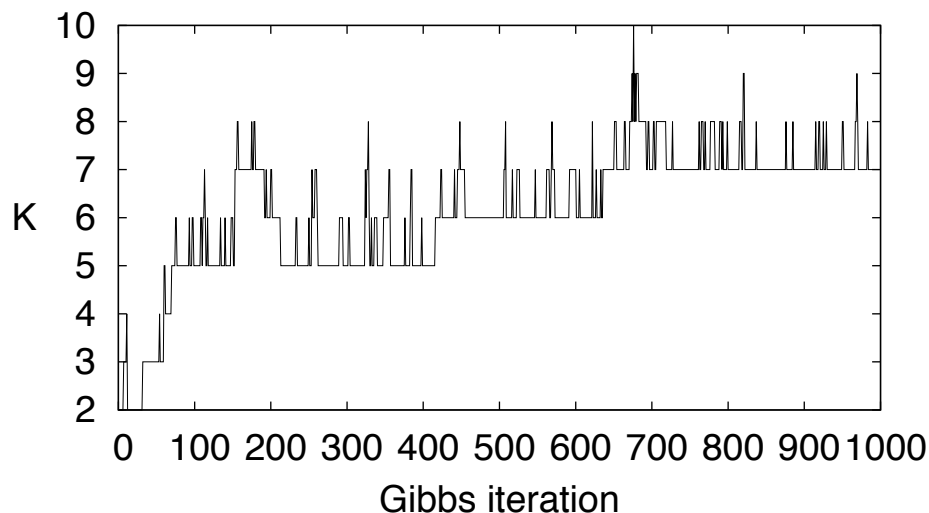
# 教師なし品詞解析



- 1990年代: Merialdo+(1994), Kupiec(1992)→失敗
- 2000年代: ベイズ学習で成功 (Goldwater+07, van Gael+ 09)
  - Baum-Welchは最尤推定なので局所解にはまる
  - MCMC法による学習

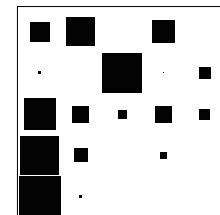
# 無限隠れMarkovモデル

- ノンパラメトリックベイズ法により、隠れ状態数  $K$  すら推定できる
- Forward-backwardも可能 (van Gael+ 07)



# “Alice in Wonderland”の解析

状態遷移行列

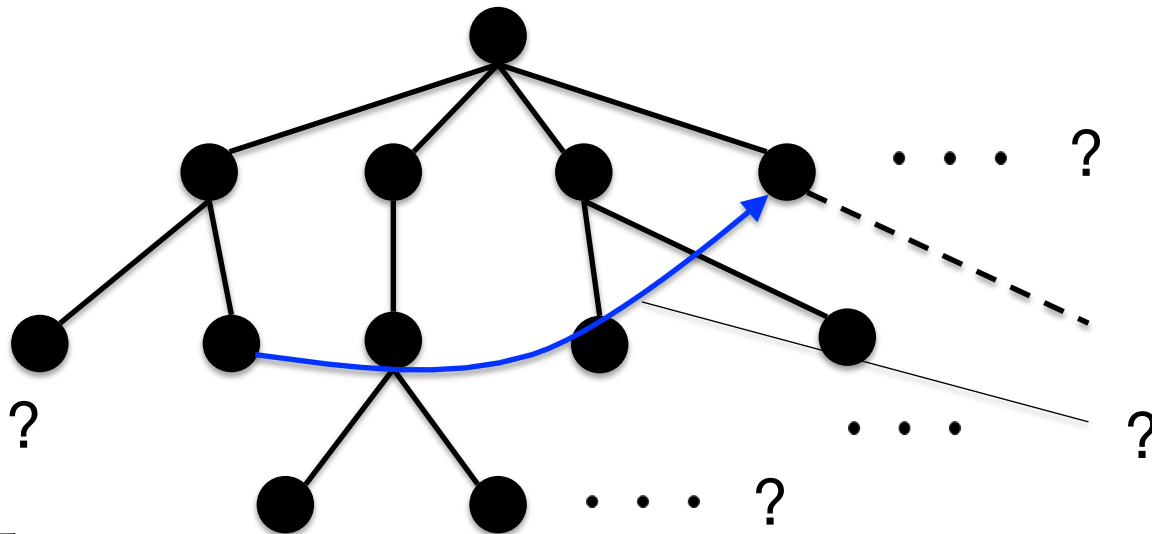


1		2		3		5	
she	432	the	1026	was	277	way	45
to	387	a	473	had	126	mouse	41
i	324	her	116	said	113	thing	39
it	265	very	84	\$	87	queen	37
you	218	its	50	be	77	head	36
alice	166	my	46	is	73	cat	35
and	147	no	44	went	58	hatter	34
they	76	his	44	were	56	duchess	34
there	61	this	39	see	52	well	31
he	55	\$	39	could	52	time	31
that	39	an	37	know	50	tone	28
who	37	your	36	thought	44	rabbit	28
what	27	as	31	herself	42	door	28
i'll	26	that	27	began	40	march	26

# これで充分か…?

- 京大コーパスやBCCWJ等の実際の品詞は、**階層化**されている
  - 名詞—一般名詞—地名
  - 動詞—他動詞—サ変
- 構文解析でのシンボル細分化 (松崎05, 進藤12など):
  - VP-1, ADVP-5 のように文法的カテゴリを細分化
  - ただし、一段階のみしか不可能
- 「品詞体系」を統計的に導出できないか?

# 階層的な隠れ状態の学習



- 問題:

- 各分岐の数を何個にすればよいのか? (無限の選択)
- どの深さまで階層を考えればよいのか? (指数爆発)
- ノード間の遷移確率をどう考えればよいのか?

⇒ ナイーブな方法では不可能!

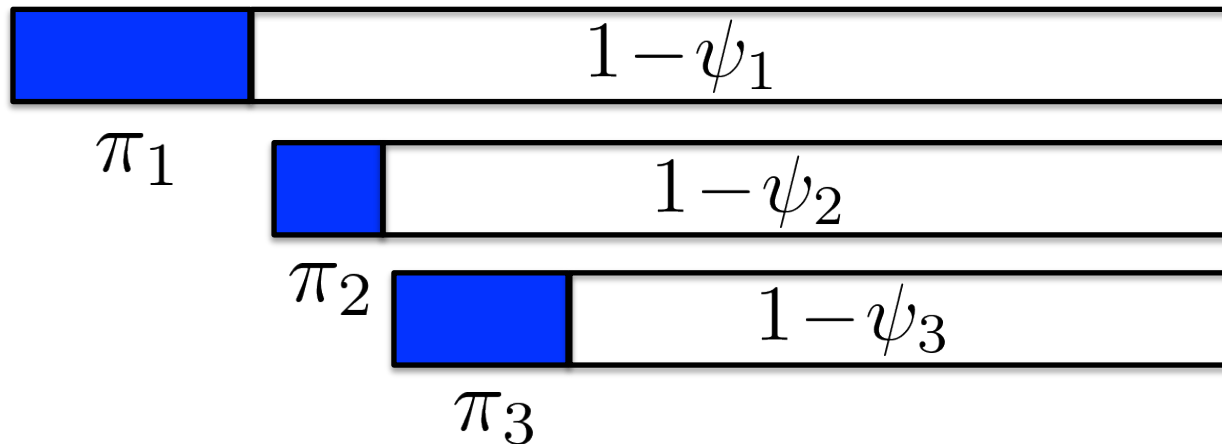


# 無限木構造を生成するモデル

- 木構造Stick-breaking過程 (Tree-structured stick-breaking process, TSSB)  
(Adams+ NIPS 2010):
  - 無限の深さと分岐を持つ木構造上の離散分布を生成する確率過程
  - Stick-breaking過程 (=Dirichlet process)の拡張

# Stick-breaking process (SBP)

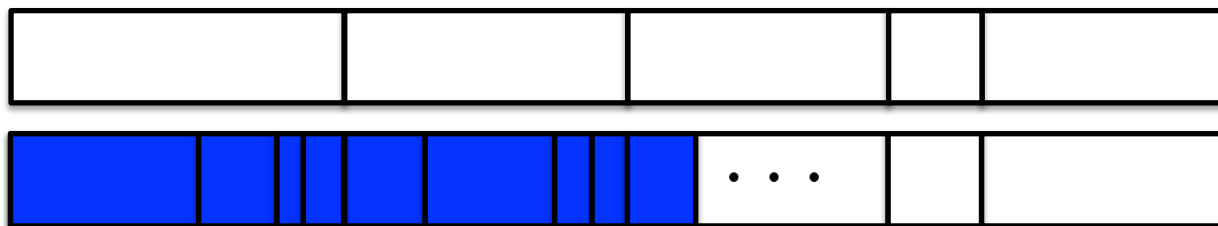
- 無限次元の多項分布  $\pi = (\pi_1, \pi_2, \pi_3, \dots)$  を生成する確率過程
  - ディリクレ過程と等価 (Sethuraman 1994)



$$\pi_k = \psi_k \prod_{j=1}^{k-1} (1 - \psi_j), \quad \psi_j \sim \text{Be}(1, \gamma)$$

# 階層的離散分布

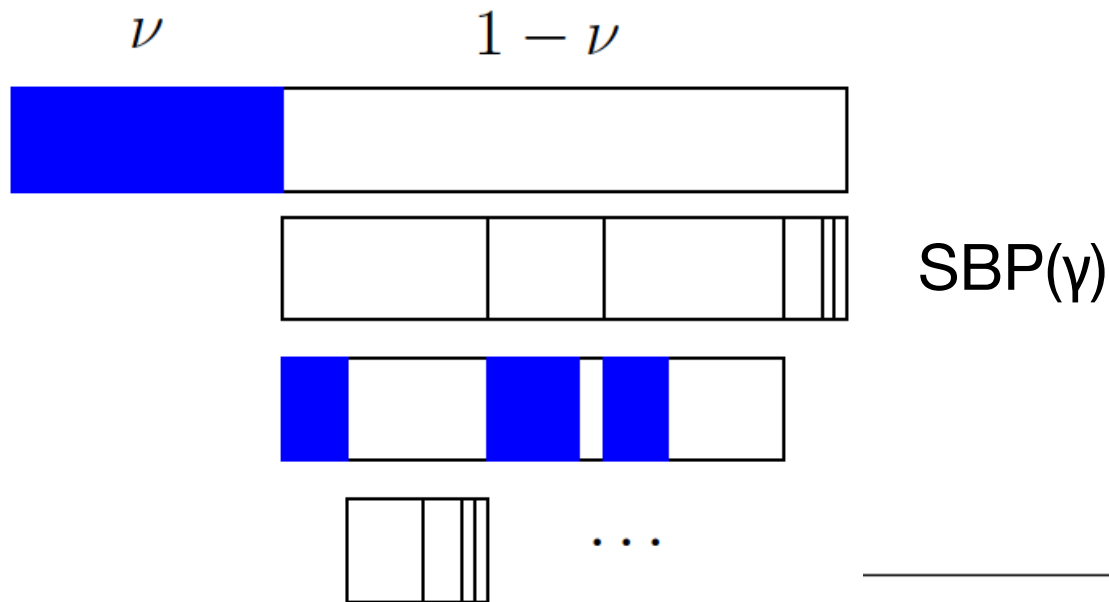
- 最も簡単な階層的離散分布:  
SBPの各stick  $\pi_k$  を、再帰的にさらにSBPで分割  
(Polya trees)



- これだと、データは常に最も細かいカテゴリにしか存在しない
  - 「よくわからないが、動詞なことは確か」な言葉?
  - “thing”, “way” など、抽象的な名詞?

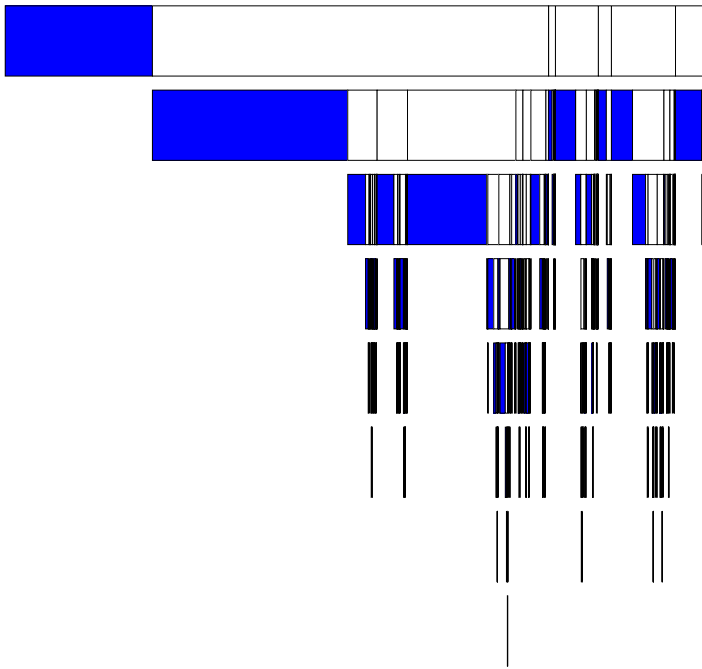
# Tree-structured stick-breaking process

- TSSB: 先にまず、“そのカテゴリで止まる確率”  $\nu$  を生成
  - (1)  $\nu \sim \text{Be}(1, \eta)$  で棒を分割して、 $\pi_s$  を生成
  - (2) 残った  $(1-\nu)$  を  $\text{SBP}(\gamma)$  で分割して、各stickに同じ操作を適用.

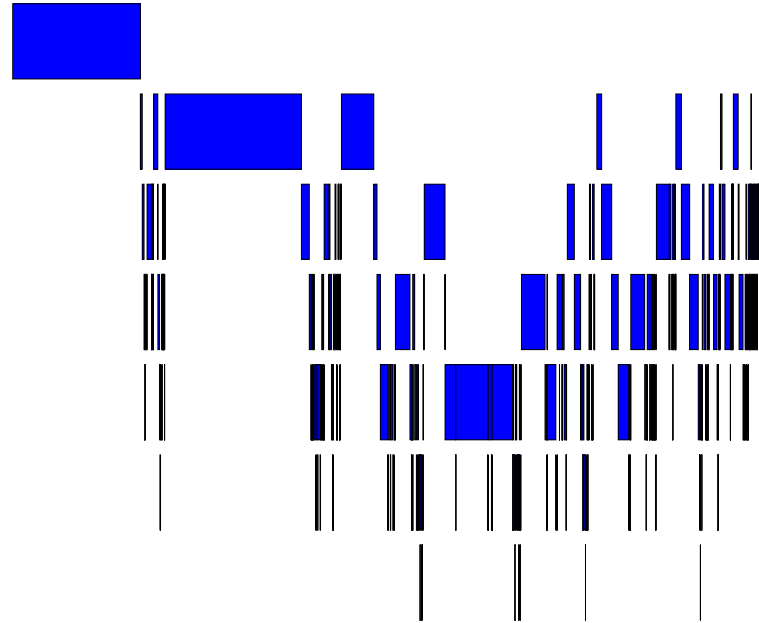


# TSSBの構成

- $\pi \sim \text{TSSB}(\eta, \gamma)$  から実際に生成したサンプル

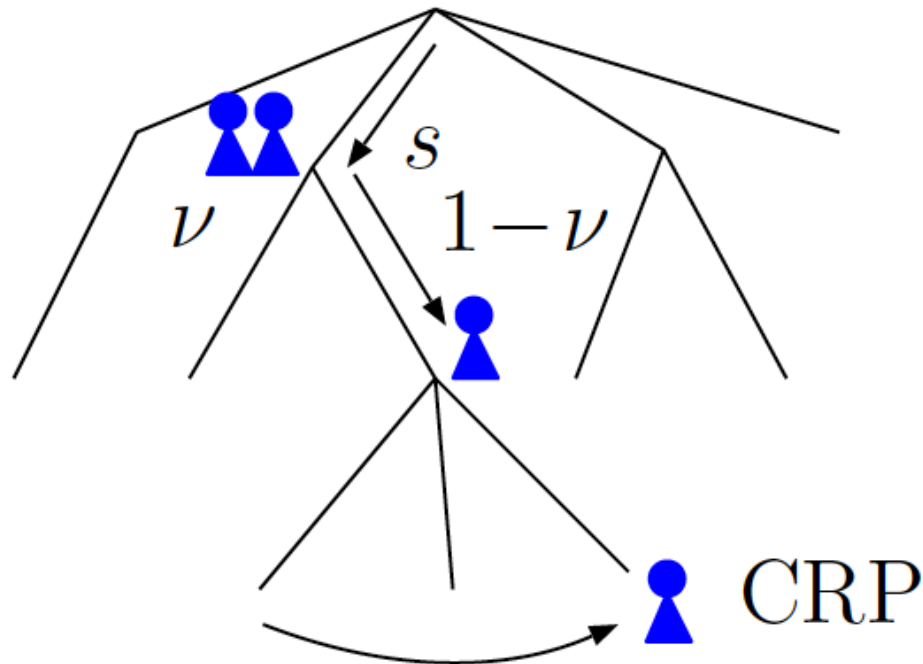


TSSB(1,1,1)



TSSB(1,5,0.5)

# TSSBのCRP(CDP)表現



- 客を木の根から辿って追加
  - 確率 $\nu$ でそのノードに残る
  - 確率 $(1-\nu)$ で子供に降り、CRPで子供を選択

# TSSBの確率モデル

- TSSBは無限木構造に対応し、そのノードは整数列

$$\mathbf{s} = s_1 s_2 s_3 \cdots$$

で番号づけられる (例:  $\mathbf{s} = [2\ 1\ 3]$ )

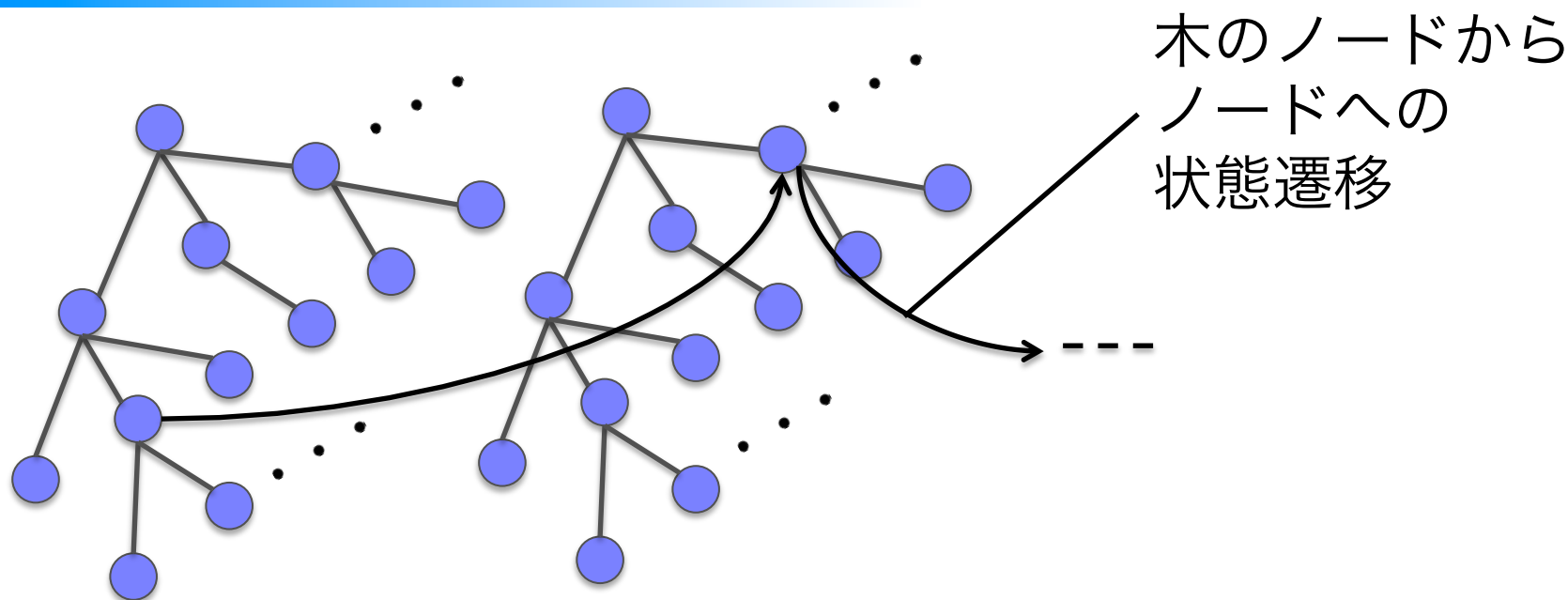
- ノード  $\mathbf{s}$  の確率  $\pi_{\mathbf{s}}$  は、縦方向と横方向のSBPの積

$$\pi_{\mathbf{s}} = \nu_{\mathbf{s}} \prod_{\mathbf{s}' \prec \mathbf{s}} (1 - \nu_{\mathbf{s}'}) \cdot \prod_{\mathbf{s}' \prec \mathbf{s}} \phi_{\mathbf{s}'}$$

$$\phi_{\mathbf{s}k} = \psi_{\mathbf{s}k} \prod_{j=1}^{k-1} (1 - \psi_{\mathbf{s}j}) \quad \text{SBP}(\gamma)$$

$$\psi_{\mathbf{s}j} \sim \text{Be}(1, \gamma)$$

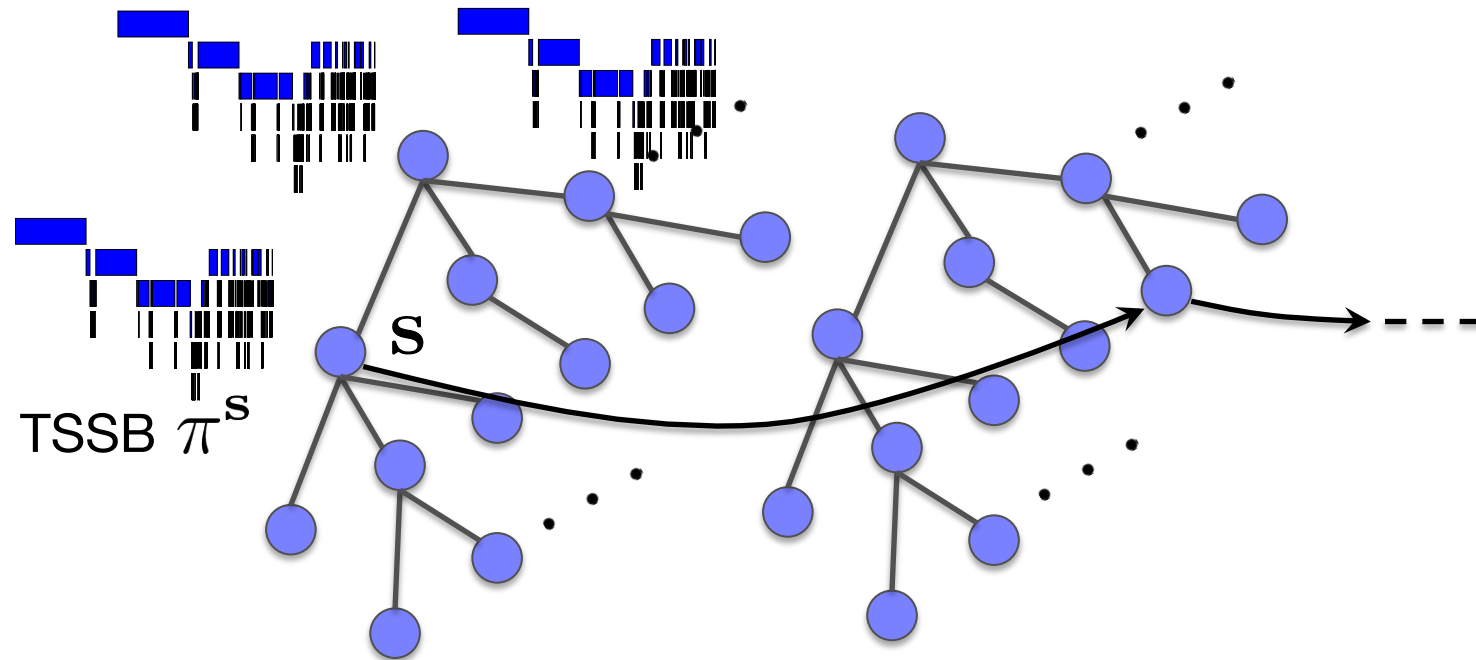
# TSSB上の状態遷移モデル



- HMMでは、無限木構造のノード間に状態遷移
- 木構造の各ノードが、次の時刻の木構造への確率分布を持っている
  - 普通のHMMのときは単純な $K \times K$ の遷移行列



# TSSB上の状態遷移モデル (2)



- 各ノード  $s$  が、次の状態への確率分布(TSSB)  $\pi^s$  を持っている

# TSSB上の状態遷移モデル (3)

- $\pi^S$  は独立ではない!
  - $[1\ 2\ 4]$  = 「名詞-固有名詞-一般」からの遷移確率は、  
 $[1\ 2]$  = 「名詞-固有名詞」を引き継いでいる
  - $[1\ 2]$  は  $[1]$  を、 $[1]$  は  $[\ ]$  に影響されている



階層モデル!

TSSB

$$\pi^{[1\ 2\ 4]} \sim \text{HTSSB}(\alpha, \pi^{[1\ 2]})$$

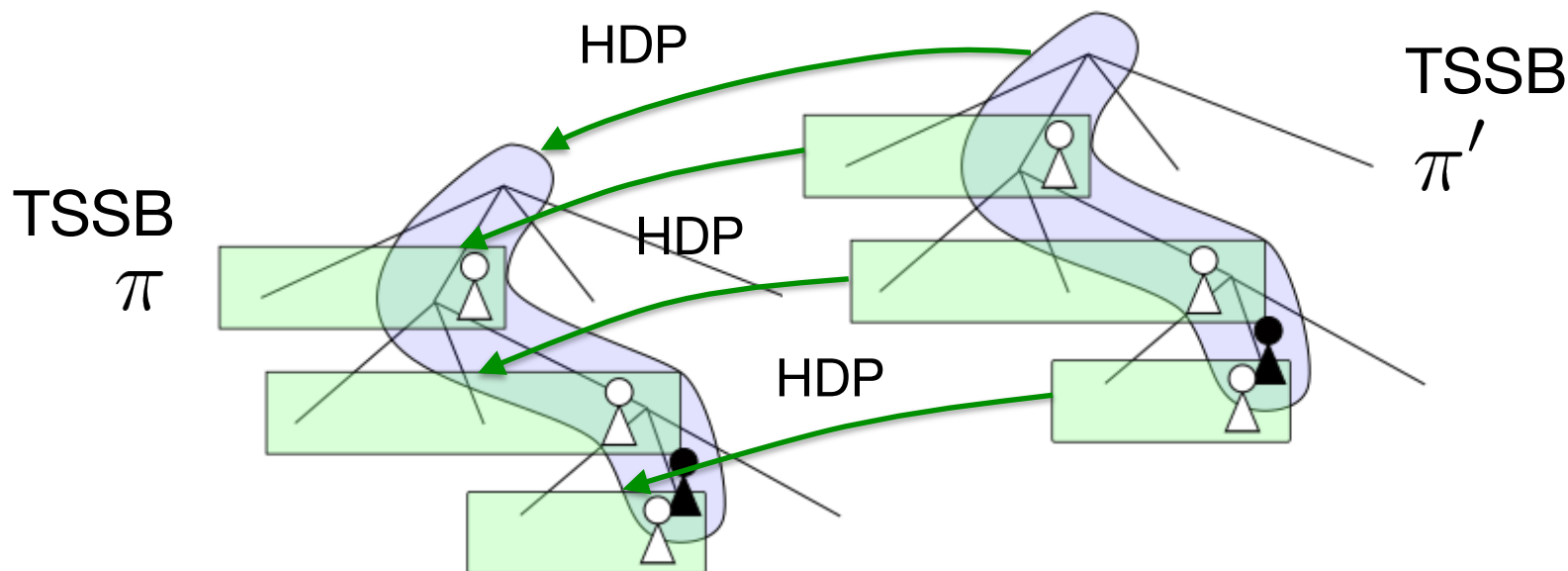
$$\pi^{[1\ 2]} \sim \text{HTSSB}(\alpha, \pi^{[1]})$$

$$\pi^{[1]} \sim \text{HTSSB}(\alpha, \pi^{[\ ]})$$

# 階層的木構造Stick-breaking過程 (HTSSB)

$$\pi \sim \text{HTSSB}(\alpha, \pi')$$

- $\pi$  を構成するSBP(=ディリクレ過程)が、親の  $\pi'$  の対応するディリクレ過程から生成されている  
→階層ディリクレ過程 (HDP)



# HTSSB (2)

- HDPのStick-breaking表現より、データDの下で

$$E[\nu_s | D] = \frac{\alpha \nu'_s + n_0(\mathbf{s})}{\alpha(1 - \sum_{\mathbf{u} \prec \mathbf{s}} \nu'_u) + n_0(\mathbf{s}) + n_1(\mathbf{s})}$$

$$E[\psi_{sk} | D] = \frac{\alpha \psi'_{sk} + m_0(\mathbf{sk})}{\alpha(1 - \sum_{j=1}^{k-1} \psi'_{sj}) + m_0(\mathbf{sk}) + m_1(\mathbf{sk})}$$

- $\nu'_s, \psi'_s$  は親の  $\pi'$  での  $E[\nu'_s | D], E[\psi'_s | D]$  の値
- 親の  $\nu'_s, \psi'_s$  はさらにその親の  $\nu''_s, \psi''_s$  に依存！  
→ (すさまじい)再帰的な計算が必要

# HTSSB (3)

- $E[\nu_s|D], E[\psi_s|D]$  がわかれば、 $\pi$  の各要素 $s$ での確率は

$$\pi_s = \nu_s \prod_{s' \prec s} (1 - \nu_{s'}) \cdot \prod_{s' \preceq s} \phi_{s'},$$

$$\phi_{sk} = \psi_{sk} \prod_{j=1}^{k-1} (1 - \psi_{sj})$$

# HTSSBの学習

---

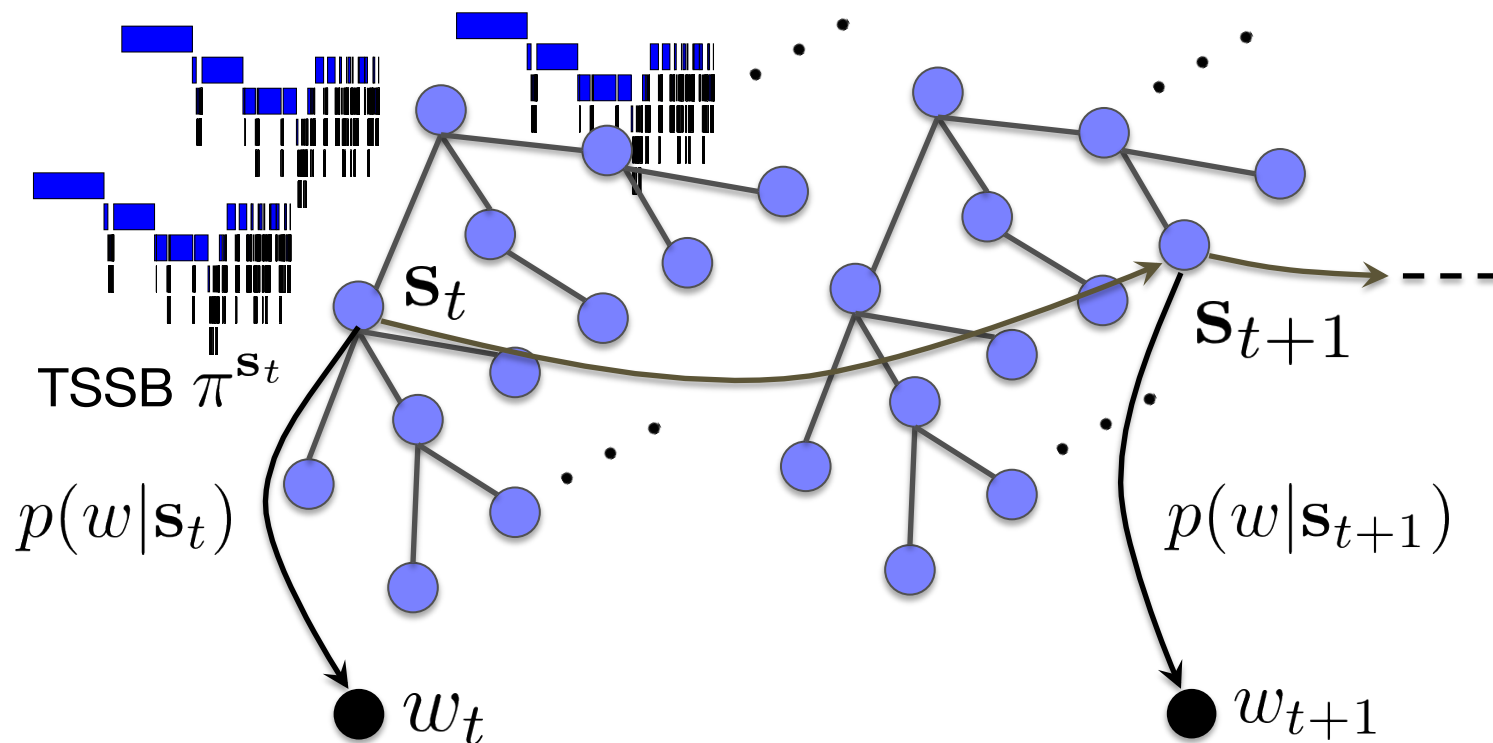
- TSSB  $\leftrightarrow$  CDPで事後分布



- HTSSB  $\leftrightarrow$  HCDP  
(階層的Chinese District Process)で事後分布
  - HDPに対する階層的CRPと同様
  - 詳細は、論文を参照ください

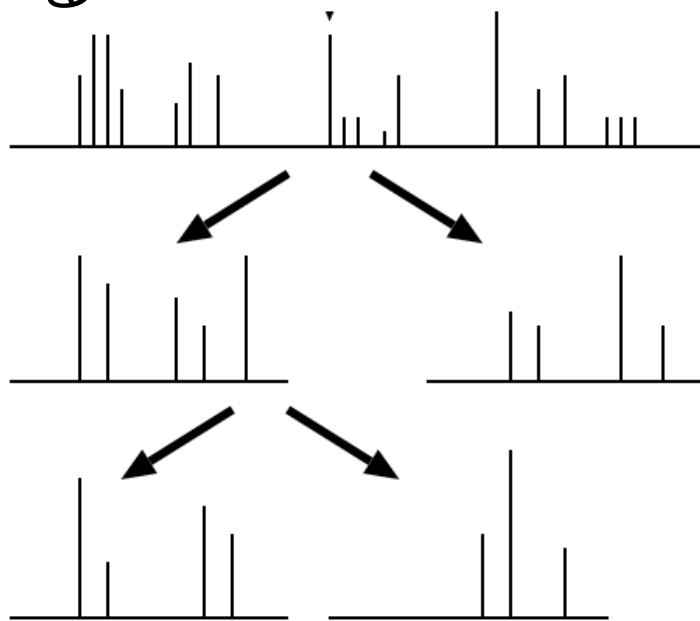
# 無限木構造HMM (iTHMM)

- HTSSBにより、無限木構造上の状態遷移確率とその事後分布が計算できる  
→ HTSSB-HMM = Infinite Tree HMM (iTHMM)



# iTHMMの単語出力確率

- 親子関係にある  $p(w|s)$  と  $p(w|s')$  は独立ではない
  - [2 1]=“動詞-動作” ~ [2]=“動詞”
- 本研究では、階層Pitman-Yor過程 (Teh 2006) を用いる



- ハイパーパラメータ  $d, \theta$  も自動推定
- カウントの追加/削除で、木構造上の分布が自動的に更新



# 無限木構造HMMの生成モデル

- iTHMMの生成モデル

(1) TSSB  $\pi^{\square} \sim \text{TSSB}(\eta, \gamma, \lambda)$  を生成.

(2) 無限木構造の各ノード $s$ について、

(a) 状態遷移確率  $\pi^s$  を親の  $\pi^{s'}$  から

$$\pi^s \sim \text{HTSSB}(\alpha, \pi^{s'})$$

と生成.

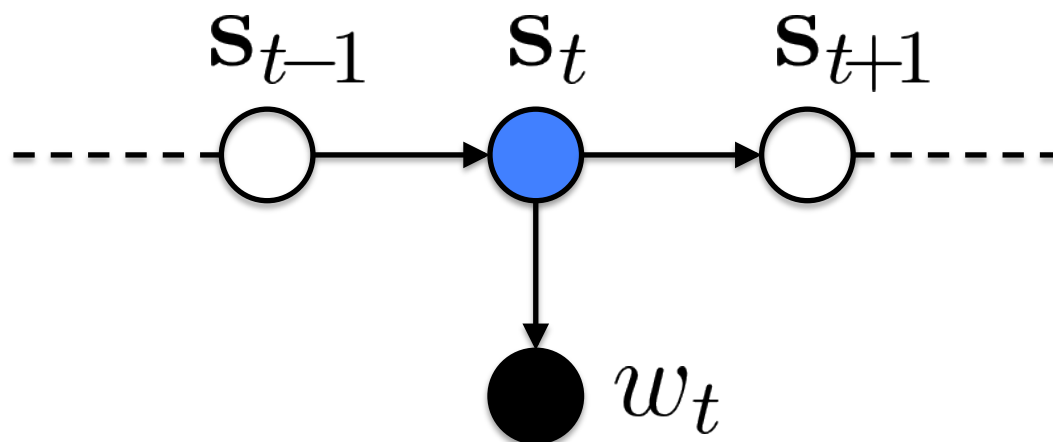
(b) 単語出力確率分布  $G_s$  を親の  $G_{s'}$  から

$$G_s \sim \text{HPY}(d_{|s|}, \theta_{|s|}, G_{s'})$$

と生成.

- BOSから始めて、隠れ状態列  $s_1, s_2, \dots$  と単語列  $w_1, w_2, \dots$  を生成.

# iTHMMの学習



- Gibbsサンプリング (Goldwater+2007)

$$p(\mathbf{s}_t | w_t, \mathbf{s}_{t+1}, \mathbf{s}_{t-1})$$

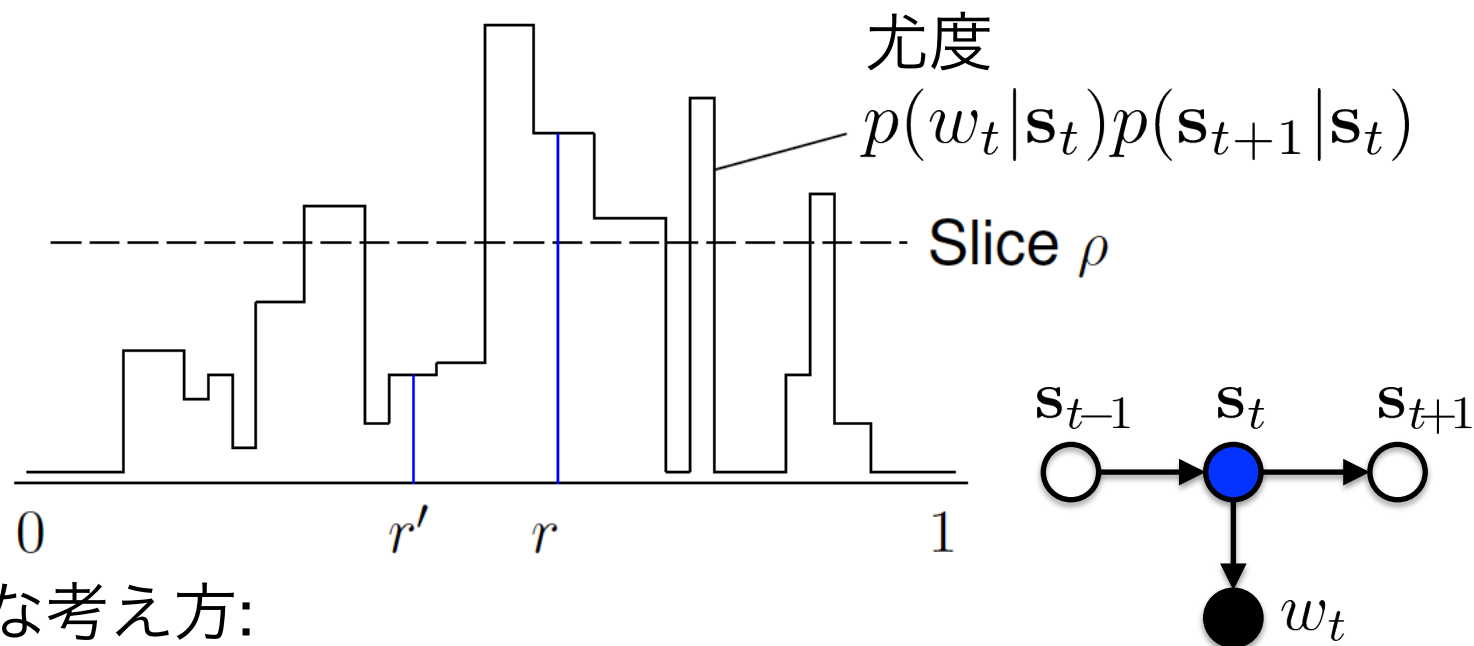
$$\propto p(w_t | \mathbf{s}_t) p(\mathbf{s}_{t+1} | \mathbf{s}_t) p(\mathbf{s}_t | \mathbf{s}_{t-1})$$

- $\mathbf{s}_t$  を次々とサンプリング  $\rightarrow$  正しい値に収束

# iTHMMの学習 (2)

- 問題:  $s_t$  を数え上げられない!
  - $s_t = [], [1\ 1], [1\ 1\ 2], [2\ 4\ 3], [17\ 5\ 3], \dots$   
と無限に候補が存在
  - iHMMのように、確率的に右側を切り落とすことはできない
  - どうするか?

# iTHMMの学習 (3)

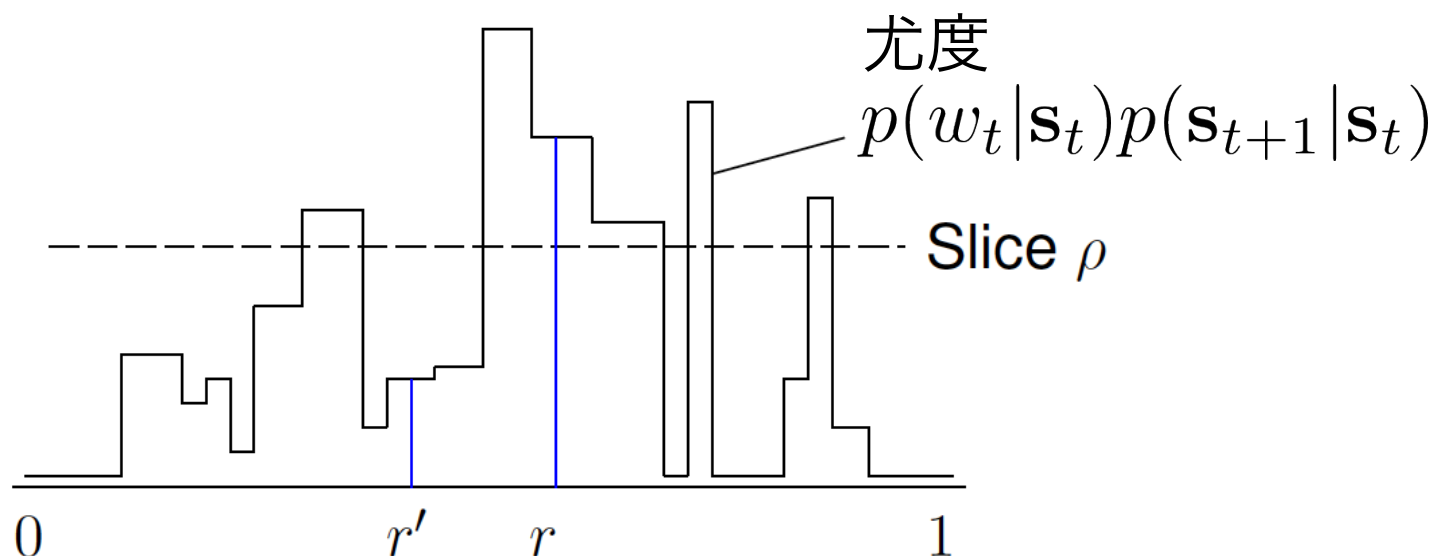


- 基本的な考え方:

$$p(w_t | s_t)p(s_{t+1} | s_t)p(s_t | s_{t-1})$$

- から  $s_t$  をランダムにサンプルするには、まず  $p(s_t | s_{t-1})$  から  $s_t$  を一様に選び、それを尤度  $p(w_t | s_t)p(s_{t+1} | s_t)$  に従って選べばよい

# iTHMMの学習 (3)



## ● 解法:

- $p(\mathbf{s}_t | \mathbf{s}_{t-1})$  から一様にサンプリングするには、先に一様乱数を決め、対応するノードを選べばよい (Retrospective sampling; Papaspiliopoulos 2008)
- 次に、 $p(w_t | \mathbf{s}_t) p(\mathbf{s}_{t+1} | \mathbf{s}_t)$  に比例してスライスサンプリング



# iTHMMの学習 (5)

- (1) 現在の確率  $p = p(w_t | s_t)p(s_{t+1} | s_t)$  から、スライス  $\rho = p \cdot \text{Unif}[0, 1)$  を作る
- (2) 一様乱数  $r \sim \text{Unif}(0, 1]$  を引いて、対応するノード  $s$  を求める
  - $s$  が存在しなければ作成
- (3)  $p(w_t | s)p(s_{t+1} | s) > \rho$  なら  $s$  を accept
- (4) そうでなければ、乱数の範囲を左右に変更して (一種の二分探索)、(2)に戻る

# 実装

- C++で7000行程度
  - boost::serializationのお蔭
  - 現在, 数1000単語/秒のサンプリング速度
- 無限木構造を必要に応じて実体化
  - ノード  $s_{t-1}$  からの遷移を表すTSSBで新しいノードが作られた際、もとの木構造自体を拡張
  - 各ノード $s$ のTSSB  $\pi^s$  が、もとの木構造自体と **自己同型**になっている (ポインタが張られている)
- 状態の参照カウントを管理して、Gibbsのiteration毎に不要な状態を削除して全体をリナンバー

# 実験 (1)

- 教師なし学習: “Alice in Wonderland”, 学習1200文, テスト231文

[2 3]

know	69	0.1976
think	41	0.1172
say	20	0.0568
wish	18	0.0489
wonder	16	0.0431
tell	16	0.0453
see	14	0.0343
do	12	0.0357

[2 7]

be	80
have	47
go	14
remember	11
do	11
get	11
take	10
talk	9



# 実験 (1)

- 教師なし学習: “Alice in Wonderland”, 学習1200文, テスト231文

[]			[0 0]		
next	13	0.0027	don't	50	0.0650
one	9	0.0004	could	43	0.0563
that	8	0.0017	are	31	0.0404
mind	7	0.0004	can	30	0.0391
two	7	0.0004	would	28	0.0358
indeed	6	0.0004	must	27	0.0351
round	6	0.0004	might	24	0.0311
bill	6	0.0004	should	23	0.0298

# 実験 (1)

- 教師なし学習: “Alice in Wonderland”, 学習1200文, テスト231文

[4]			[4 0]		
mock	52	0.0413	voice	33	0.0542
queen	49	0.0389	way	29	0.0495
gryphon	48	0.0381	tone	26	0.0431
hatter	34	0.0263	thing	19	0.0313
mouse	33	0.0261	side	13	0.0202
duchess	29	0.0228	bit	13	0.0211
caterpillar	27	0.0212	face	13	0.0211
cat	25	0.0196	cat	12	0.0208

# 実験 (1)

- 教師なし学習: “Alice in Wonderland”, 学習1200文, テスト231文
  - 学習が終われば、尤度の計算は通常の前向きアルゴリズム

モデル		PPL
iHMM	$\gamma=1$	384.351
	$\gamma=2$	348.773
	$\gamma=4$	329.830
	$\gamma=8$	316.036
iTHMM	$M=3$	<b>302.336</b>
	$\lambda=0.1$	350.846
	$\lambda=0.2$	357.951

# 実験 (2)

- 半教師あり学習: 京大コーパスから10000文の品詞を教師ありデータとして固定、37400文をサンプル

[0 0]

れて	356	0.2108
なら	176	0.1041
れ	173	0.1023
い	123	0.0727
なって	66	0.0389
せ	39	0.0229
せて	35	0.0205
どう	31	0.0181

[0 0 0]

に	228	0.2563
が	228	0.2563
の	196	0.2203
を	156	0.1753
も	40	0.0449
する	16	0.0179
、	14	0.0156
会	6	0.0066

# 実験 (2)

- 半教師あり学習: 京大コーパスから10000文の品詞を教師ありデータとして固定、37400文をサンプル

[3 1]

ついて	231	0.2009
OOV	92	0.0838
よって	73	0.0632
とって	64	0.0554
対し	63	0.0545
対して	56	0.0484
より	31	0.0266
して	25	0.0216

[3 1 6]

よる	297	0.5674
対する	97	0.1852
関する	41	0.0781
おける	17	0.0323
基づく	17	0.0323
かかわる	12	0.0227
伴う	10	0.0189
OOV	9	0.0171

# 実験 (2)

- 半教師あり学習: 京大コーパスから10000文の品詞を教師ありデータとして固定、37400文をサンプル

[5 3]

金融	37	0.1494
自由	35	0.1412
可能	35	0.1412
両	34	0.1376
安全	24	0.0962
労働	21	0.0840
民主	20	0.0799
国際	9	0.0348

[5 5]

一	521	0.1091
二	358	0.0750
三	314	0.0658
OOV	245	0.0522
四	189	0.0395
五	143	0.0299
八	118	0.0247
十	117	0.0244

# 実験 (2)

- 半教師あり学習: 京大コーパスから10000文の品詞を教師ありデータとして固定、37400文をサンプル

[11]

これ	293	0.1017
それ	236	0.0822
OOV	124	0.0436
日本	74	0.0253
そこ	42	0.0145
昨年	41	0.0138
米国	38	0.0125
今年	33	0.0111

[11 0 1]

大蔵	35	0.2139
外務	25	0.1526
村山	23	0.1422
通産	13	0.0791
厚生	13	0.0791
運輸	12	0.0730
文部	11	0.0668
警視	9	0.0544

# 実験 (3)

- “未知の言語”：クリンゴン語、Star Trekの宇宙人語
- クリンゴン語「ハムレット」
  - 3733行, 19927語

Qo'noS ta'puq Hamlet lotlut  
lutvaD ghotvam luDalu'  
Qo'noS ta' ghaH  
ben ta' puqloD; DaHjaj ta' loDnI'puqloD je ghaH  
Qang ghaH  
Hamlet jup ghaH  
polonyuS puqloD ghaH  
toy'wl'pu' chaH



# 実験 (3)

[1]			[1 1]		
tugh	48	0.0417	DaH	116	0.1578
*Hamlet*	38	0.0333	vaj	70	0.0957
ta'	32	0.0296	reH	40	0.0546
not	28	0.0243	tugh	26	0.0407
jIHvaD	25	0.0213	jIHvaD	19	0.0236
*polonyuS*	25	0.0199	chIch	16	0.0198
'eH	20	0.0161	yo'	13	0.0169

- 1 = 副詞&呼びかけ?

- tugh=“soon”, DaH=“now”, vaj=“then”

# 実験 (3)

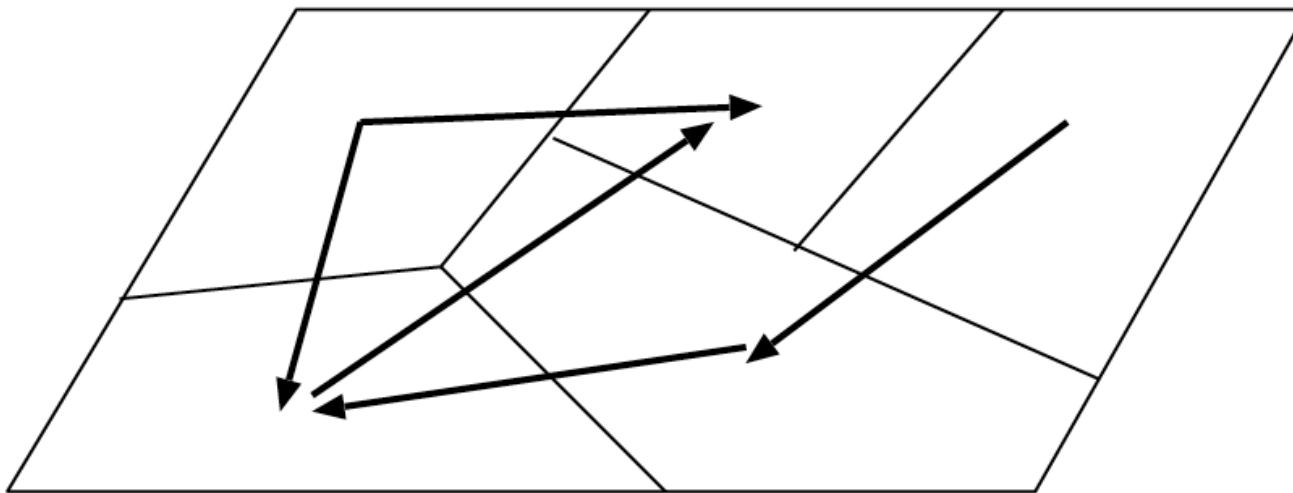
[2 0 0]			[2 1]		
'el	58	0.2703	vaj	70	0.6278
mej	37	0.1764	je	18	0.1493
Ha'	22	0.1018	po'	6	0.0469
joH	17	0.0787	pol	1	0.0016
naDev	11	0.0505	vIDa	1	0.0016
wa'	10	0.0450	ta'be'nal	1	0.0016
Hegh	7	0.0319	jabbI'ID	1	0.0016

- 2 = 動詞?

- 'el="go", mej="leave", vaj="then", Ha'="let's go"

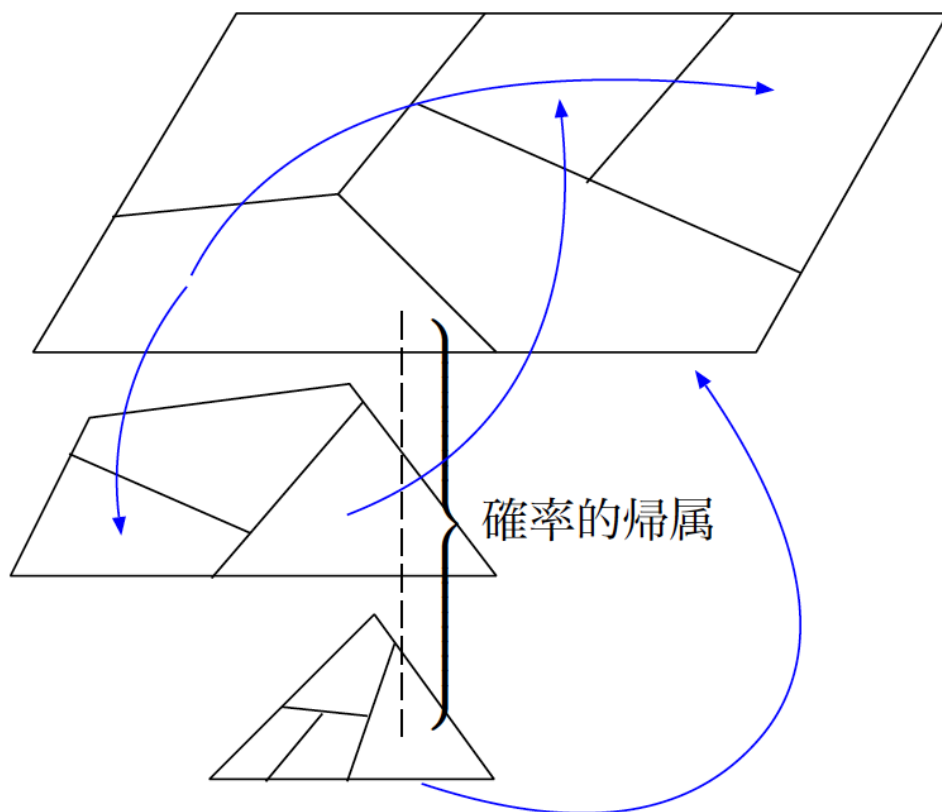
# 測度の空間と分割

- 通常のHMMは、出力確率測度全体の空間を分割して、各クラスタの間の遷移を考えていることと等価



# 再帰的分割とiTHMM

- iTHMMは、状態空間を再帰的に分割して、より細かい遷移を表現
  - カウントの多さに応じた階層ベイズスムージング



# まとめ

---

- 木構造Stick-breaking過程 (Adams+ 2010)を  
それ自体、無限木構造上で階層化した  
階層的木構造Stick-breaking過程を提案  
= Infinite Tree HMM
  - 自然言語処理や品詞推定に限らない、HMMの  
本質的な拡張
- HMMの状態空間の再帰的な分割+ベイズ推定
- 「品詞体系」の教師なし学習が初めて可能に
  - ハイパーパラメータの推定など、学習にはまだ課題  
がある

# 課題

---

- ハイパーパラメータの学習
  - 現状、スライスサンプリングすると0に次第に縮退
  - 尤度自体は0でない方が最終的に高い
- Forward-backward
  - 通常の方法では無理だが、状態はすべて $[0,1)$ の範囲で表せるため、Embedded HMM (Neal 2004)が使える可能性が高い
- 行列式点過程 (Determinantal point process)による、重複した状態の抑制
- トピックモデルへの適用