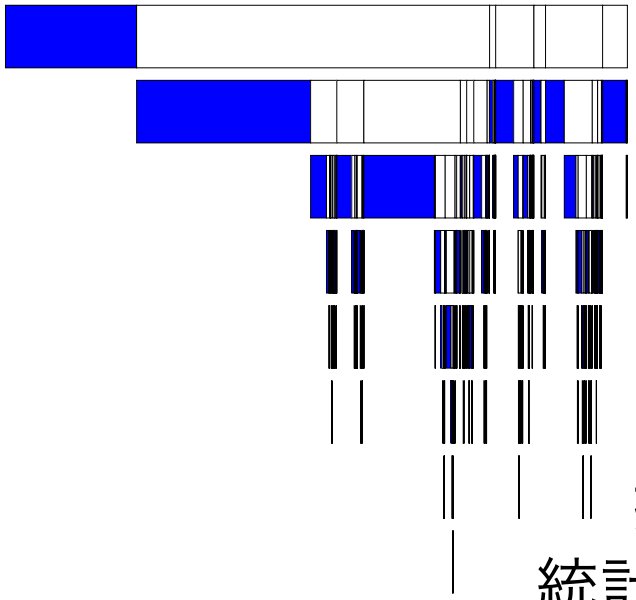
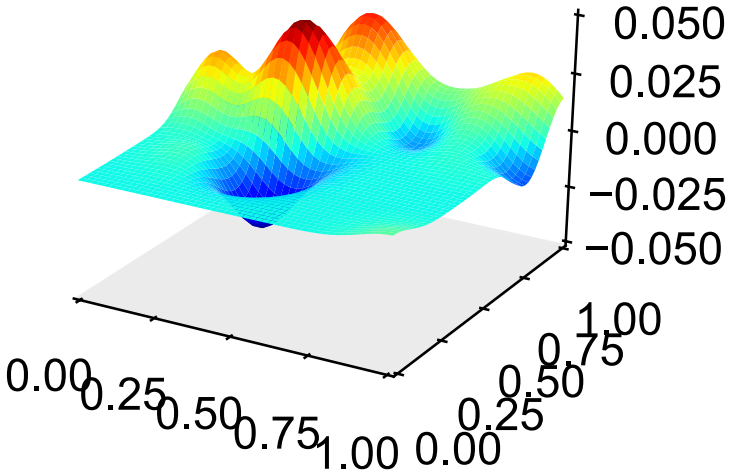
# ノンパラメトリックベイズ統計と自然言語処理






持橋大地
統計数理研究所
daichi@ism.ac.jp

Summer School 数理物理
2020-8-27 (金)

# 自己紹介

- 統計数理研究所 数理・推論研究系 准教授／
  総合研究大学院大学 統計科学専攻
- 電子情報通信学会 IBISML (情報論的学習理論と
  機械学習) 研究会 専門委員 (2016-2021)
- 専門: 自然言語処理 、機械学習 (特に教師なし学習)

立川・統数研

# 自己紹介 (2)

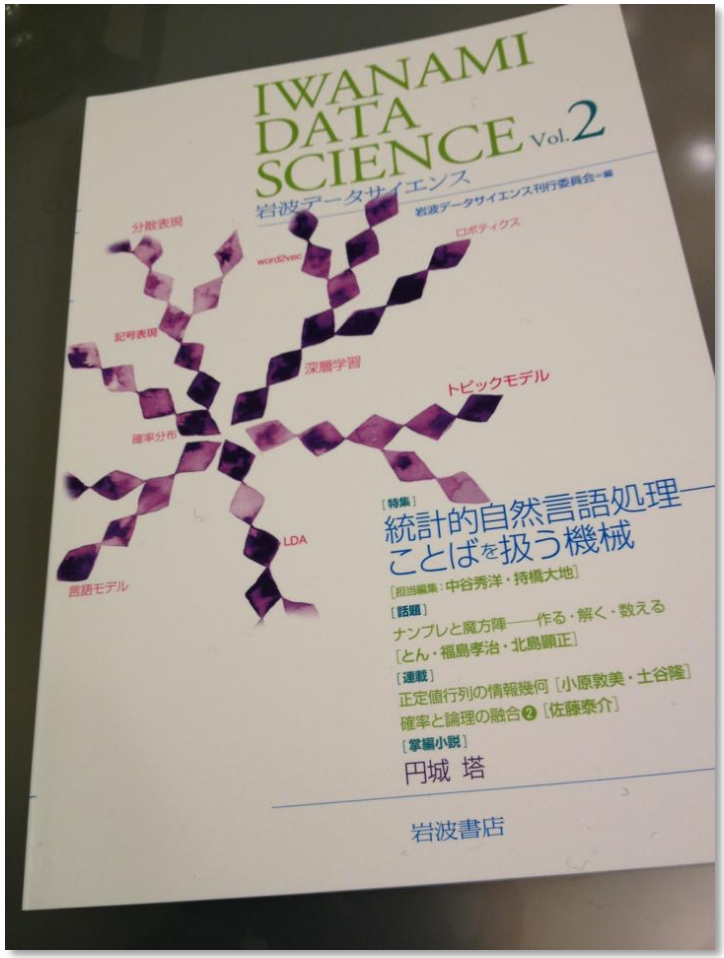
- 文科三類 12組フランス語
  → 教養学部基礎科学科第二 (文科全体から2名の定数外)
  - 現在の広域科学科
- NAIST博士課程→ATR音声研→NTT CS基礎研
  →統数研

# 「ガウス過程と機械学習」


- 講談社機械学習プロフェッショナルシリーズ(MLP), 2019年3月発売
  - 持橋大地・大羽成征著
  - 現在、レビュー49件
- 線形回帰モデルの非常にやさしい導入から入っています
- 確率過程としての話ではなく、統計の道具としての意味と使い方の話

# 岩波データサイエンス

- 岩波データサイエンス2巻：「統計的自然言語処理
  ーことばを扱う機械」
  編集：持橋 (統数研)・
  中谷 (サイボウズラボ)




数解研のご出身

- 特集統括・記事の一部を執筆
- 「統計数理」2016年12月号も自然言語処理の特集です

# 機械学習界の変化と歴史

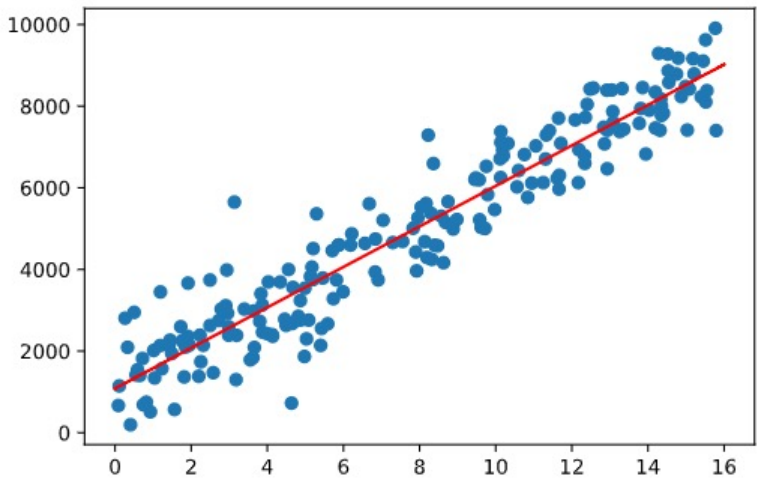
- 「機械学習」が盛んになったのは、Webが普及した2000年代〜
- 2000年代：統計的機械学習のブーム
  - SVM, カーネル法, LDA, …

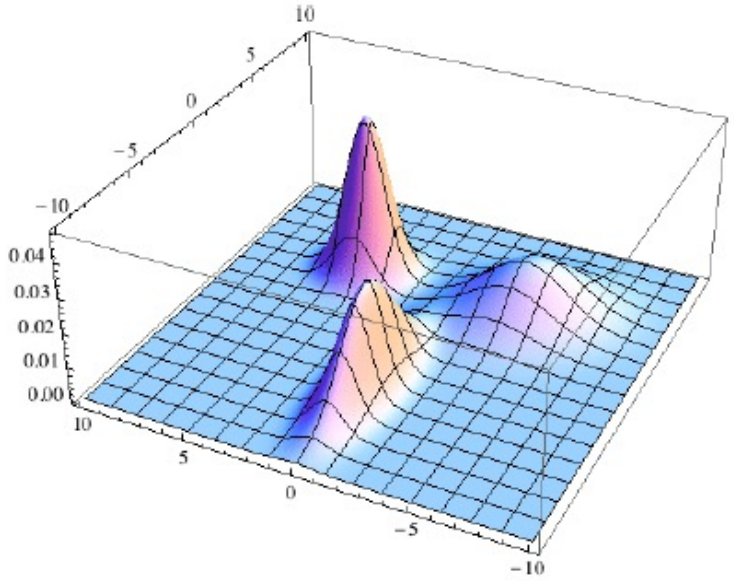
← 2004年 ニューラル言語モデル (Bengio)

- 2010年代〜：深層学習へと移行
  - 2012年 AlexNet (画像処理)
  - 2013年 Word2Vec (自然言語処理)
- 2020年代：両者が融合？ (研究レベルでは多数例あり)

# ノンパラメトリックベイズ統計

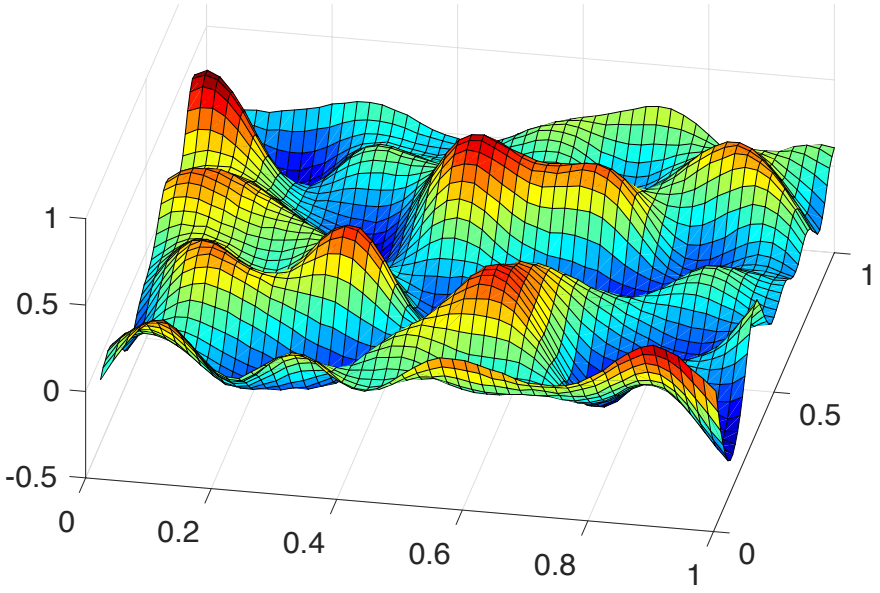
- 「パラメトリック」な統計モデル…
  少数(<データ数)のパラメータで確率分布を表現
  – ガウス分布、多項分布、ガンマ分布やその混合分布

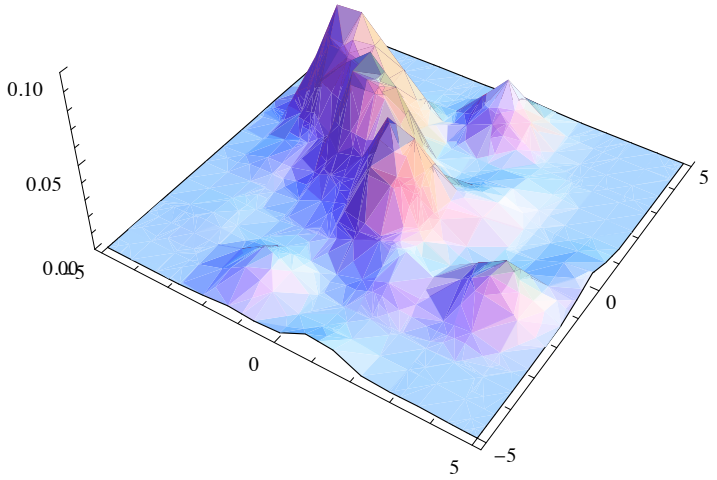


# ノンパラメトリックベイズ統計

- 「ノンパラメトリック」な統計モデル…
  データを直接使った、柔軟な確率分布を実現




  - これをベイズ的に考えたもの→ノンパラメトリックベイズ法
    - 離散的な場合と連続的な場合がある

# 持橋担当分のスケジュール

- 1日目：概要と目次、ノンパラメトリックベイズ法
  (離散的な場合; 無限モデル)
- 2日目：ガウス過程とその適用
  (連続的な場合; ベイズ的関数回帰)
- 3日目：ノンパラメトリックベイズ法と自然言語処理への応用
  (研究紹介)

# 深層学習について

- 自然言語処理での深層学習は、中で何が起きているかほとんど分かっていない ($\leftrightarrow$ 画像処理)
  - 上手く動くこともあるが、変な動きをしないかは保証できない
  - 多数のハイパーパラメータを手作業で調整する必要
- 深層学習の幾つかは、数学的に等価な表現がある
  - Word2Vecは、特別な行列のSVDと等価
  - (多層)ニューラルネットは、(多層)ガウス過程と等価

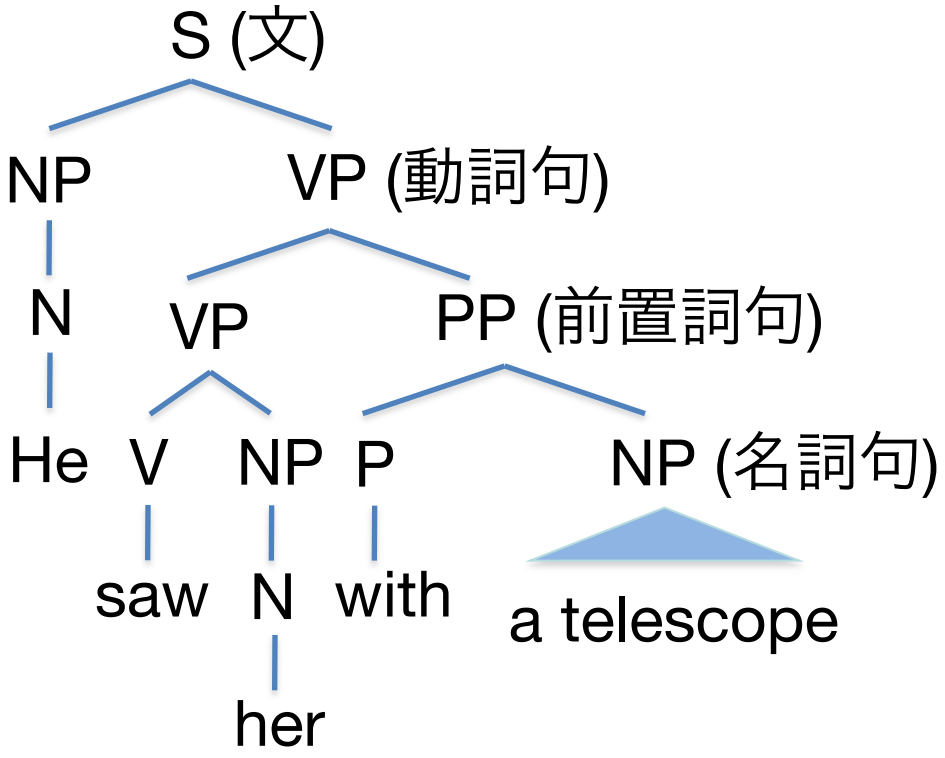
  →数学的な性質が明らか、計算が簡単、
  モデルとして拡張しやすい

# 言語学と統計的自然言語処理

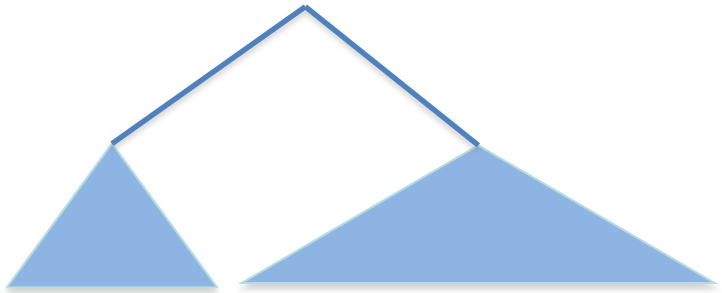
- 言語の研究→言語学科に行けばよいか?
- いわゆる「言語学」＝手で書いたルールの固まり！
- 例：構文解析

S→NP VP
VP→V PP
VP→V NP
PP→P NP
NP→DET N
NP→N

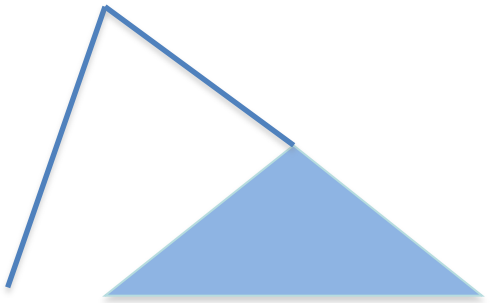
文法ルール



# 言語学と統計的自然言語処理 (2)



He saw her with a telescope

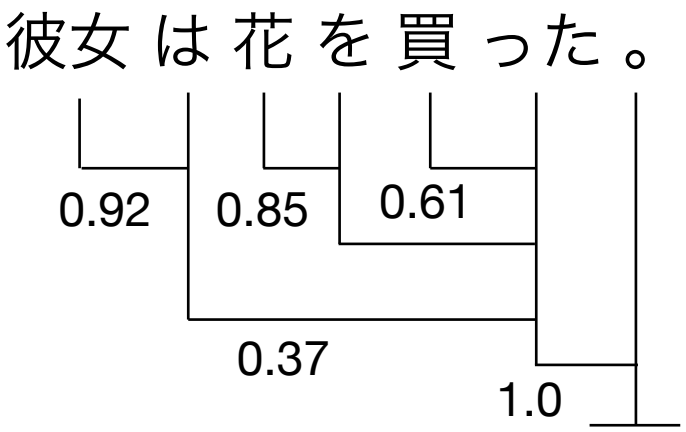


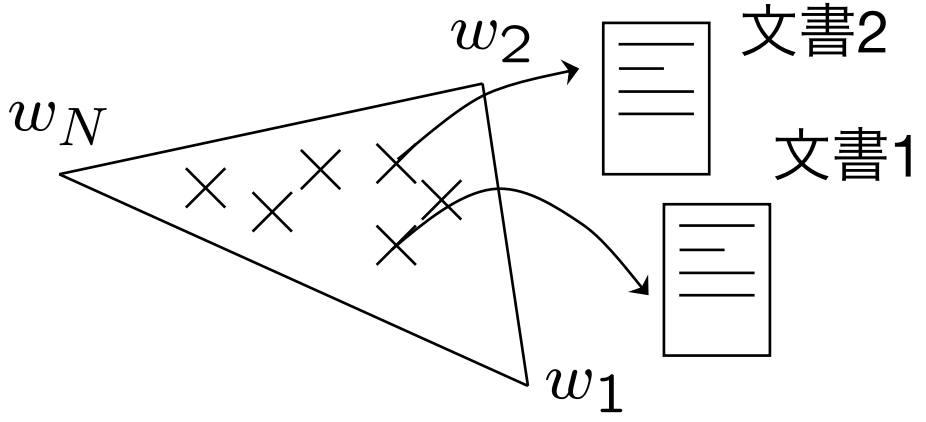
He saw her with a hat

- 解釈が名詞によってなぜ違うのか?
- 古典的な言語学では答えが出せない・そもそも主観的
  → 確率モデル・統計学として数学的に
  考え直す必要
  - cf. 中世の天動説から地動説の物理理論へ

# 統計的自然言語処理

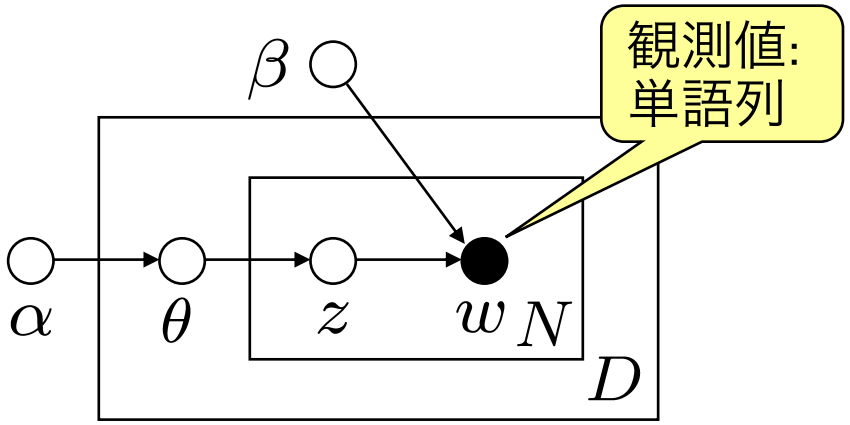
- 構文解析, 文書モデル, 評判分析, 古文書解読, ...




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

ある単語xの品詞
が形容詞である確率



# 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”
- 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?

# 言語の巾乗則

- 自然言語の単語の出現は、巾乗則 (Power law)に従っているといわれている (Zipf則; Zipf 1935)
- 単語wについて、その頻度n(w)は

$$n(w) \propto r^{-\alpha}$$

( $r$：頻度順の順位)

- よって

$$\log n(w) \propto -\alpha \log r$$

– log n(w)とlog rが右下がりの直線関係 $(\alpha \approx 1)$

# 京大コーパスの統計

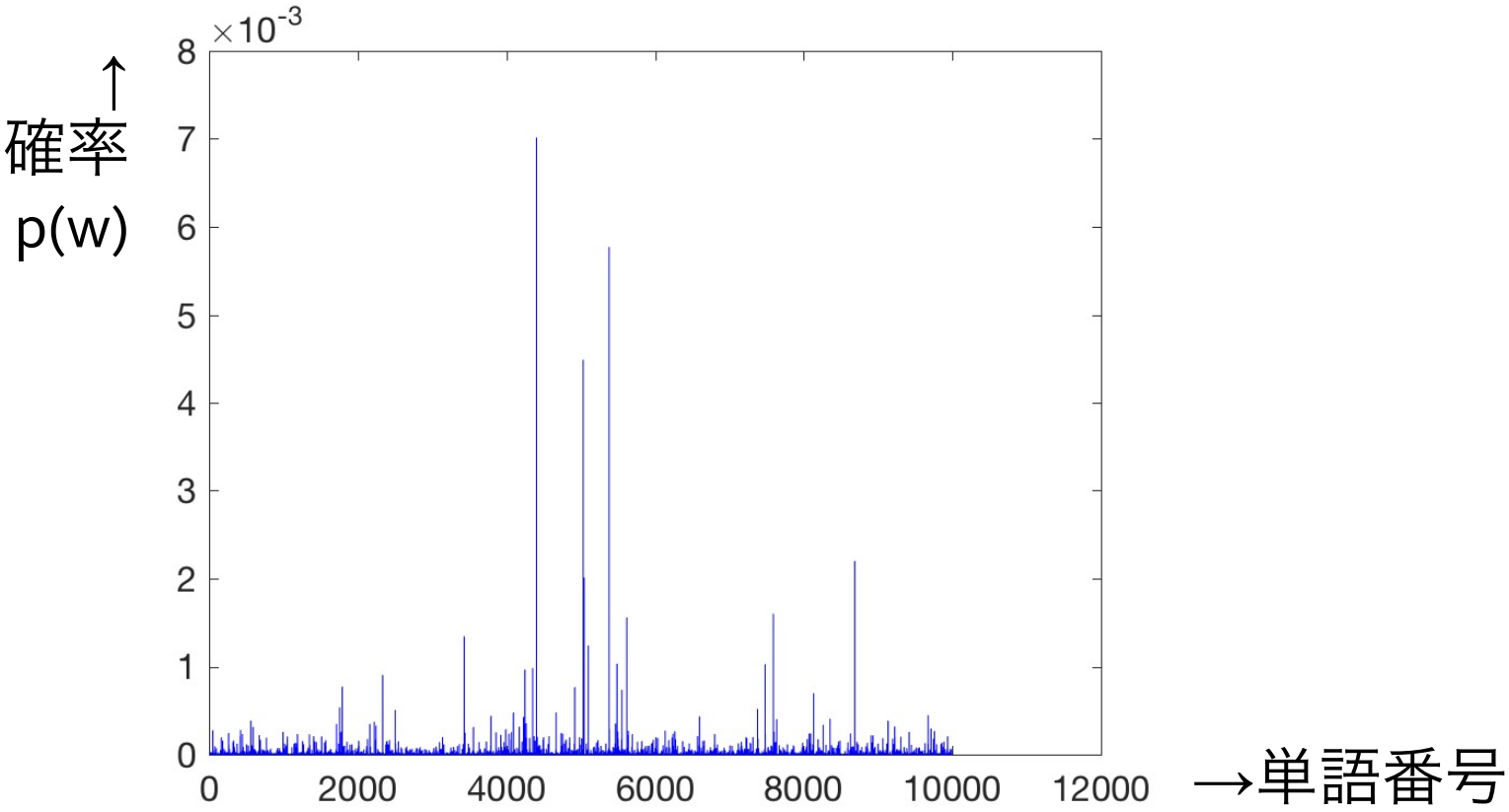
- 毎日新聞1995年度のテキスト約4万記事に、形態素解析(単語分割・品詞付与・その他)を行ったテキストデータ

後 に 昭和 に 入って 、 反軍 演説 で 名 を 馳せた 憲政 会 の 斎藤 隆夫 が 、 日記 に こう 記した の は 、 いわゆる 男子 普通 選挙 法 が 成立 した 一九二五 年 三 月 二十九 日 の こと であった 。
それ まで 総 人口 の わずか 二 % 程度 だった 有権者 数 が 、 一挙 に 二〇 % に 拡大 さ れ 、 確かに 、 選挙 制度 と して 画期的な もの で は あった 。
「 我國 の 政界 に 新 時代 を 畫す べき 當日 の 兩院 の 傍聽 席 に は 流石 に 熱心な 聽衆 の 緊張 した 顔 が 幾重にも ぎっしり と 重なり合ひ ..... 」
毎日新聞 の 前身 、 「 東京 日日新聞 」 は 同年 三 月 三十 日 付 朝刊 で 成立 の 様子 を こう 伝えて いる 。
：

# 単語の確率

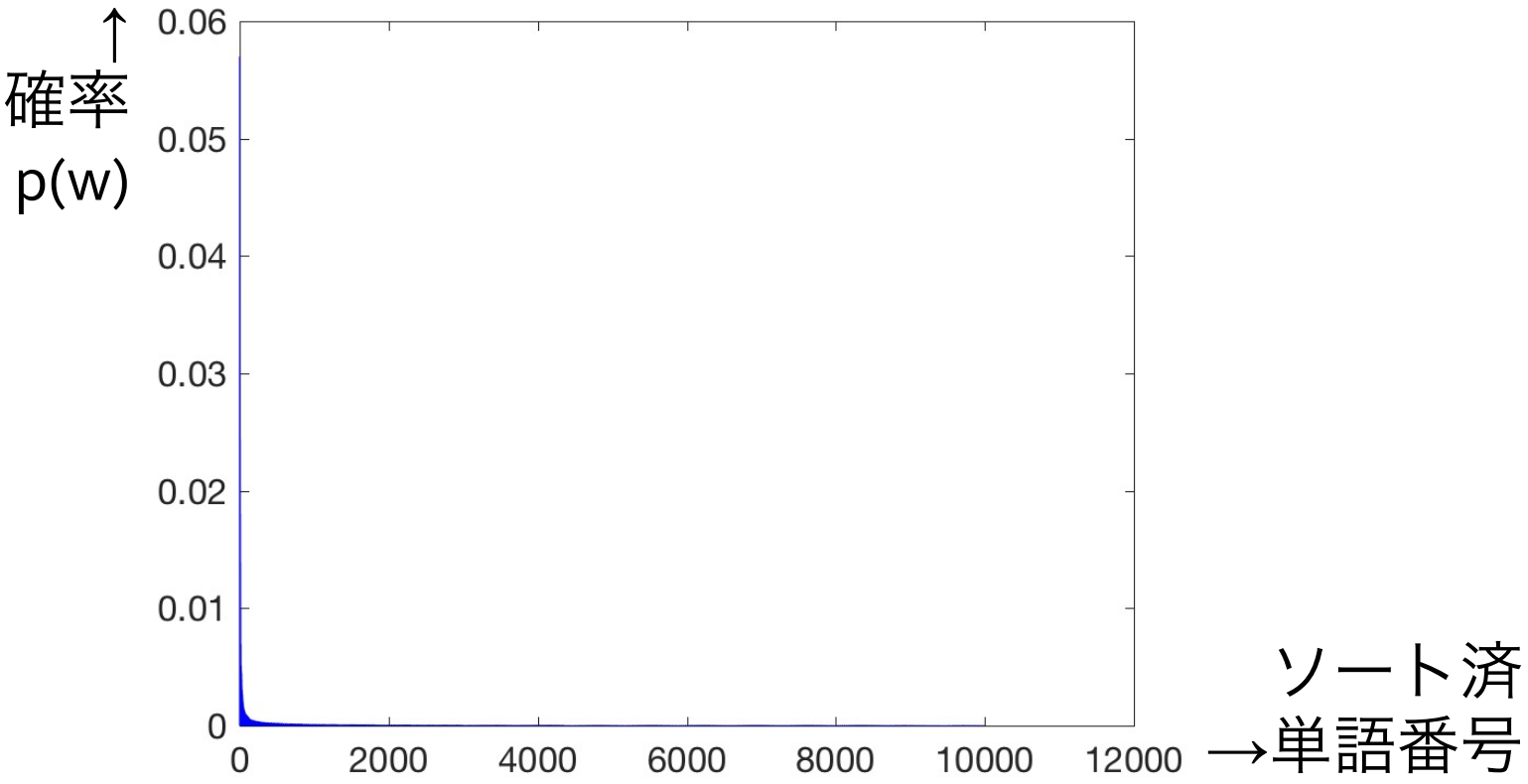
- 観測した全単語数を$N$とすると、単語wの確率は

$$p(w) = \frac{n(w)}{N}$$



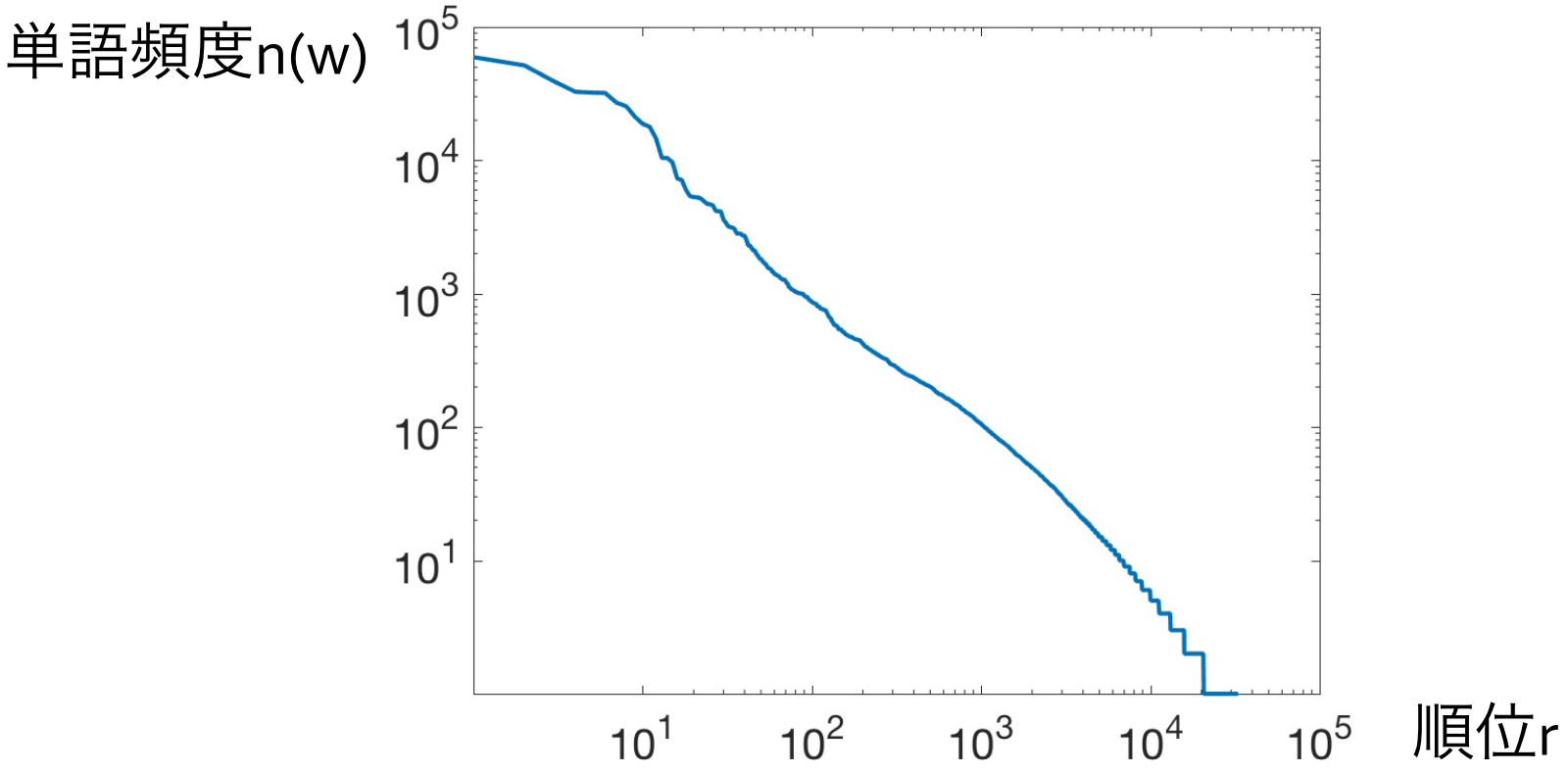
# 単語の確率

- 単語確率 p(w) の順(=頻度順)にソートしてみる



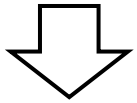
# 京大コーパスの統計

- 京大コーパスでの確率を両対数プロットにしたもの



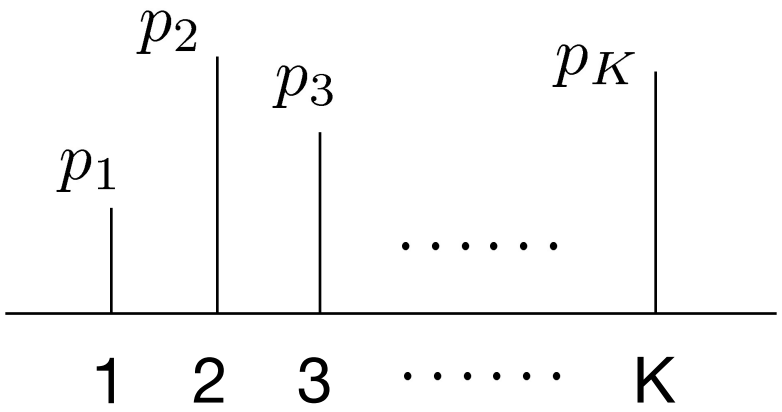
# 言語の巾乗則 (2)

- どうして、こんな巾乗分布が生まれるのか?
- ‘Rich gets richer’の現象



- 回答(の一つ)：Dirichlet過程、Pitman-Yor過程
  - 数学(確率論)では、Gibbs partition, Ewens公式などとして詳しく研究されている

# 準備: 多項分布 (離散分布)

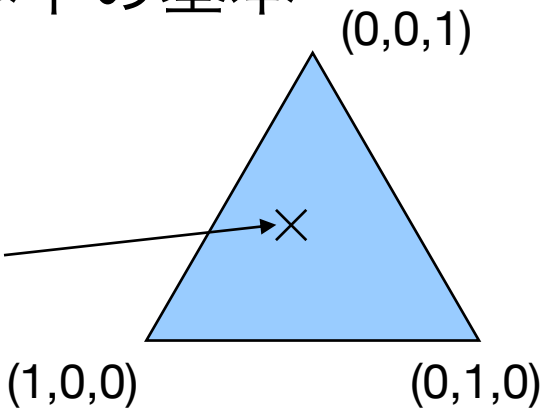



$$\sum_{k=1}^{K} p_k = 1,$$

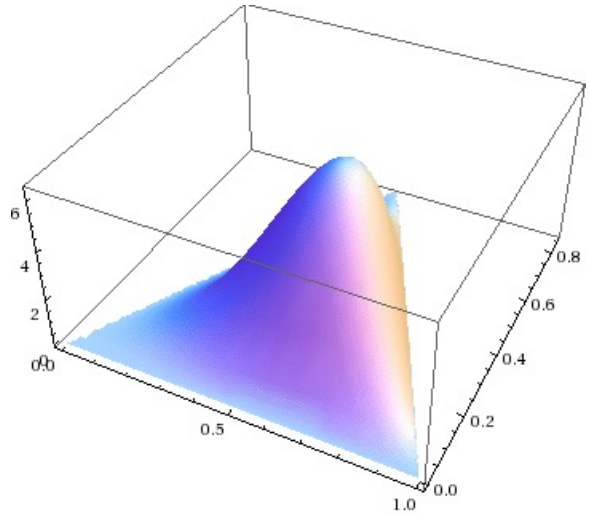
$$\forall k,\ p_k \geq 0$$

- K種類のアイテ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

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



# ディリクレ分布



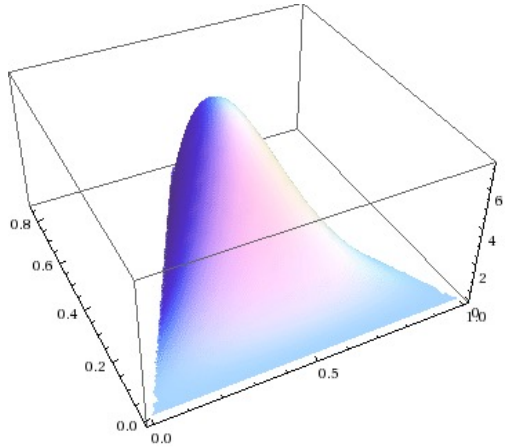

$$\mathrm{Dir}(\mathbf{p}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} p_k^{\alpha_k - 1}$$

パラメータ:

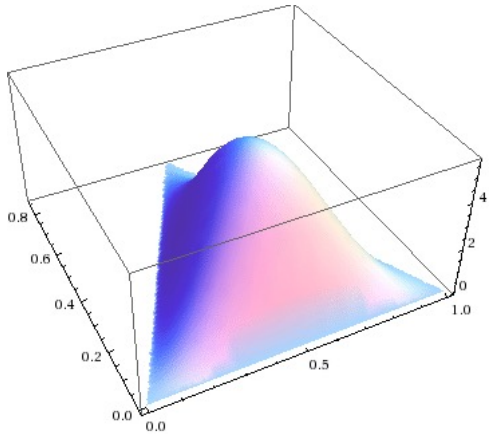
$$\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_K)$$

- 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」： $\bar{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_K) / \sum_k \alpha_k$
  「分散」　： $\alpha = \sum_k \alpha_k$

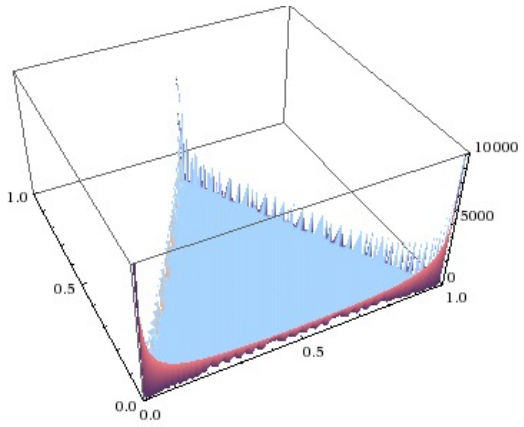
# ディリクレ分布 (2)




$\boldsymbol{\alpha} = (2, 4, 2)$




$\boldsymbol{\alpha} = (2, 2, 2)$

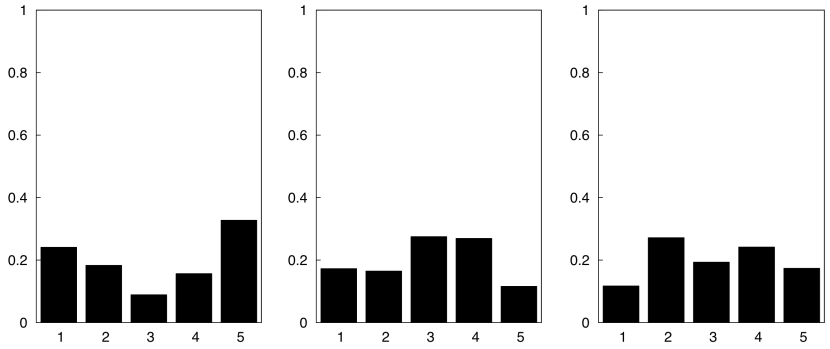



$\boldsymbol{\alpha} = (0.5, 0.5, 0.5)$

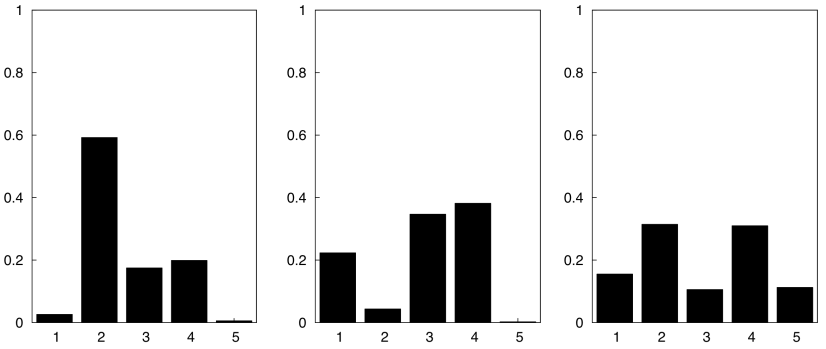
- $\alpha_k > 1$のとき、上に凸
- $\alpha_k < 1$のとき、下に凸
  - 統計的自然言語処理等では、多くの場合 $\alpha \ll 1$ ( $\alpha = 0.1 \sim 0.0001$くらい)

# ディリクレ分布 (3)

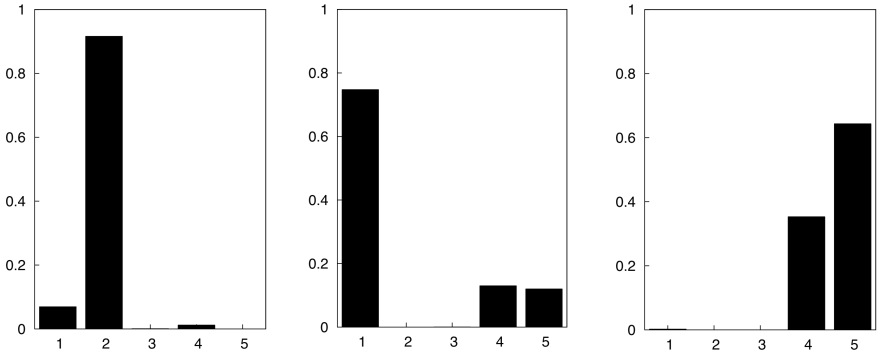
- ディリクレ分布からのサンプル $\mathbf{p}$



$$\boldsymbol{\alpha} = (10, 10, \cdots, 10)$$



$$\boldsymbol{\alpha} = (1, 1, \cdots, 1)$$



$$\boldsymbol{\alpha} = (0.1, 0.1, \cdots, 0.1)$$

# 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: $p(\mathbf{p}|X) \propto p(X|\mathbf{p})p(\mathbf{p})$

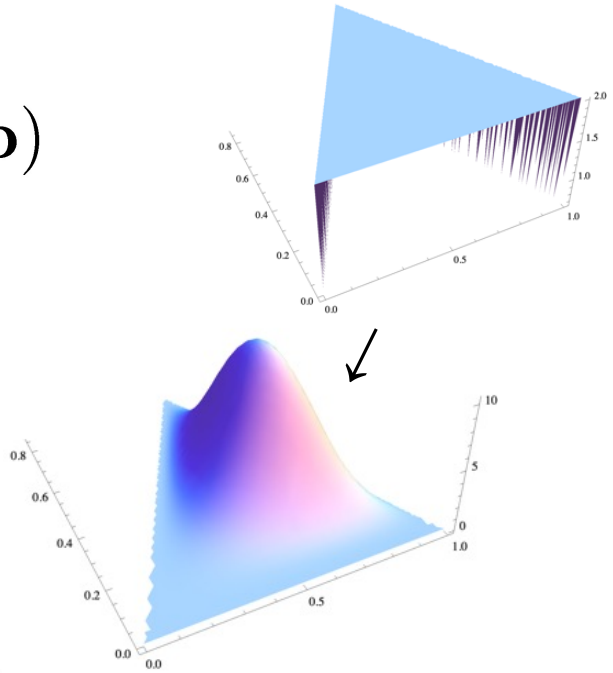
$$\propto \left(p_1^1 \cdot p_2^3 \cdot p_3^2\right) \cdot \left(\prod_{k=1}^3 p_k^{\alpha_k - 1}\right)$$

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

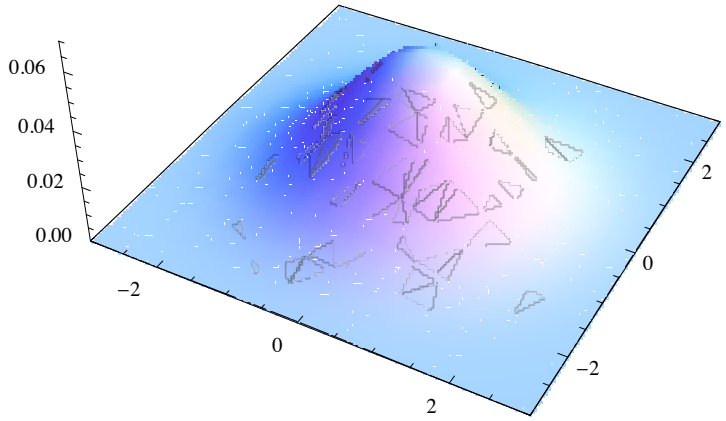
$$= \mathrm{Dir}(\alpha_1+1, \alpha_2+3, \alpha_3+2)$$

- $\mathbf{p}$ の期待値は、

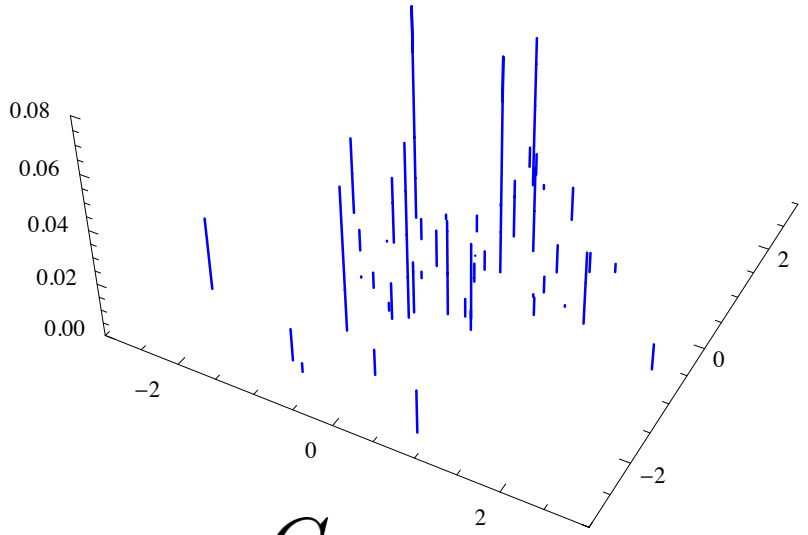
$$E[\mathbf{p}|X] = \left(\frac{\alpha_1+1}{\alpha+6}, \frac{\alpha_2+3}{\alpha+6}, \frac{\alpha_3+2}{\alpha+6}\right) \quad (\alpha = \textstyle\sum_k \alpha_k)$$



# 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

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

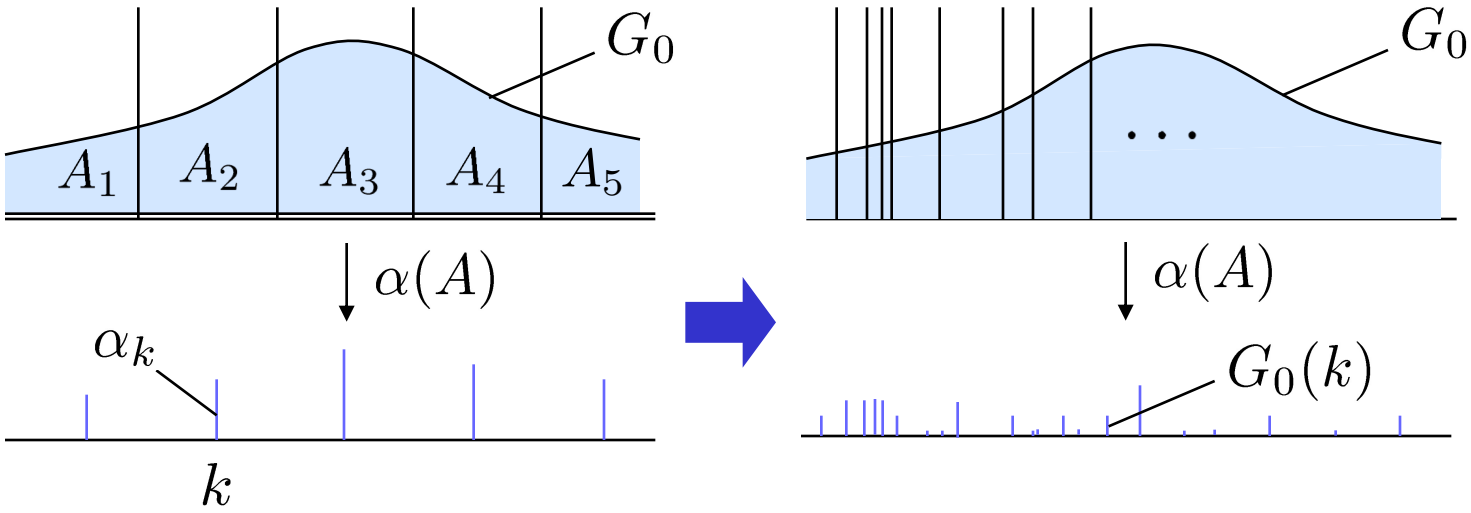
# ディリクレ過程 (2)

- 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 $(\mathscr{X}, \mathscr{A})$ with parameter $\alpha$ if for any measurable partition $(A_1, \cdots, A_k)$ of $\mathscr{X}$, the random vector $(P(A_1), \cdots, P(A_k))$ has a Dirichlet distribution with parameter $(\alpha(A_1), \cdots, \alpha(A_k))$ .

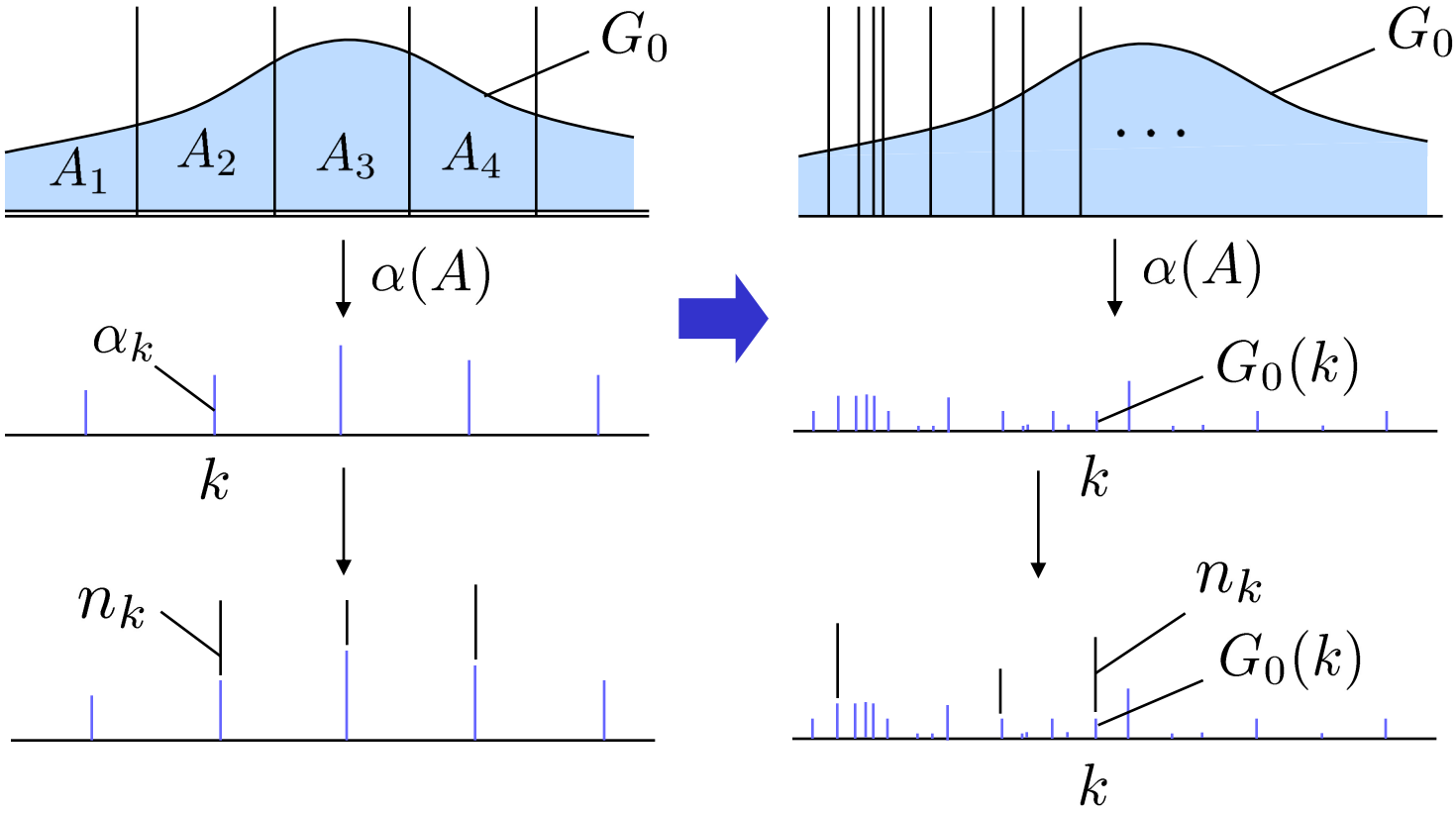
- つまり..

# 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

# ディリクレ過程 (4)




予測確率:

$$\frac{\alpha_k + n_k}{\alpha + n} \qquad \frac{\alpha G_0(k) + n_k}{n + \alpha}$$

# ディリクレ過程による予測

- 無限次元多項分布 $G$ を積分消去することで、予測ルールが得られる

$$p(x|x_1,\cdots,x_n) = \int p(x|G)p(G|x_1,\cdots,x_n)dG$$

$$= \frac{n_k}{\alpha+n}\delta(x=X_k) + \frac{\alpha}{\alpha+n}G_0(x)$$

前に出たもの

新しいもの

- つまり、

$$G\,|\,x_1,\cdots,x_n \sim \mathrm{DP}(\alpha + \textstyle\sum_{i=1}^{n}\delta(x_i))$$

# ディリクレ過程による予測 (証明)

$$G \,|\, x_1, \cdots, x_n \sim \mathrm{DP}(\alpha + \textstyle\sum_{i=1}^{n} \delta(x_i))$$

- $n$=1のとき示せば充分なので、
  $G \sim \mathrm{DP}(\alpha),\ X \sim G$ のとき、
  $$G \,|\, X \sim \mathrm{DP}(\alpha + \delta(X))$$
  を示す.

- 考えている空間$\mathcal{X}$の可測な分割 $A_1, \cdots, A_k$ について
  $$(G(A_1), \cdots, G(A_k)) \,|\, X$$
  $$\sim \mathrm{Dir}(\alpha(A_1) + \delta_X(A_1), \cdots, \alpha(A_k) + \delta_X(A_k))$$
  を示せばよい.

# ディリクレ過程による予測 (2)

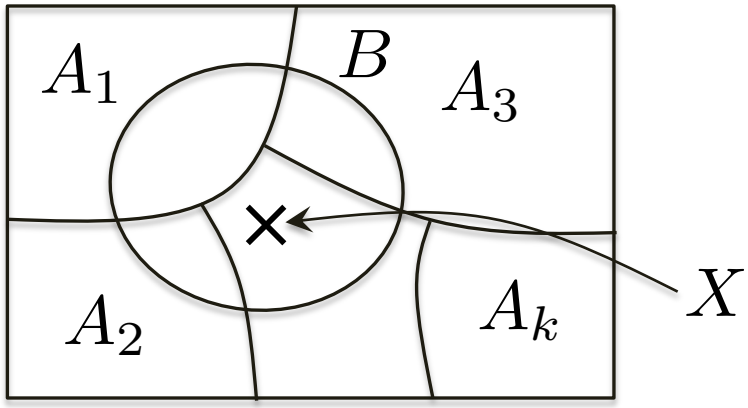
- 考えている $X$ の含まれる集合を $B$ とおくと

$$p(X \in B, G(A_1) \leq y_1, \cdots, G(A_k) \leq y_k)$$
$$= \mathbb{E}_{X \sim G}[D(y_1, \cdots, y_k | \alpha(A_1) + \delta_X(A_1), \cdots, \alpha(A_k) + \delta_X(A_k)), \{X \in B\}]$$

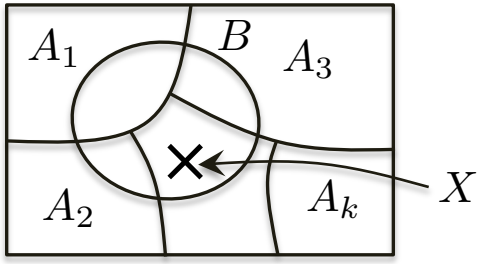
を示せばよい.

- ここで*D*はディリクレ分布の分布関数で

$$D(y_1, \cdots, y_k | \alpha_1, \cdots, \alpha_k)$$
$$= p(Y_1 \leq y_1, \cdots, Y_k \leq y_k),$$
$$Y \sim \mathrm{Dir}(\alpha_1, \cdots, \alpha_k)$$



# ディリクレ過程による予測 (3)



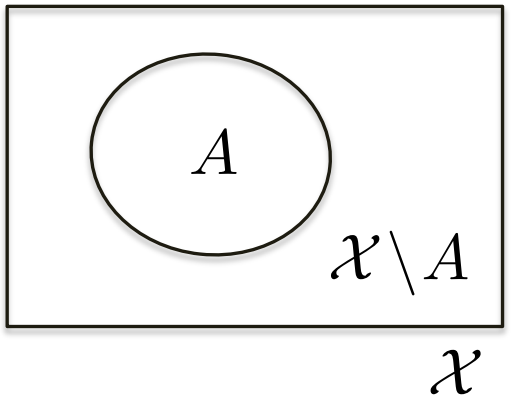

$$p(X \in B, G(A_1) \le y_1, \cdots, G(A_k) \le y_k)$$
$$= \mathbb{E}_{X \sim G}[D(y_1, \cdots, y_k | \alpha(A_1) + \delta_X(A_1), \cdots, \alpha(A_k) + \delta_X(A_k)), \{X \in B\}]$$

$$\alpha_i^{(j)} = \begin{cases} \alpha_i & (i \neq j) \\ \alpha_i + 1 & (i = j) \end{cases}$$

- の右辺を計算すると、

$$\int_B D(y_1, \cdots, y_k | \alpha(A_1) + \delta_X(A_1), \cdots, \alpha(A_k) + \delta_X(A_k))\, d\frac{\alpha(x)}{\alpha(\mathcal{X})}$$
$$= \sum_{j=1}^{k} \int_{A_j \cap B} D(y_1, \cdots, y_k | \alpha_1^{(j)}, \cdots, \alpha_k^{(j)})\, d\frac{\alpha(x)}{\alpha(\mathcal{X})}$$
$$= \sum_{j=1}^{k} \frac{\alpha(A_j \cap B)}{\alpha(\mathcal{X})} D(y_1, \cdots, y_k | \alpha_1^{(j)}, \cdots, \alpha_k^{(j)}) = \text{左辺.}$$

# 補題1




- $G \sim \mathrm{DP}(\alpha)$ のとき、$\forall A \in \mathcal{A}$ について

$$\mathbb{E}[G(A)] = \alpha(A)/\alpha(\mathcal{X}).$$

- Proof:
  ディリクレ過程の定義より、$A_1 = A,\ A_2 = \mathcal{X}\backslash A$
  ととれば、ディリクレ分布の性質から

$$(G(A_1), G(A_2)) = (G(A), G(\mathcal{X}\backslash A)) \sim \mathrm{Be}(\alpha(A), \alpha(\mathcal{X}\backslash A))$$

  よって、

$$\mathbb{E}[G(A)] = \frac{\alpha(A)}{\alpha(A) + \alpha(\mathcal{X}\backslash A)} = \frac{\alpha(A)}{\alpha(\mathcal{X})}.$$

# 補題2

- $G \sim \mathrm{DP}(\alpha),\ X \sim G$ とすると、

$$\forall A \in \mathcal{A},\ p(X \in A) = \alpha(A)/\alpha(\mathcal{X}).$$

- Proof:
  定義より $p(X \in A \,|\, G) = G(A)$ なので、

$$p(X \in A) = \mathbb{E}_G[p(X \in A \,|\, G)] = \mathbb{E}_G[G(A)] = \frac{\alpha(A)}{\alpha(\mathcal{X})}.$$

# 補題3

- $G \sim \mathrm{DP}(\alpha),\ X \sim G$ とする. このとき、
  $\forall B \in \mathcal{A},\ \forall \{A_1, \cdots, A_k\}$ について

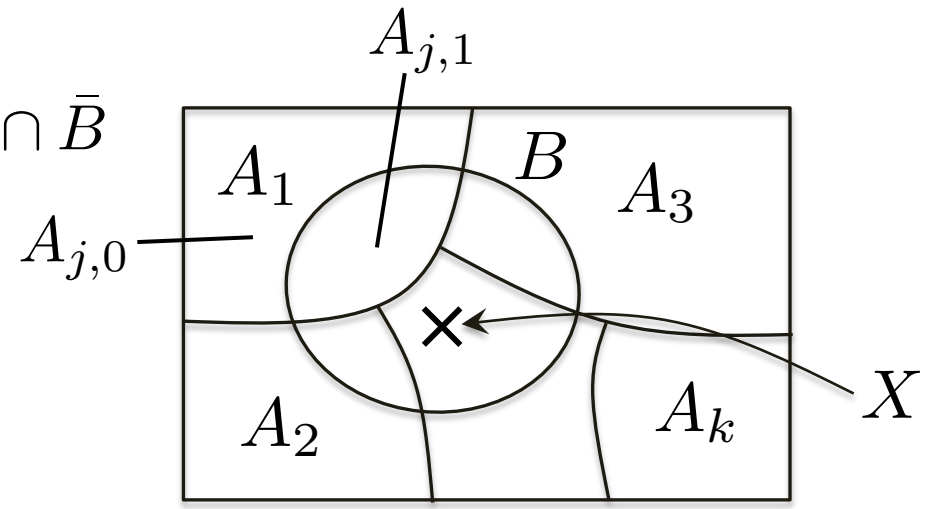
$$p(X \in B, G(A_1) \le y_1, \cdots, G(A_k) \le y_k)$$
$$= \sum_{j=1}^{k} \frac{\alpha(A_j \cap B)}{\alpha(\mathcal{X})} D(y_1, \cdots, y_k \,|\, \alpha_1^{(j)}, \cdots, \alpha_k^{(j)}) .$$

- Proof:

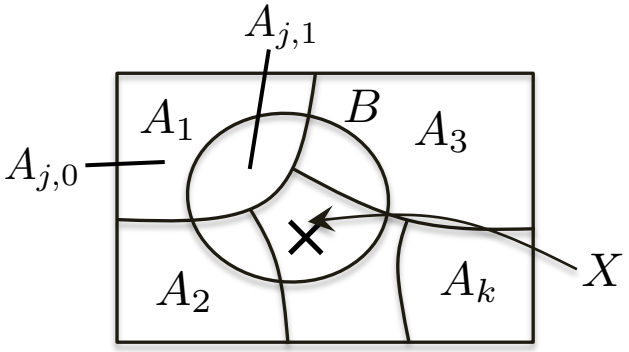
  $A_{j,1} = A_j \cap B,\ A_{j,0} = A_j \cap \bar{B}$
  とおく.

$$Y_{j,v} = G(A_{j,v})$$

  と表せば、



# 補題3 (2)




- 定義から

$$p(X \in B \,|\, Y_{j,v}; j = 1, \cdots, k, j = 0, 1)$$
$$= \sum_{j=1}^{k} G(X \in B_{j,1} | Y_{j,v}) = \sum_{j=1}^{k} Y_{j,1}$$

- よって

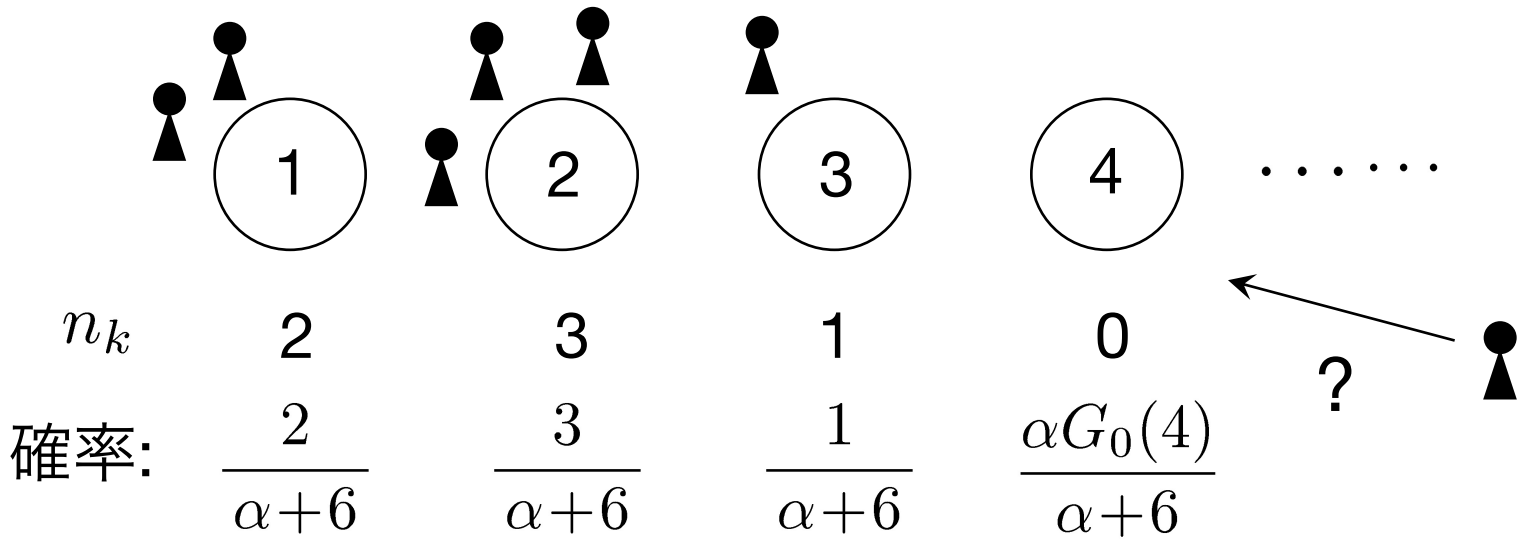
$$p(X \in B, Y_{j,v} \leq y_{j,v}\,;\, j = 1, \cdots, k, v = 0, 1)$$
$$= \mathbb{E}\left[\sum_{j=1}^{k} Y_{j,1}, \{Y_{j,v} \leq y_{j,v}\,;\, j = 1, \cdots, k, v = 0, 1\}\right]$$
$$= \sum_{j=1}^{k} \frac{\alpha(A_{j,1})}{\alpha(\mathcal{X})} D(\underline{y_{1,0}}, \cdots, y_{k,0}, \underline{y_{1,1}}, \cdots, y_{k,1}))$$
$$= \sum_{j=1}^{k} \frac{\alpha(A_j \cap B)}{\alpha(\mathcal{X})} D(y_1, \cdots, y_k \,|\, \alpha_1^{(j)}, \cdots, \alpha_k^{(j)})\,. \quad \square$$

# Chinese Restaurant Process (CRP)

- 予測確率

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

  - 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

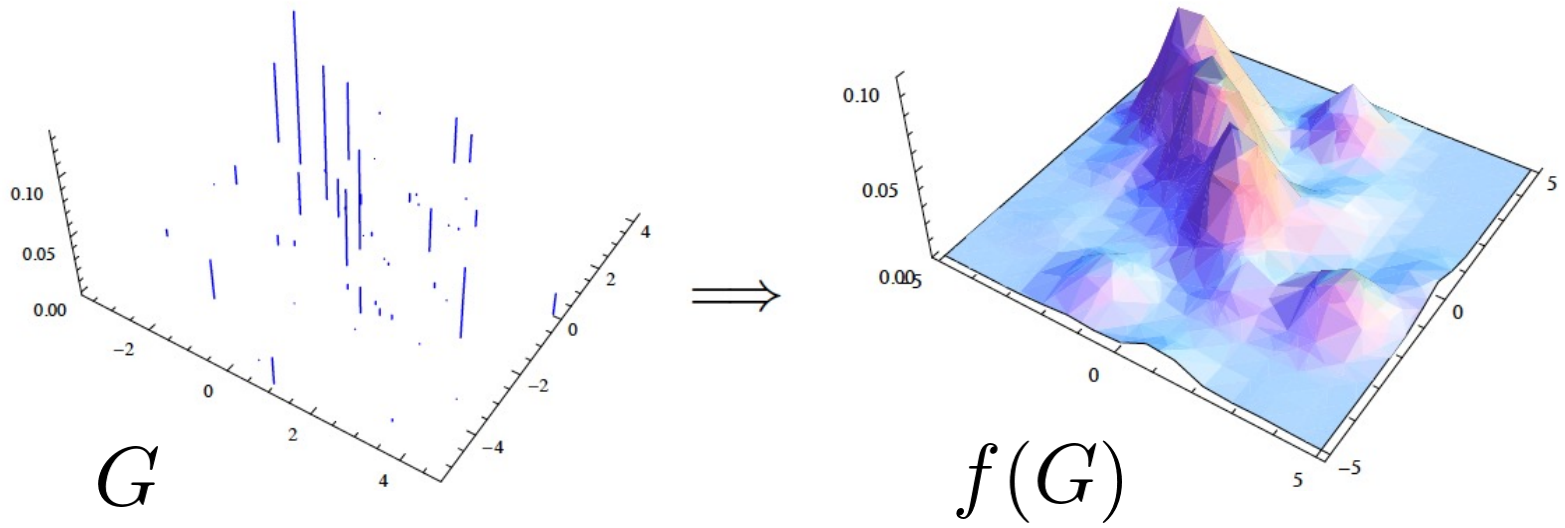
$$p(w|X) = \frac{n(w)}{\alpha + n} + \frac{\alpha}{\alpha + n} G_0(w)$$

  - カウント$n(w)$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、クラスタ数を決めない無限混合モデルが可能になる

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



# ディリクレ過程混合モデル (2)

- MCMC法による学習: 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、

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

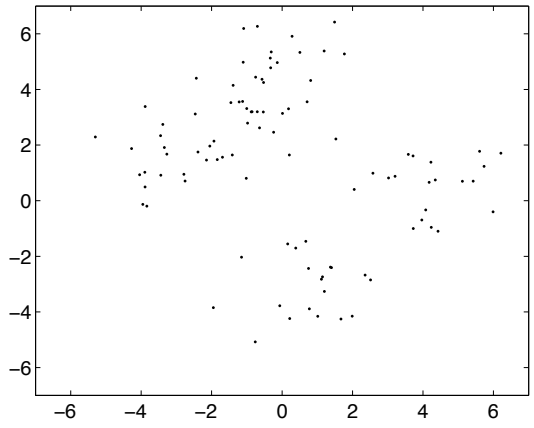
# ディリクレ過程混合モデル (3)

- Gibbs samplerによる学習 (物理では熱浴法)

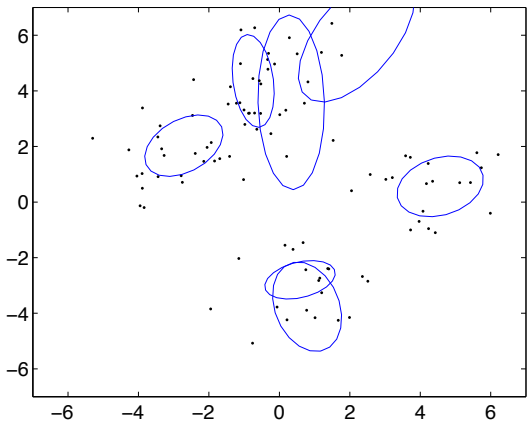
1: **while** not converged **do**
2: **for** $n$ in randperm $(1, \cdots, N)$ **do**
3: $x_n$ をクラスタ $z_n$ から削除してパラメータを更新
4: $z_n \sim p(z_n|X, Z_{-i})$ をサンプル
5: $x_n$ をクラスタ $z_n$ に追加してパラメータを更新
6: **end for**
7: **end while**
8: $z_1, \cdots, z_N$ を出力

  - randperm(): 引数のランダムな並び替えを返す関数

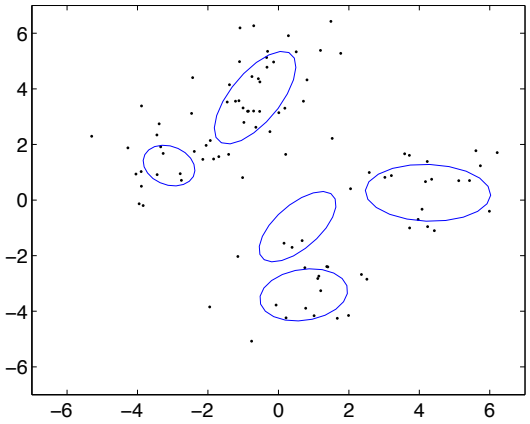
# ディリクレ過程混合モデル (4)



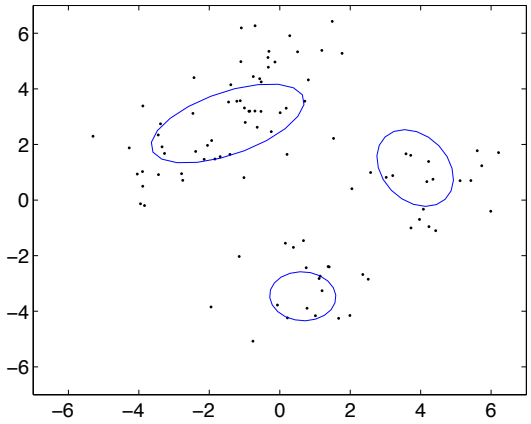
(a) データ



(b) 繰り返し数=10



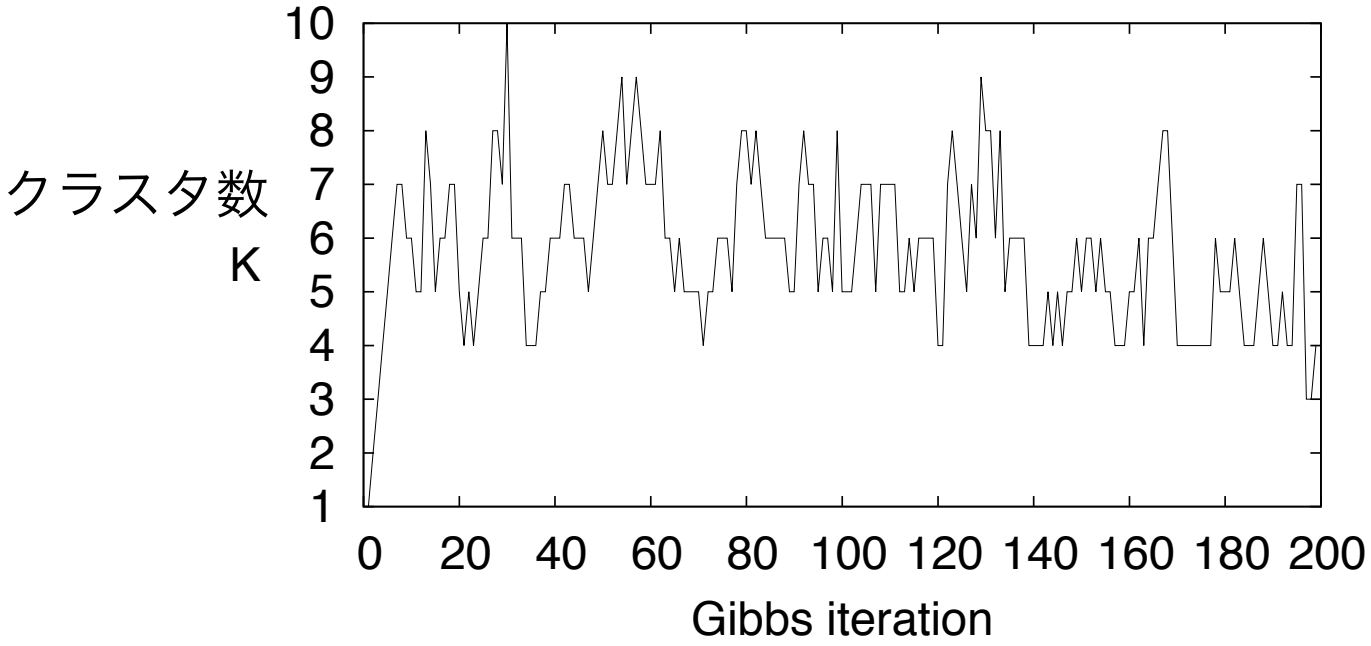
(c) 繰り返し数=100



(d) 繰り返し数=200

- データから自動的にクラスタ数を決定できる
- 対数尤度が収束すれば、学習は終了

# ディリクレ過程混合モデル (5)




- Gibbsの繰り返し毎に、クラスタ数がだんだん収束
  - データがもっと多いと、収束はかなり顕著
  - クラスタ数の推定には、それまで多数のヒューリスティックが提案されていた

# Pitman-Yor過程 (Pitman and Yor 1997)

- Dirichlet過程で、新しいカテゴリが作られる確率は $\frac{\alpha}{n+\alpha} G_0(w)$

  → nが大きくなると、新しいカテゴリを作ることはどんどん難しくなる
- 現実の単語確率分布をより忠実に反映するにはどうすればよいか?

# Pitman-Yor過程 (2)

- 新しいカテゴリの出る確率を、これまでに現れたカテゴリ数に依存させる

$$\begin{cases} p(k \leq K) & \propto \dfrac{n_k - d}{\alpha + n} \\ p(k = K+1) & \propto \dfrac{\alpha + dK}{\alpha + n} \end{cases}$$

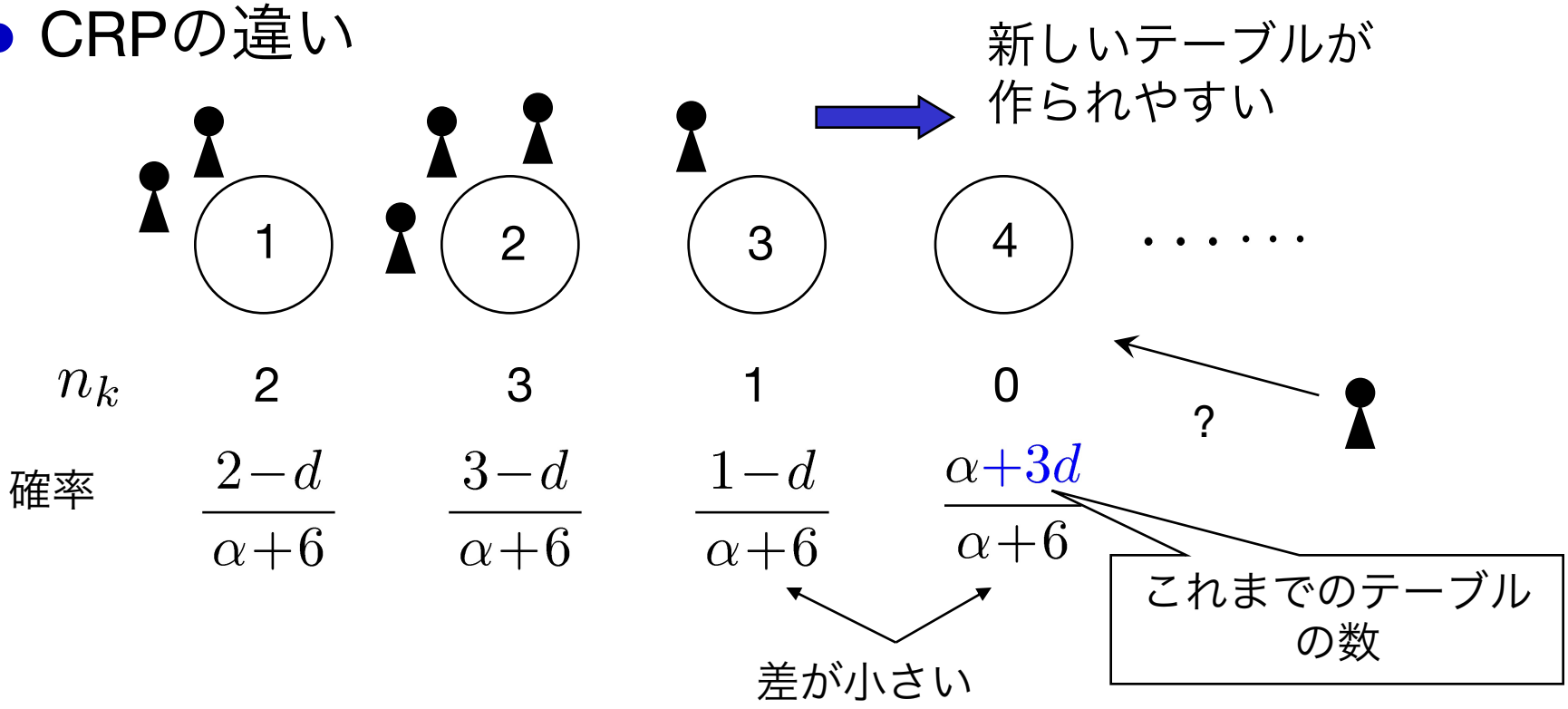
- CRP形式で書くと、

$$p(x|\mathbf{x}, \mathrm{PD}(d, \alpha)) = \sum_{k=1}^{K} \frac{n_k - d}{n + \alpha} \delta_{X_k} + \frac{\alpha + dK}{n + \alpha} G_0(x)$$

($K$：これまでに出たテーブルの数)

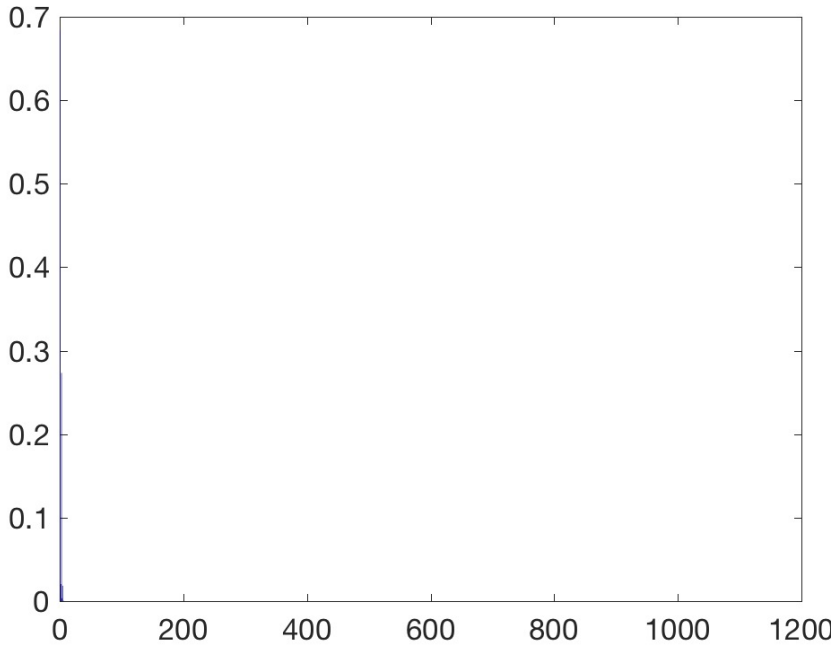
# Pitman-Yor過程 (3)

- Pitman-Yor過程 (Pitman and Yor 1997): PY(α,d)
  - ディリクレ過程の拡張, Poisson-Dirichlet過程とも
  - 新たにディスカウント係数dを持つ
- CRPの違い

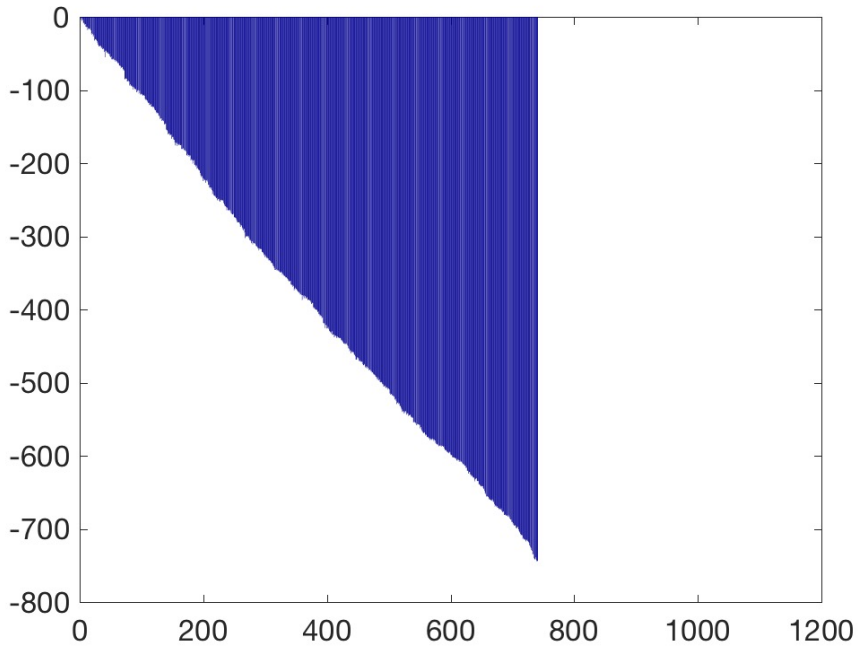


# ディリクレ過程から生成される離散分布

- SBP(1)からのサンプル (最初の1000次元)



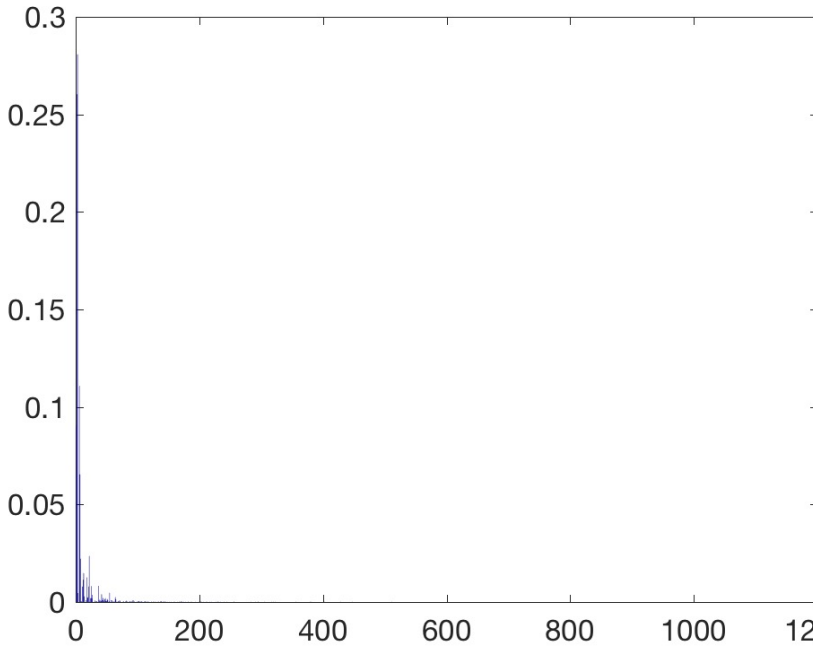

確率分布p



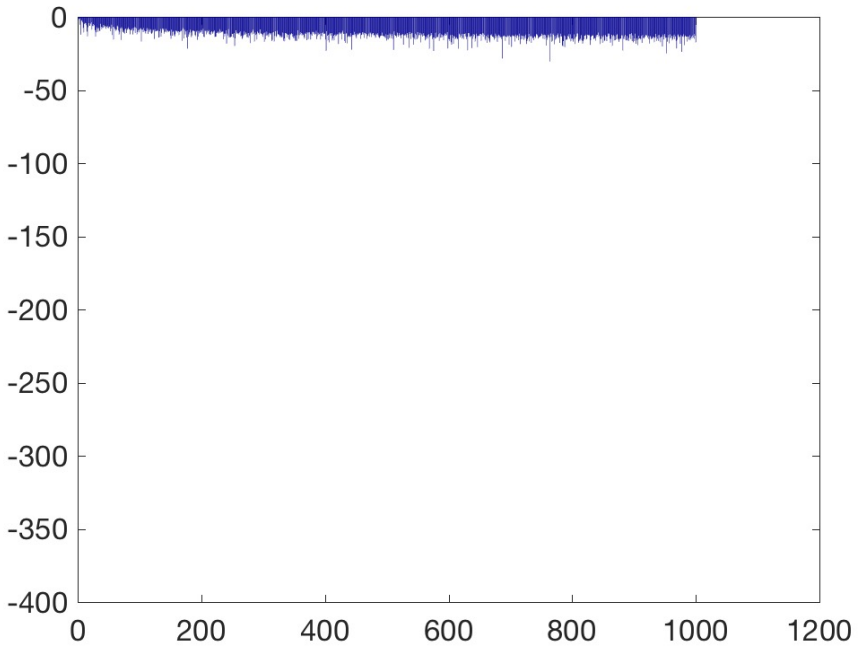

対数 log(p)

# Pitman-Yor過程から生成される離散分布

- PY(1,0.5)からのサンプル (最初の1000次元)



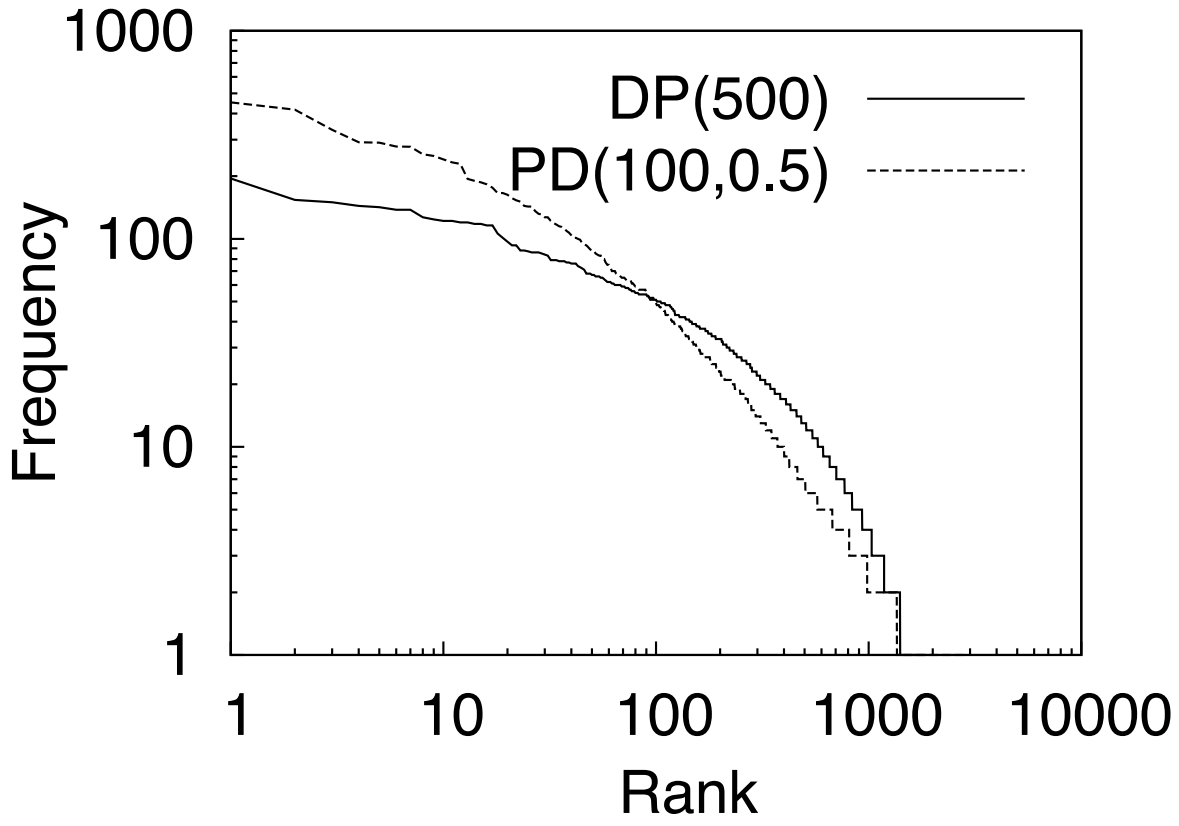

確率分布p




対数 log(p)

# ディリクレ過程とPitman-Yor過程の比較

- 『数学セミナー』2007年11月号「生きたことばをモデル化する」に使った図
  - Pitman-Yor過程の方が、言語の冪乗則をよく再現

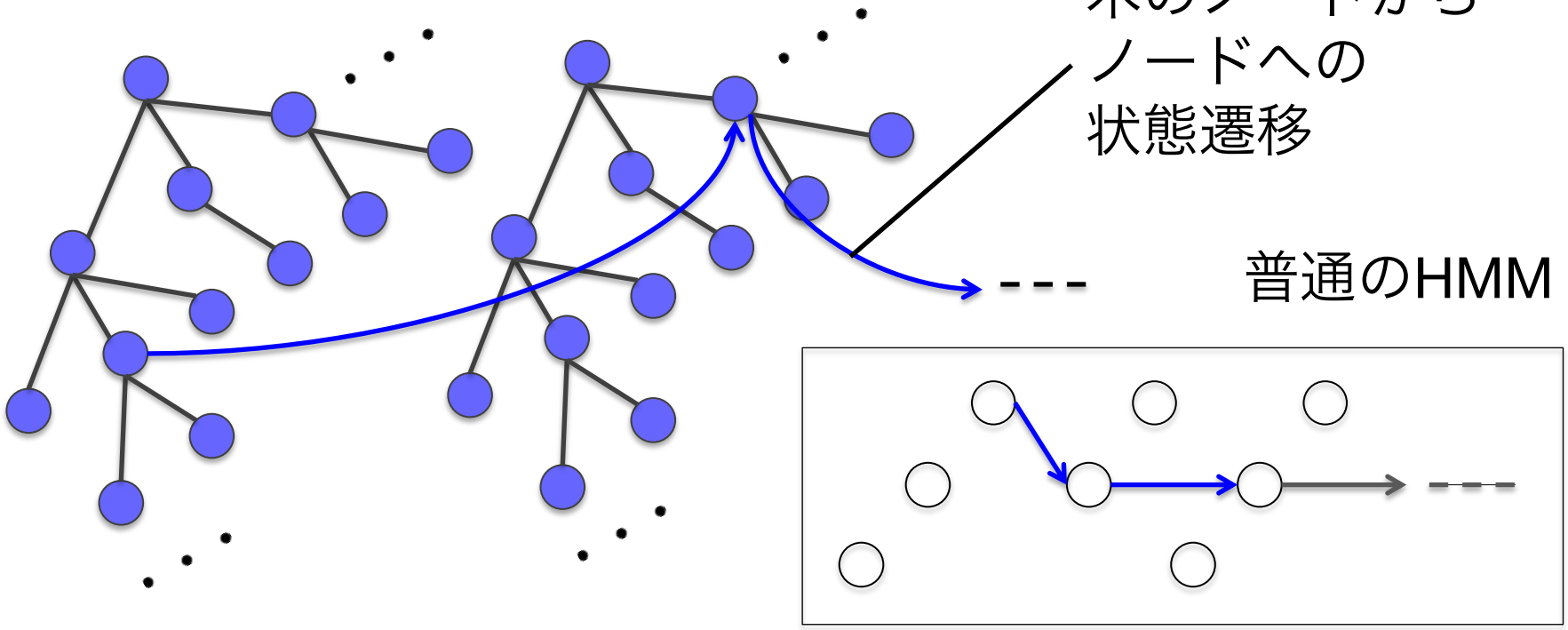


# 無限木構造隠れMarkovモデル

(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



# 説明の流れ

- 隠れMarkovモデル(HMM)とは
- 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

# 単語の時系列データ

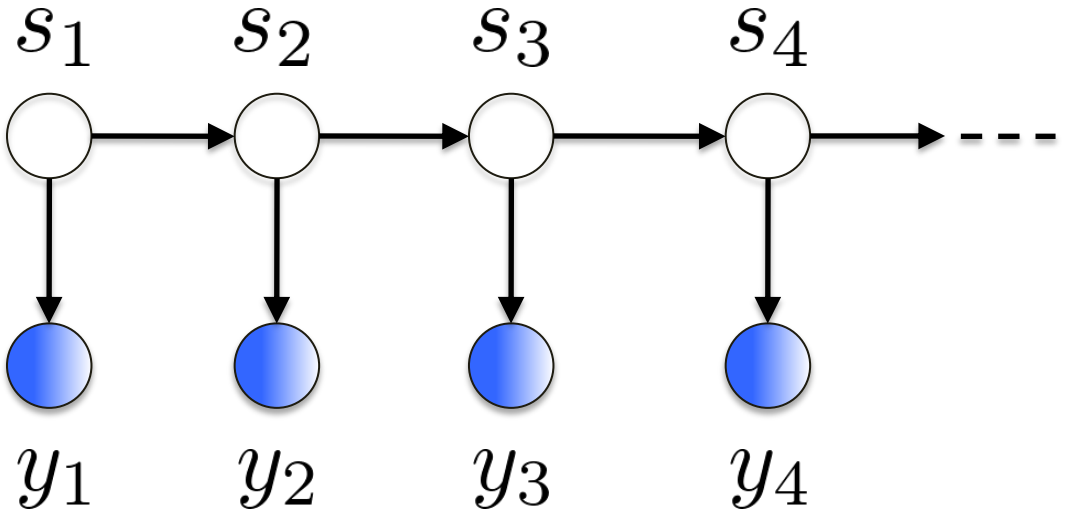
- 本当は、言語の入力は時系列

$$\mathbf{x} = (17\ 5\ 3\ 2\ 108\ 91\ 2\ 34\ \cdots)$$

*When he was a young boy, a book …*

- これをどのようにモデル化するか？
  - 面白い複雑なモデルは色々考えられるが、
  - 最も簡単な隠れマルコフモデル (HMM)について
- HMMは、言語に限らずコンピュータサイエンス全般の基礎 (ロボティクス、バイオ、経済学、…)

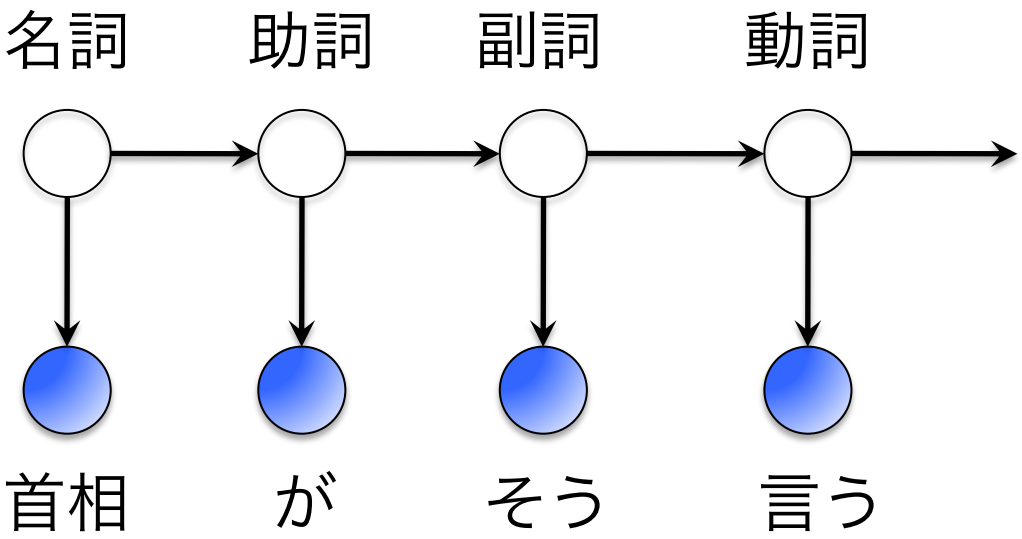
# HMMの基礎




- 各時刻tの観測値 $y_t$ に、隠れ状態 $s_t$ が存在
  - 一般には $y_t \in \mathbb{R}^d,\ s_t \in \mathbb{R}^K$
  - 自然言語処理での最も簡単な場合は、
    $y_t = w_t \in \{1, \cdots, V\}$: 単語、$s_t \in \{1, \cdots, K\}$ : 隠れ状態

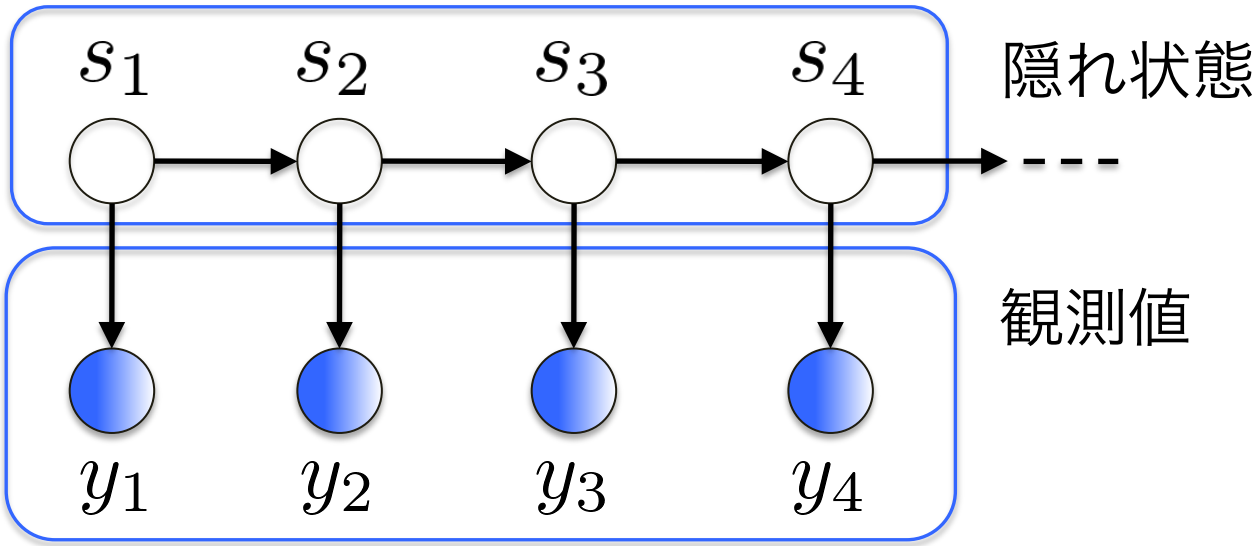
# 統計的自然言語処理での離散HMM

- 最もわかりやすい例→品詞の学習 (形態素解析)




  - 日本語形態素解析器・茶筌はHMMの教師あり学習としてモデル化 (竹内1997)
  - 半教師あり学習にも不可欠 (Suzuki+2008)

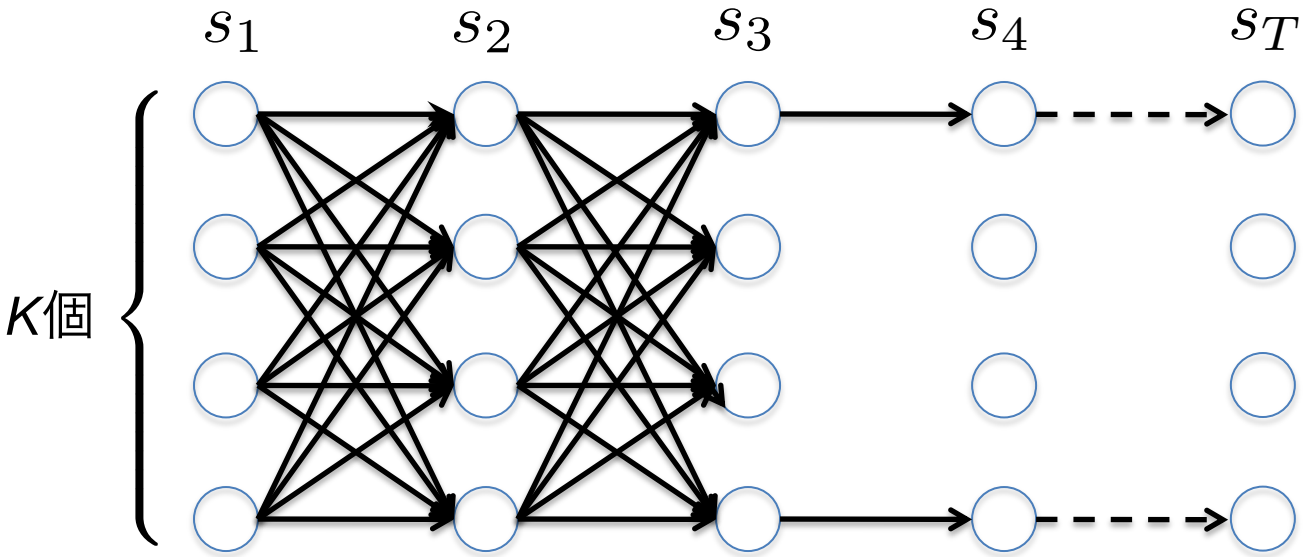
# HMMの定式化




- 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：

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

# 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

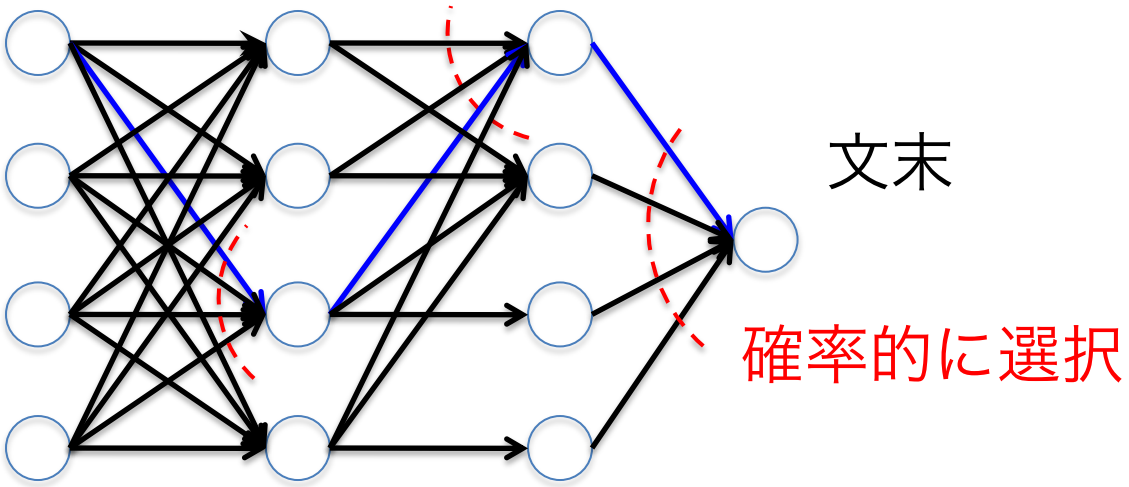
$$\alpha_t(s) = p(y_t = s, x_1 \cdots x_t) \quad \text{(内側確率)}$$

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

- 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法: 各データの持つ状態系列を実際にサンプリング → 局所解に陥らない
- Forward Filtering-Backward Sampling (Scott 2002)

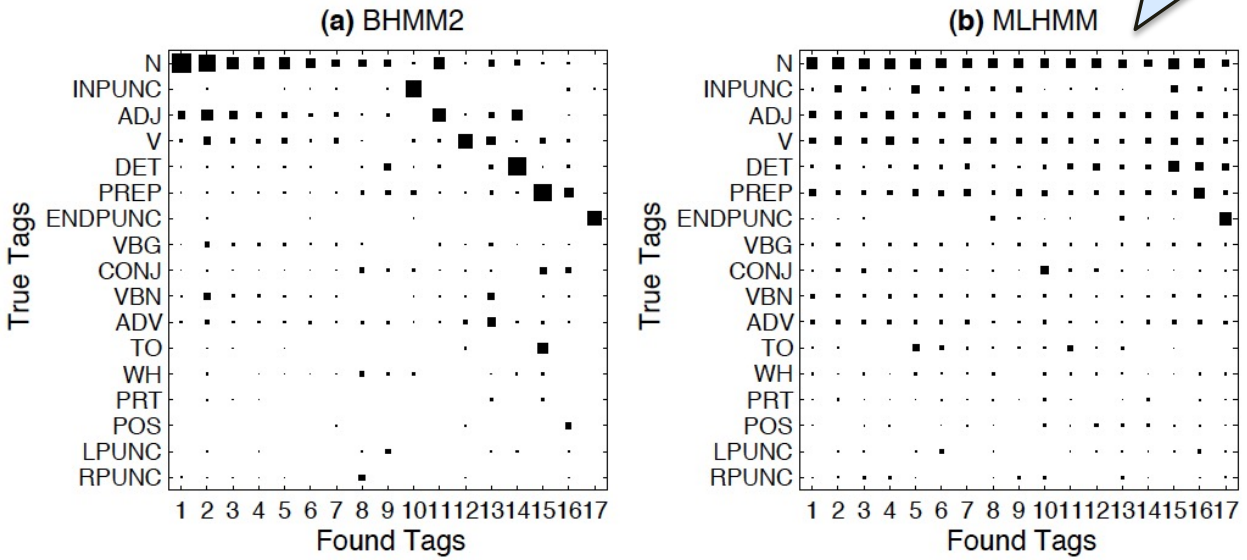



  - 内側確率を計算しておいて、文末から確率的に選択 (確率的 Viterbi)

# 教師なし品詞解析

状態遷移行列

ベイズ推定

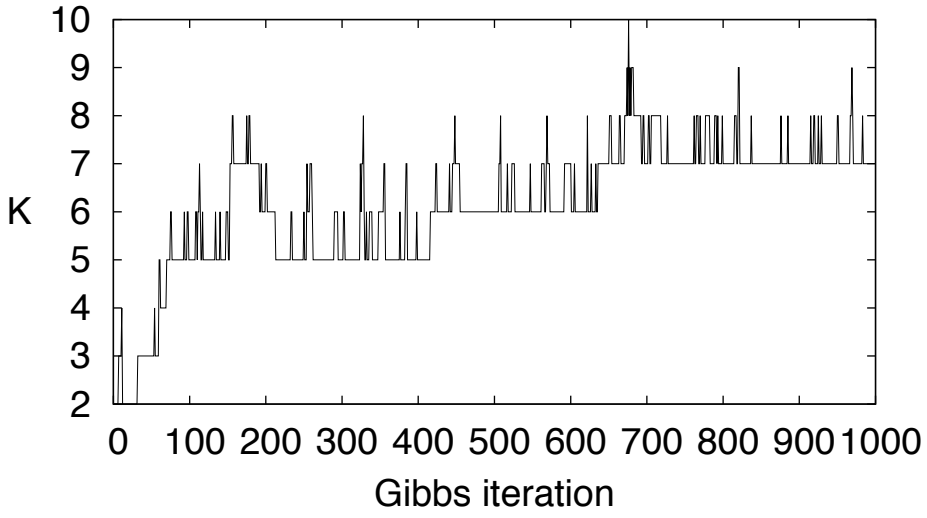



最尤推定

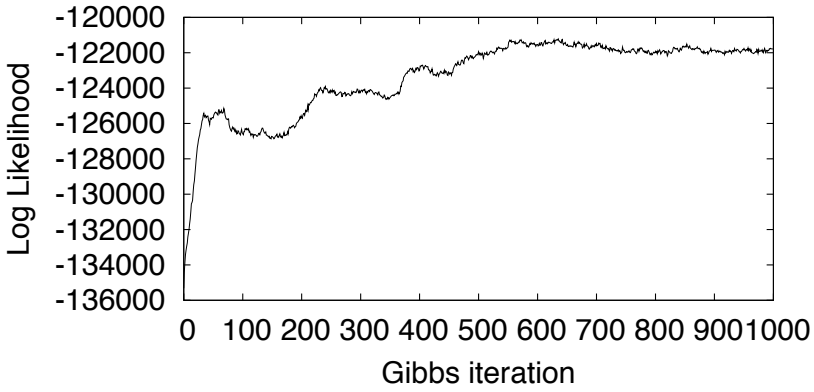
- 1990年代: Merialdo+(1994), Kupiec(1992)→失敗
- 2000年代: ベイズ学習で成功 (Goldwater+07, van Gael+09)
  - Baum-Welchは最尤推定なので局所解にはまる
  - MCMC法による学習

# 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)

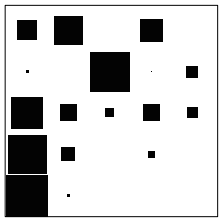



尤度



# “Alice in Wonderland”の解析

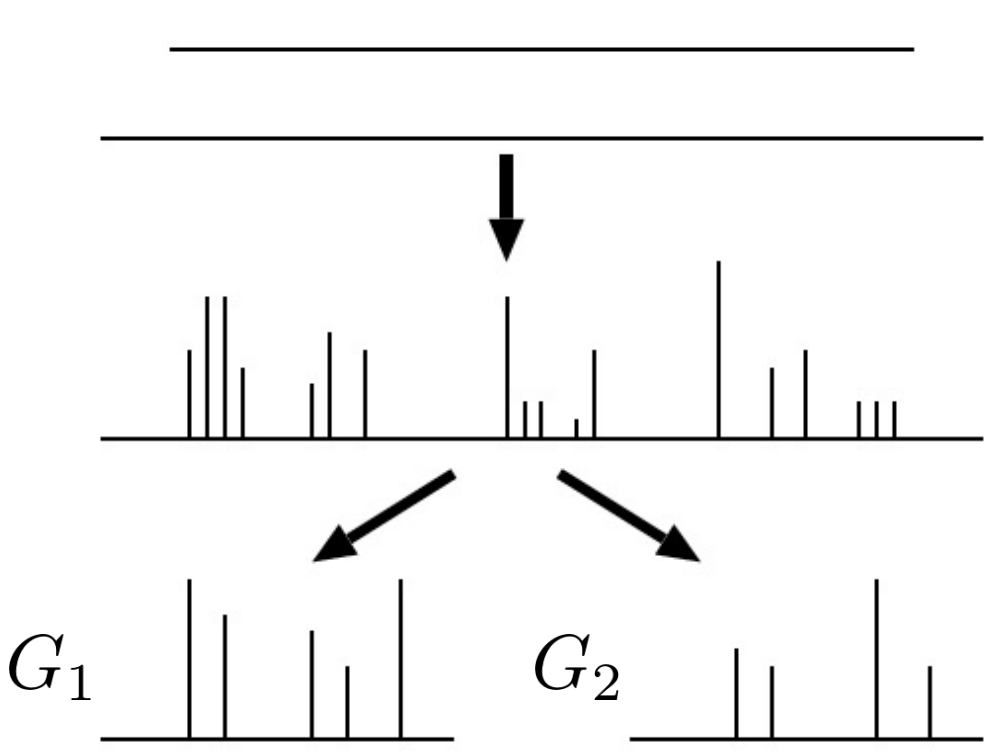
状態遷移行列



| State 1 | | State 2 | | State 3 | | State 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 |

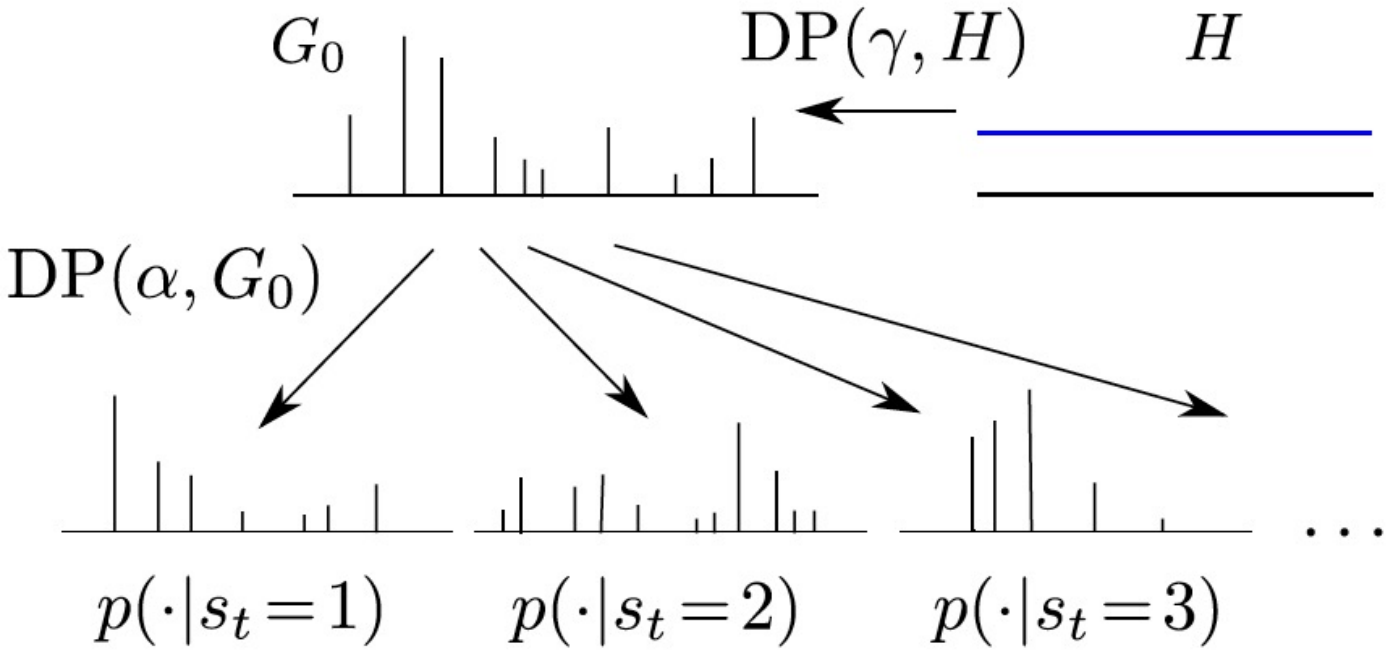
# 階層ディリクレ過程 (HDP) (Teh+ 2006)

- 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)

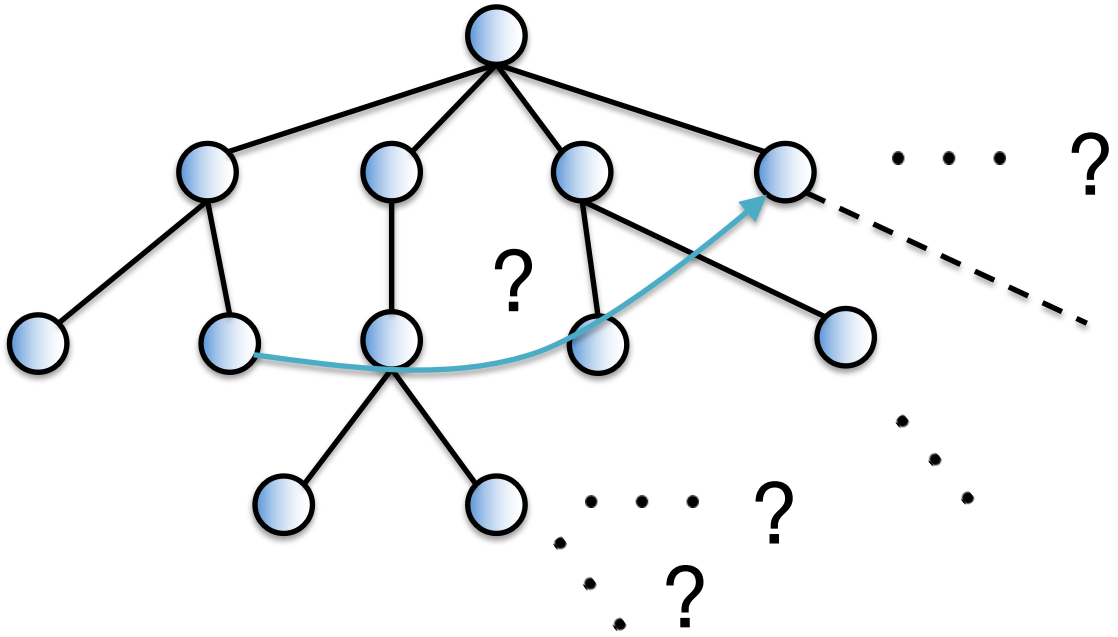
- なぜHDPが必要?→
  例えば、HMMではHDPを使わず別々に状態遷移分布を生成すると、遷移先がバラバラになってしまう



# これで充分か…?

- 京大コーパスや国立国語研究所コーパス等の実際の言語の品詞は、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?

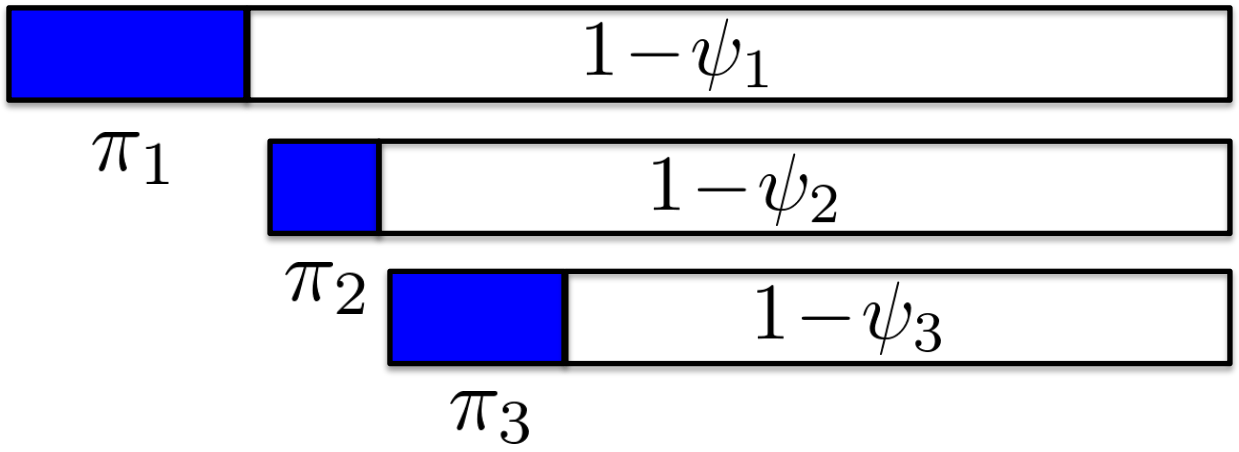
  ⇨ ナイーブな方法では不可能！

# 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

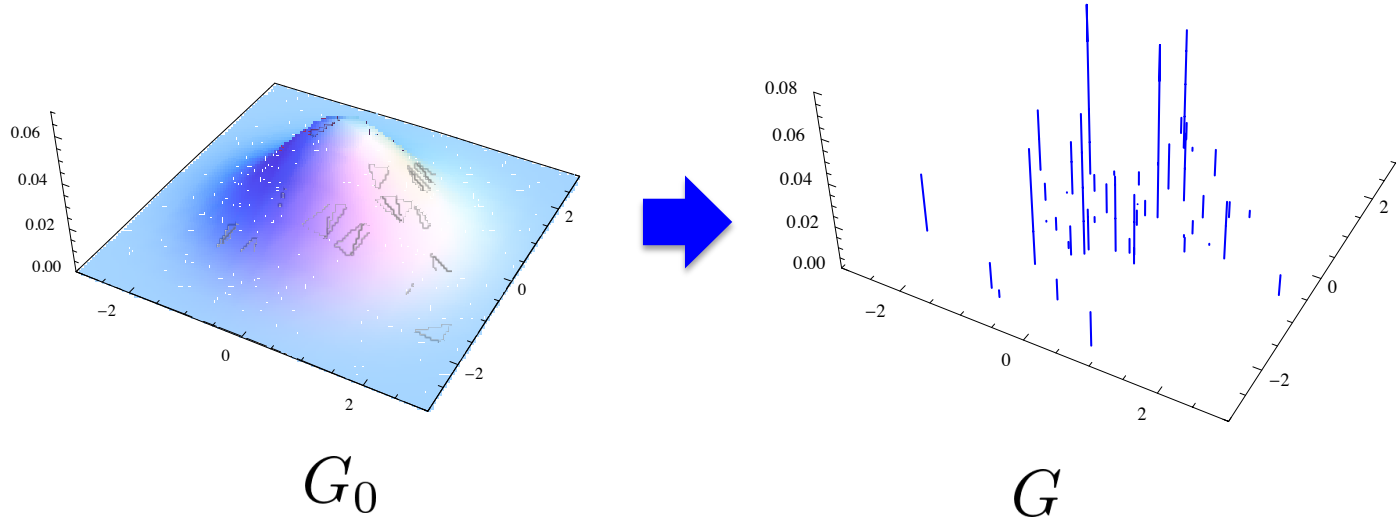
# Stick-breaking process (SBP)

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




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

# ディリクレ過程とStick-breaking表現




$G_0$ $G$

- ディリクレ過程からのサンプル $G \sim \mathrm{DP}(\alpha, G_0)$ は、次のようにして構成できる (Sethuraman 1994)

$$G = \sum_{k=1}^{\infty} \pi_k \, \delta(X_k) \qquad \pi_k = \theta_k \prod_{j=1}^{k-1} (1-\theta_j), \;\; \theta_k \sim \mathrm{Be}(1, \alpha)$$

$$X_k \sim G_0 \qquad (k = 1, \cdots, \infty)$$

# Stick-breaking process (3)

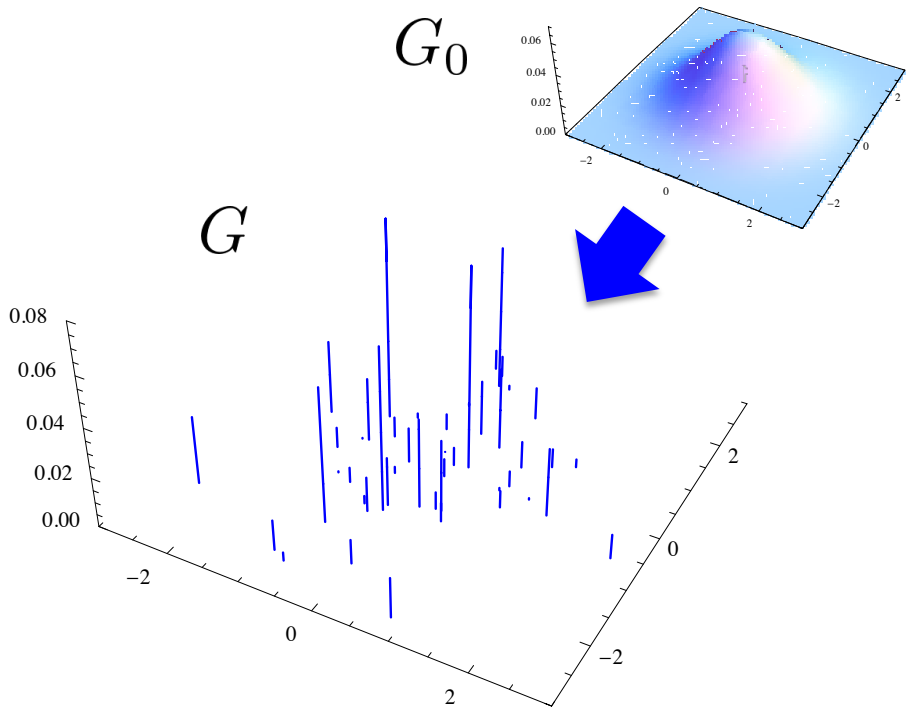
- ディリクレ過程 $G \sim \mathrm{DP}(\alpha, G_0)$ は、SBP(α)で表現できる

$$\pi_k = \theta_k \prod_{j=1}^{k-1} (1 - \theta_j),$$

$$\theta_k \sim \mathrm{Be}\,(1, \alpha),$$

$$X_k \sim G_0,$$

$$G = \sum_{k=1}^{\infty} \pi_k \, \delta_{X_k}$$




  - $G_0$からランダムに選んだ場所 $X_k$ に、高さ $\pi_k$ の棒を立てていったものがG

# DPのstick-breaking表現の証明

- まず、

$$G = \sum_{k=1}^{\infty} \theta_k \prod_{j=1}^{k-1} (1-\theta_j) \cdot \delta_{X_k}$$

$$= \theta_1 \delta_{X_1} + (1-\theta_1)\theta_2 \delta_{X_2} + \cdots$$

から

$$G \overset{d}{=} \theta_1 \delta_{X_1} + (1-\theta_1) G'$$

が成り立つことに注意する. ここでG'はGと同じ分布からの独立なサンプル.

# DPのstick-breaking表現の証明 (2)

- よって、$\mathcal{X}$ の可測な分割 $A_1, \cdots, A_k$ に対して、この測度$G$は

$$(G(A_1), \cdots, G(A_k)) \stackrel{d}{=} \theta_1(\delta_{X_1}(A_1), \cdots, \delta_{X_1}(A_k)) + (1-\theta_1)(G'(A_1), \cdots, G'(A_k))$$

を満たす.

- $W = \theta_1,\ U = (\delta_{X_1}(A_1), \cdots, \delta_{X_1}(A_k)),$ $V = (G(A_1), \cdots, G(A_k))$ とおくと、

$$V \stackrel{d}{=} WU + (1-W)V$$

となり、これを満たす$V$は一意に定まる.

# 補題4

- 補題1: $W \in (-1,1)$, $U$ を確率ベクトルとする. $U$ と同じ次元の確率ベクトル $V$ が $W, U$ とは独立で

$$V \overset{d}{=} U + WV \qquad (*)$$

のとき、$V$ の分布は一意に定まる.

Proof.
V, V' がともに(*)を満たすとする. (Wn,Un)を(W,U)の独立なコピーとし, $V_1 = V,\ V_1' = V'$ とすると、(*)から

$$\begin{cases} V_2 = U_1 + W_1 V_1 \\ V_2' = U_1 + W_1 V_1' \end{cases} \text{同様にして} \begin{cases} V_n = U_n + W_n V_n \\ V_n' = U_n + W_n V_n' \end{cases}$$

# 補題4 (2)

$$\begin{cases} V_n = U_n + W_n V_n \\ V'_n = U_n + W_n V'_n \end{cases}$$

- を順番に作ったとき、

$$|V_{n+1} - V'_{n+1}| = |W_n||V_n - V'_n| = \cdots$$
$$= \prod_{i=1}^{n} |W_i| \cdot |V - V'| \to 0 \quad (\because |W_i| < 1)$$

よって $V \overset{d}{=} V'$. $\square$

- これから、

$$V = WU + (1 - W)V$$
$$= (1 - W')V + W'V \quad (W' = 1 - W)$$

を満たすVは一意に定まる. $\square$

# DPのstick-breaking表現の証明 (3)

- 元に戻って、
$$V \overset{d}{=} WU + (1-W)V$$
ゆえ、$V \sim \mathrm{Dir}(\alpha(A_1), \cdots, \alpha(A_k))$ のとき
$$WU + (1-W)V \sim \mathrm{Dir}(\alpha(A_1), \cdots, \alpha(A_k))$$
を示せばよい. ただし $W \sim \mathrm{Be}(1, \alpha)$ .
- *U*は特定の*j*について $U = \mathbf{e}_j$ となるから、左辺の*U*の条件付き分布は
$$WU + (1-W)U \mid U = \mathbf{e}_j \sim \mathrm{Dir}(\alpha_1^{(j)}, \cdots, \alpha_k^{(j)})$$
  - ここで $\alpha_i^{(j)} = \alpha_i \ (i \neq j),\ \alpha_i^{(j)} = \alpha_i + 1 \ (i = j)$

# 補題5

- $U \sim \mathrm{Dir}(\alpha_1, \cdots, \alpha_k),\ V \sim \mathrm{Dir}(\beta_1, \cdots, \beta_k),$
  $W \sim \mathrm{Be}\left(\sum_k \alpha_k, \sum_k \beta_k\right)$
  のとき、

$$WU + (1-W)V \sim \mathrm{Dir}(\alpha_1 + \beta_1, \cdots, \alpha_k + \beta_k)$$

- Proof: $Z_1, \cdots, Z_{2k}$ について

$$Z_j \sim \mathrm{Ga}(\alpha_j, 1),\ Z_{k+j} \sim \mathrm{Ga}(\beta_j, 1) \qquad (j = 1, \cdots, k)$$

とする. ディリクレ分布は正規化ガンマ分布として書けるので、

$$(U, V, W) \overset{d}{=} \left( \frac{Z_1}{\sum_j Z_j}, \cdots, \frac{Z_k}{\sum_j Z_j}, \frac{Z_{k+1}}{\sum_j Z_{k+j}}, \cdots, \frac{Z_{2k}}{\sum_j Z_{k+j}}, \frac{\sum_{j=1}^{k} Z_j}{\sum_{j=1}^{2k} Z_j} \right)$$

# 補題5 (2)

- このとき、

$$UW + (1-W)V \overset{d}{=} \left( \frac{Z_1}{\sum_j Z_j} \left( \frac{\sum_j Z_j}{\sum_{j=1}^{2k} Z_j} \right) + \left( \frac{\sum_j Z_{k+j}}{\sum_{j=1}^{2k} Z_j} \right) \frac{Z_{k+1}}{\sum_j Z_{k+j}}, \cdots \right)$$

$$= \left( \frac{Z_1}{\sum_{j=1}^{2k} Z_j} + \frac{Z_{k+1}}{\sum_{j=1}^{2k} Z_j}, \cdots \right)$$

$$\sim \mathrm{Dir}\left(\alpha_1 + \beta_1, \cdots, \alpha_k + \beta_k\right). \quad \square$$

# DPのstick-breaking表現の証明 (4)

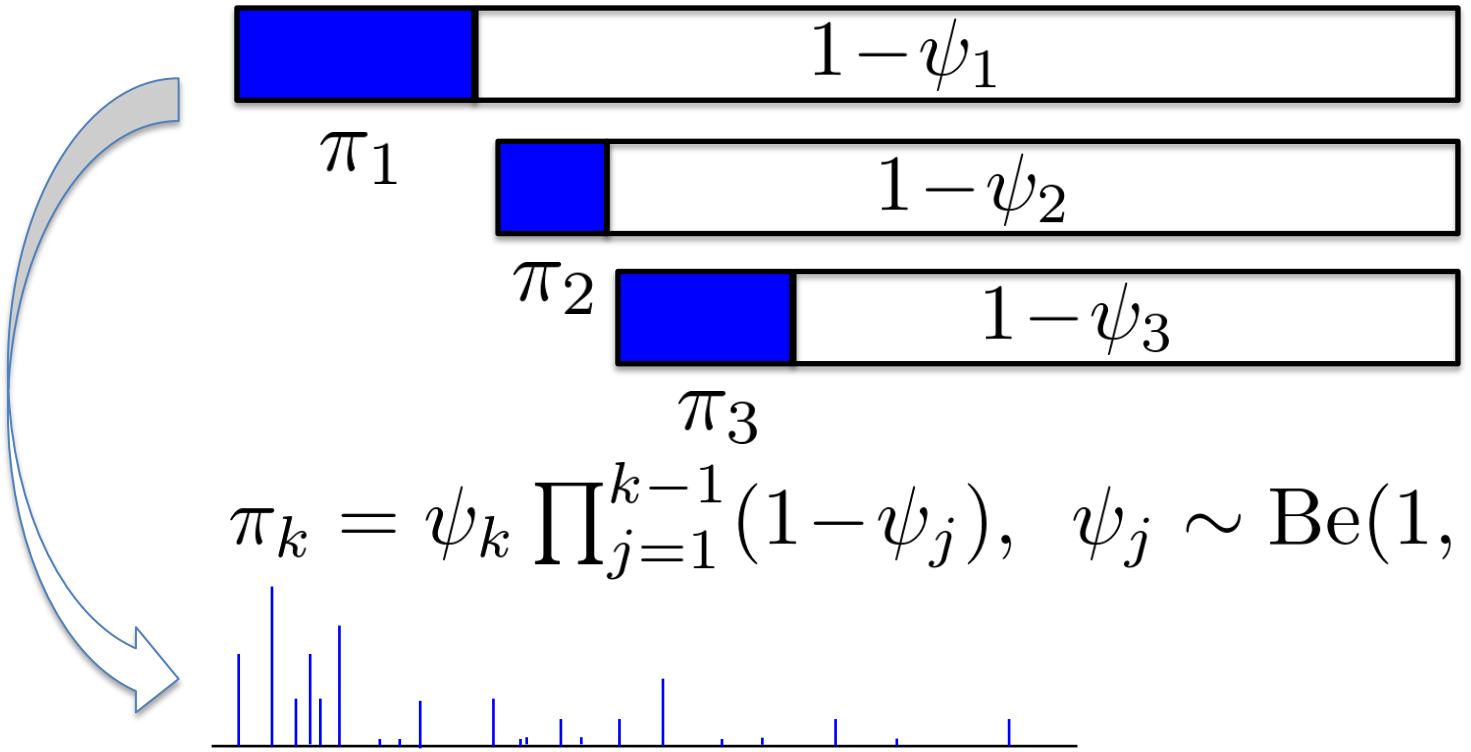
- Uについて期待値をとれば、

$$
\begin{aligned}
& WU + (1-W)V \\
&= \mathbb{E}_U \left[ WU + (1-W)V \mid U = \mathbf{e}_j \right] \\
&= \sum_{j=1}^{k} p(U)\, p(WU + (1-W)V) \\
&= \sum_{j=1}^{k} \frac{\alpha_j}{\alpha} \operatorname{Dir}(\alpha_1^{(j)}, \cdots, \alpha_k^{(j)}) \sim \operatorname{Dir}(\alpha_1, \cdots, \alpha_k)\,. \quad \square
\end{aligned}
$$

ここも補題が必要だが省略(単純計算)

# Stick-breaking process (SBP)

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



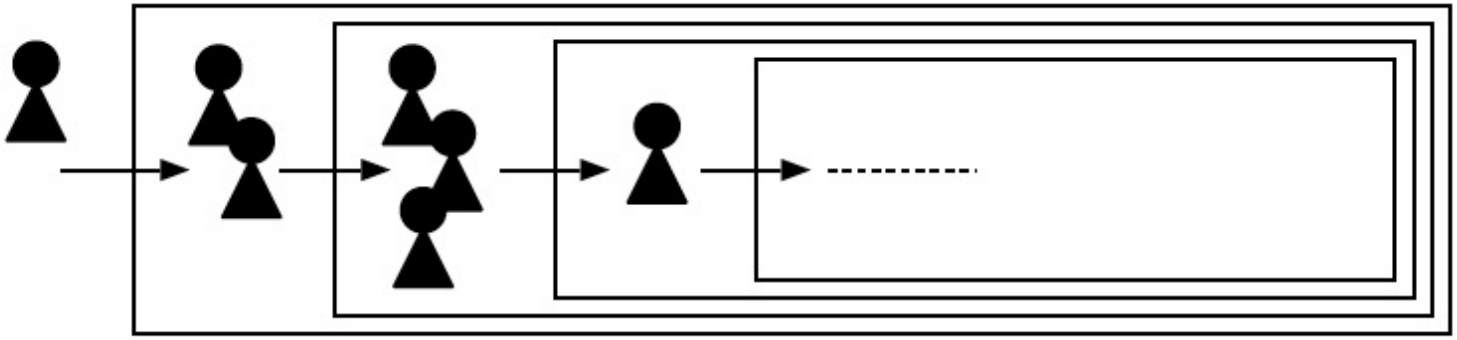

$$\pi_k = \psi_k \prod_{j=1}^{k-1}(1-\psi_j), \quad \psi_j \sim \mathrm{Be}(1, \gamma)$$

# Stick-breaking process (4)

- SBPの要素 $\psi_k$ はベータ分布 $\mathrm{Be}(1, \gamma)$ に従う
  → $\pi$ の事後分布が求まる
- 無限次元多項分布 $\pi$ からk番目の値が選ばれる

$\psi_1 \cdots \psi_{k-1}$ までは棒を折り続ける
$\psi_k$ で止まる

# Chinese District Process (CDP)



$k=1 \quad k=2 \quad k=3 \quad k=4$

- SBPのCRP表現 (Paisley+ (2008))
- k番目の領域(番人)で止まった回数を $n_0(k)$ 、通過した回数を $n_1(k)$ とすると、

$$\psi_k|\mathcal{D} \sim \mathrm{Be}(1+n_0(k), \alpha+n_1(k))$$

$$E[\psi_k|\mathcal{D}] = \frac{1+n_0(k)}{1+\alpha+n_0(k)+n_1(k)}$$

# Chinese District Process (2)

- よって、CDP表現を使えば、$\pi$ の事後期待値は

$$E[\pi|\mathcal{D}] = E[\psi_k \prod_{j=1}^{k-1}(1-\psi_j)]$$

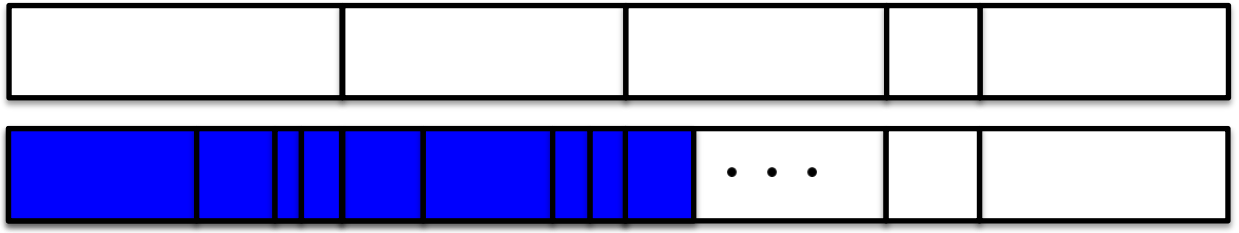
$$= \frac{1+n_0(k)}{1+\alpha+n_0(k)+n_1(k)} \prod_{j=1}^{k-1} \frac{\alpha+n_1(j)}{1+\alpha+n_0(j)+n_1(j)}$$

  - 基本的に、止まった数と通過した数を数えているだけ!

- DP(α)・CRP(α)・CDP(α)はすべて等価な表現

# 階層的離散分布

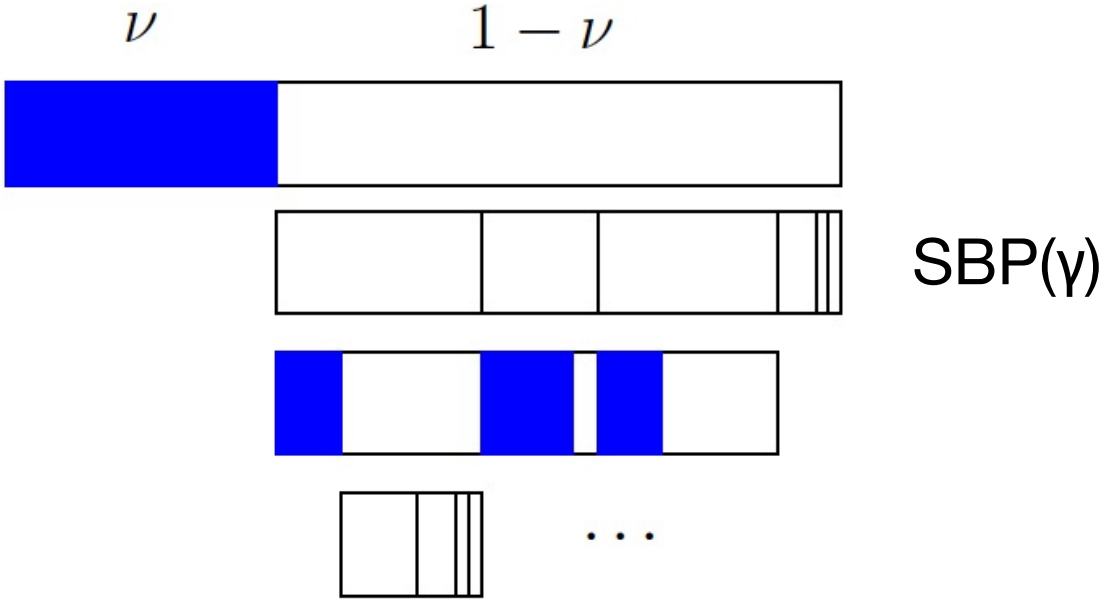
- 最も簡単な階層的離散分布:
  SBPの各stick $\pi_k$ を、再帰的にさらにSBPで分割
  (Polya trees)



- 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?

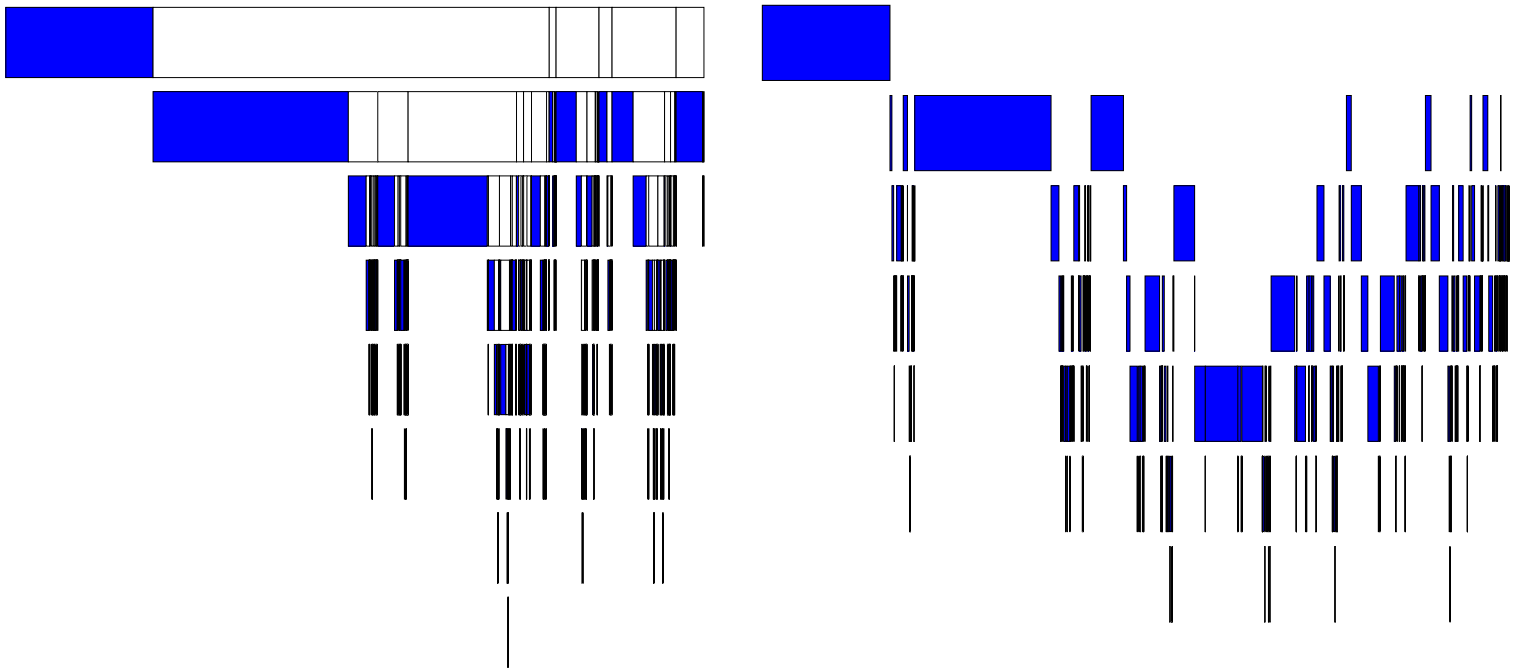
# Tree-structured stick-breaking process

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



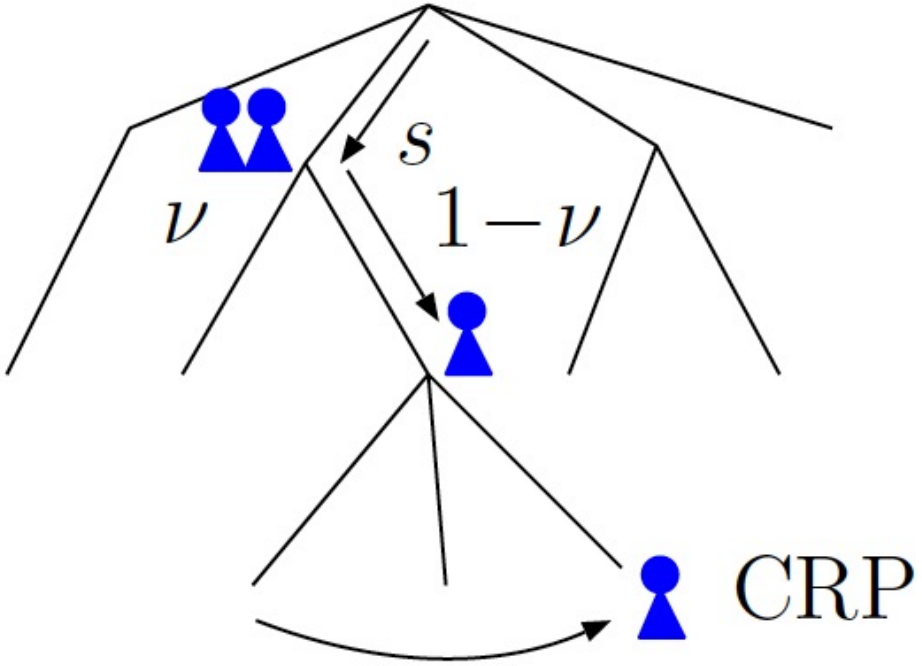
# TSSBの構成

- $\pi \sim \mathrm{TSSB}(\eta, \gamma)$ 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vでそのノードに残る
  - 確率(1-v)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

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

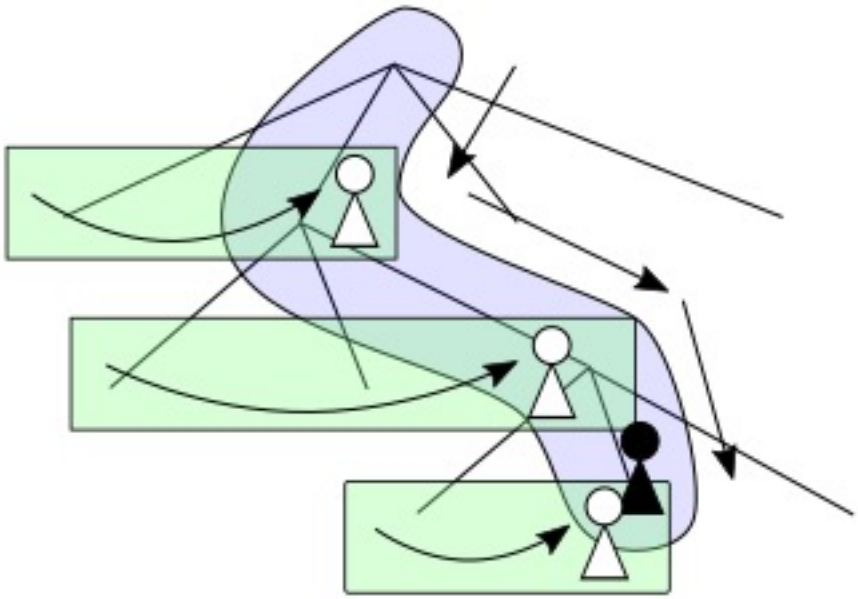
$$\phi_{\mathbf{s}k} = \psi_{\mathbf{s}k} \prod_{j=1}^{k-1} (1 - \psi_{\mathbf{s}j})$$

SBP(γ)

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

# Posterior of TSSB

- Probability of descent at node s is: let $n_0(\mathbf{s})$ be the number of times customers stop at s and $n_1(\mathbf{s})$ be the times customers have descent,

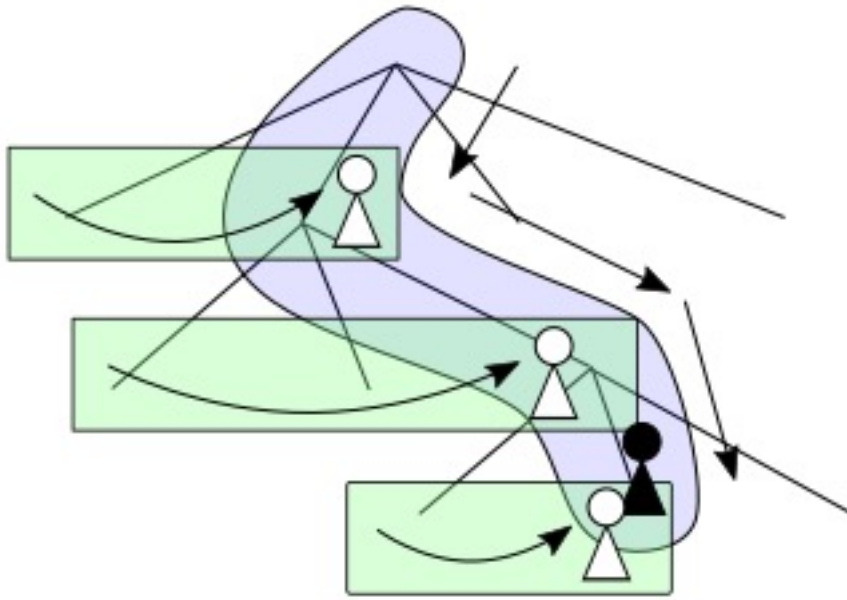


$$E[\nu_{\mathbf{s}}|\mathcal{D}] = \frac{1+n_0(\mathbf{s})}{1+\alpha+n_0(\mathbf{s})+n_1(\mathbf{s})}$$

# Posterior of TSSB (2)

- In a descent from node s, probability to choose a child k is given by SBP:

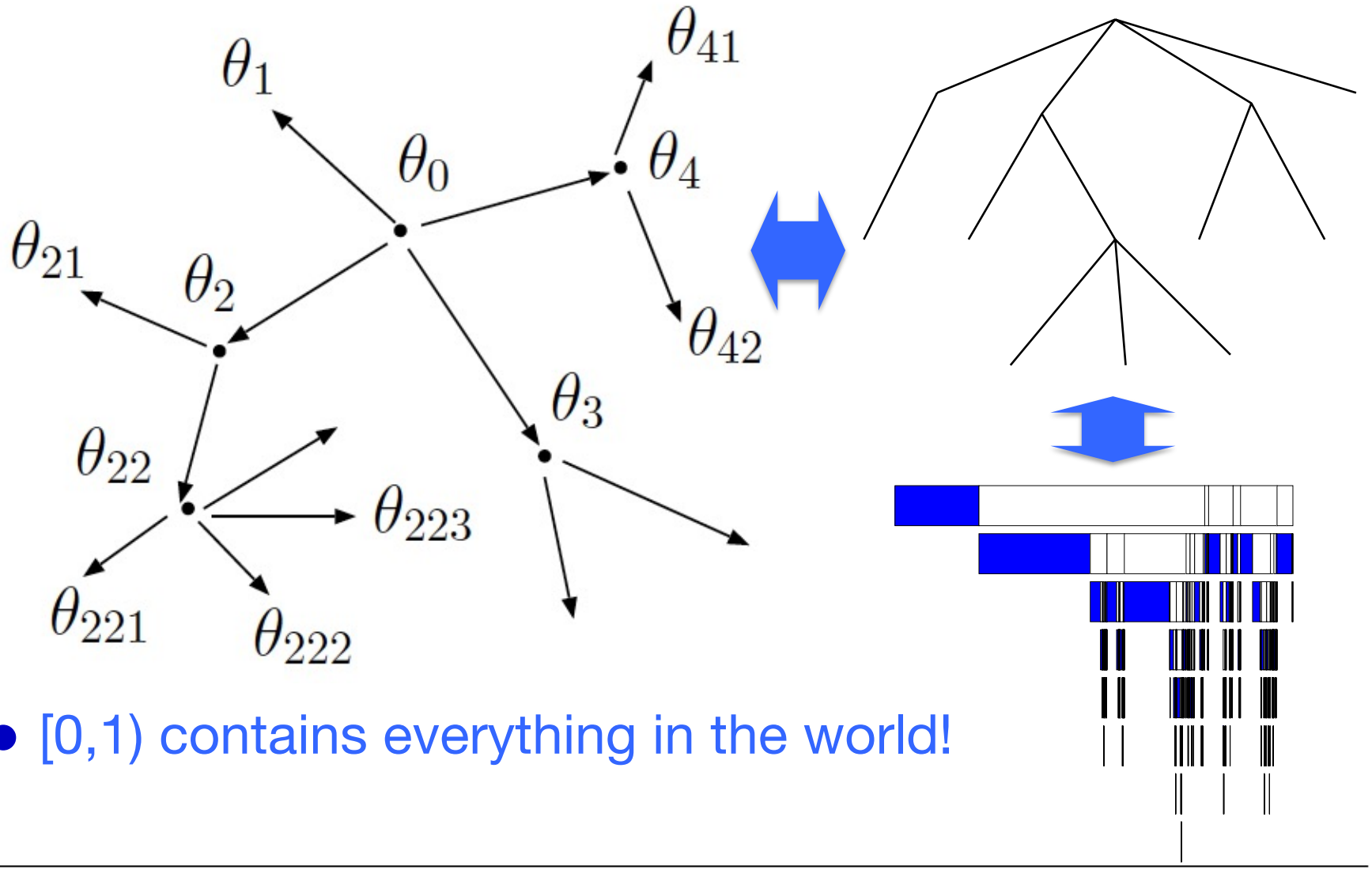
$$\psi_{\mathbf{s}k} \prod_{j=1}^{k-1} (1 - \psi_{\mathbf{s}j})$$



- Let $m_0(\mathbf{t})$ be the number of horizontal customers stayed at node t and $m_1(\mathbf{t})$ be the numbers left,

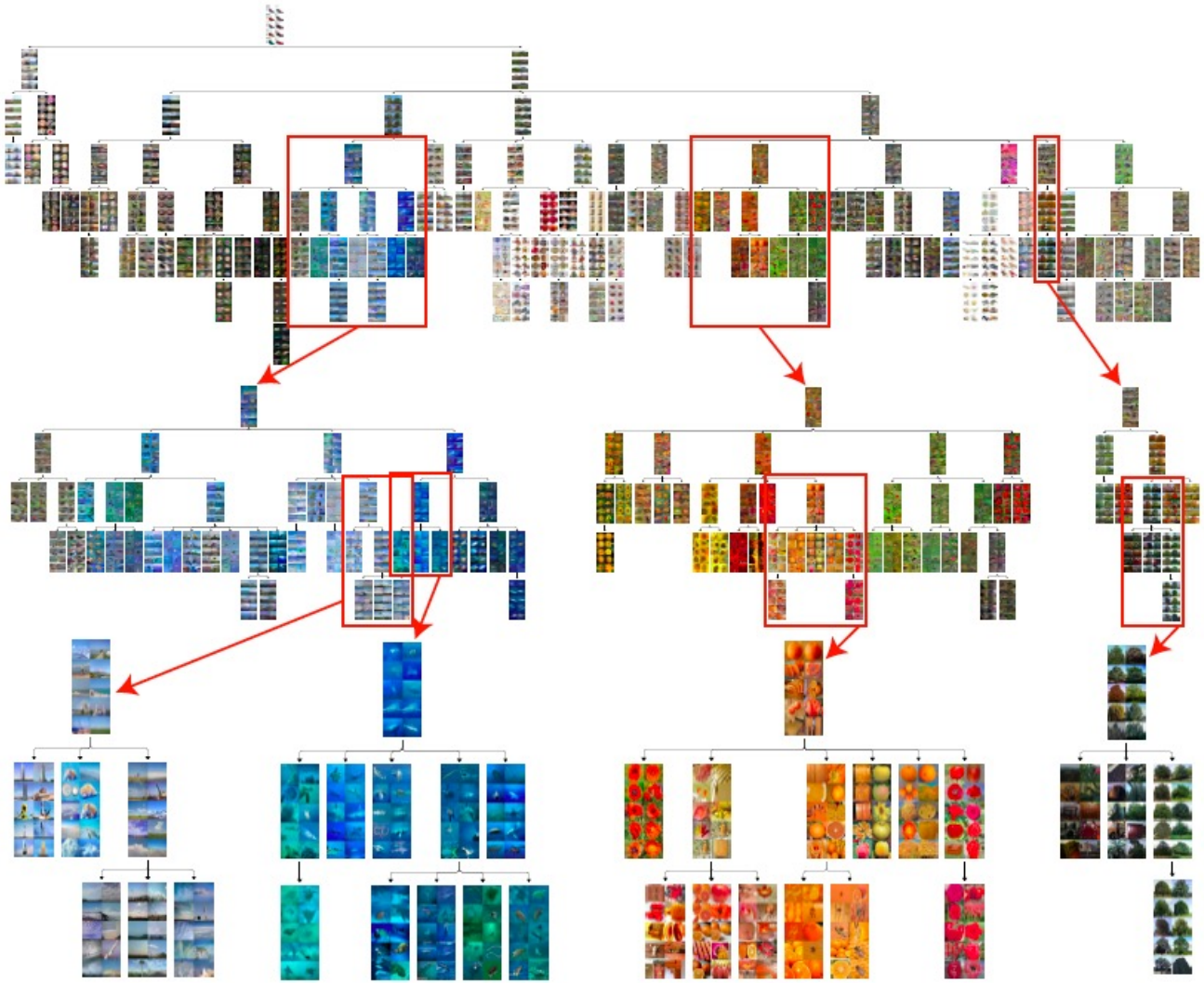
$$E[\psi_{\mathbf{t}}|\mathcal{D}] = \frac{1+m_0(\mathbf{t})}{1+\gamma+m_0(\mathbf{t})+m_1(\mathbf{t})}$$

# TSSB as a diffusion prior



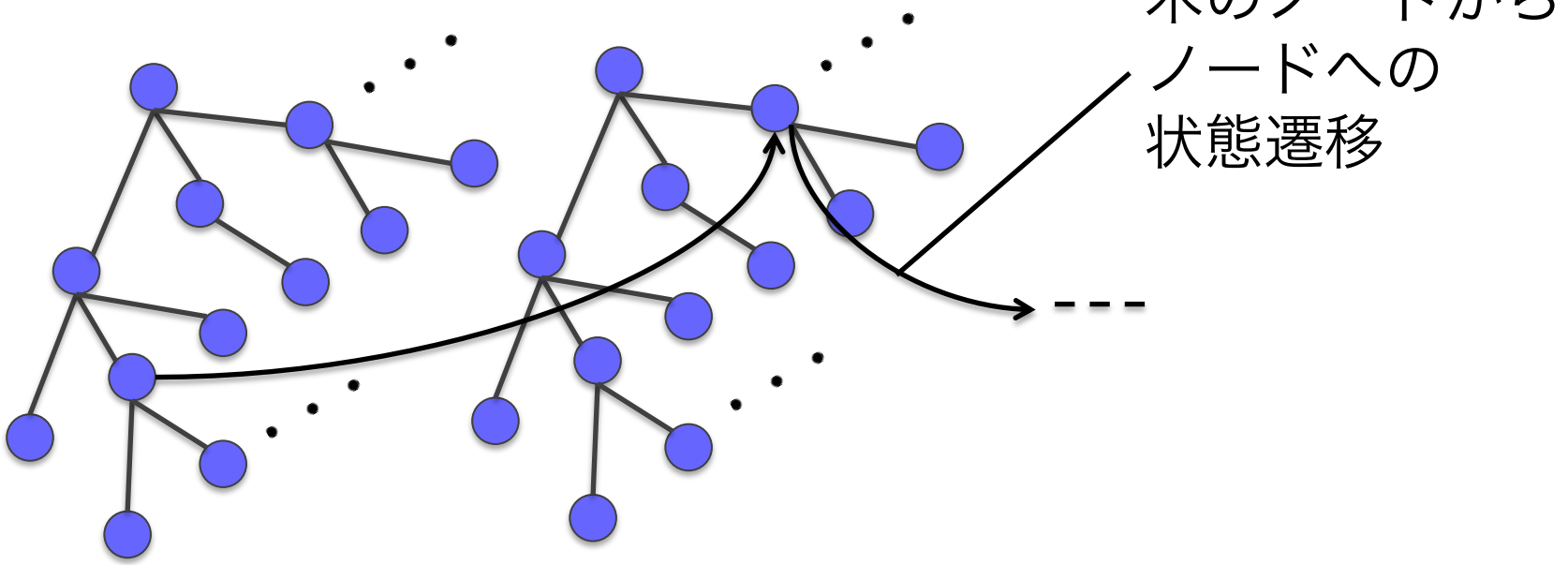

- [0,1) contains everything in the world!

# Hierarchical clustering with TSSB

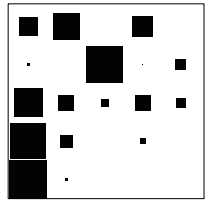


- Adams+ (NIPS 2010)
- CIFAR-10 image data, infinite hierarchy modeling

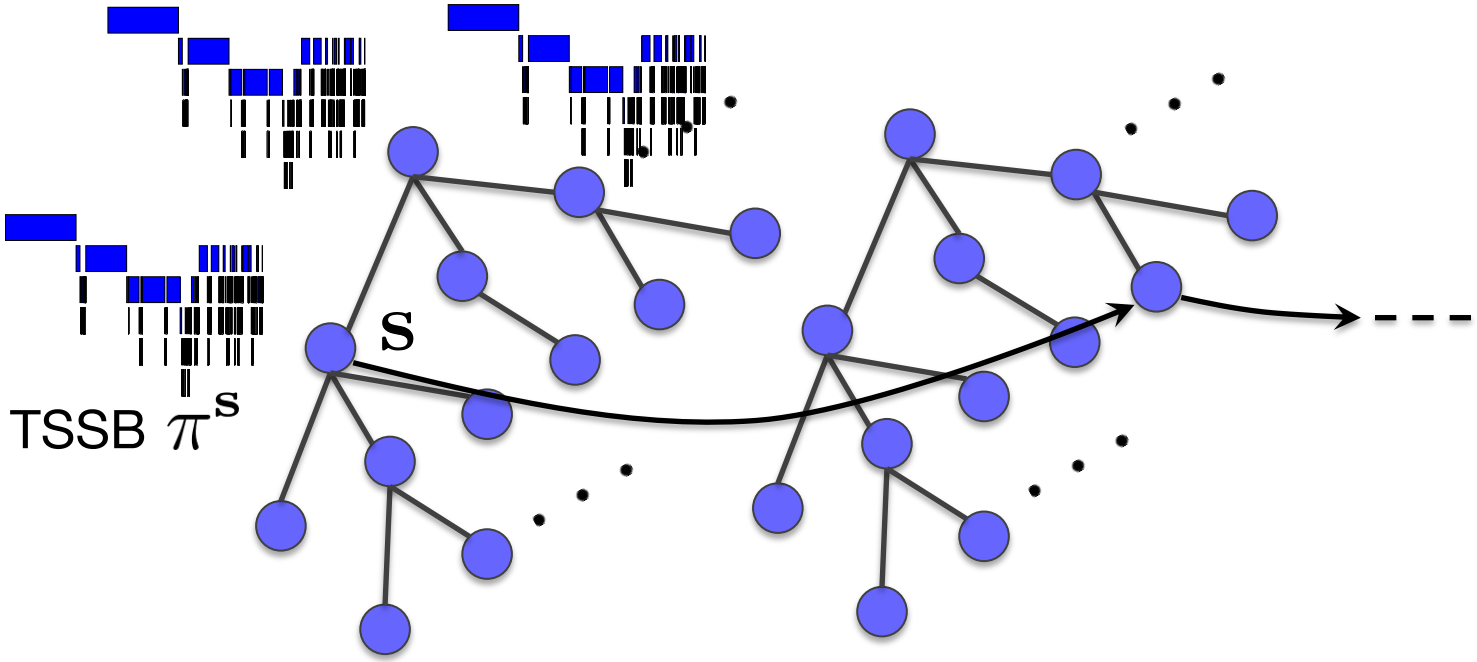
# TSSB上の状態遷移モデル




- 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KxKの遷移行列



# TSSB上の状態遷移モデル (2)




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

# TSSB上の状態遷移モデル (3)

- $\pi^{\mathbf{s}}$ 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!

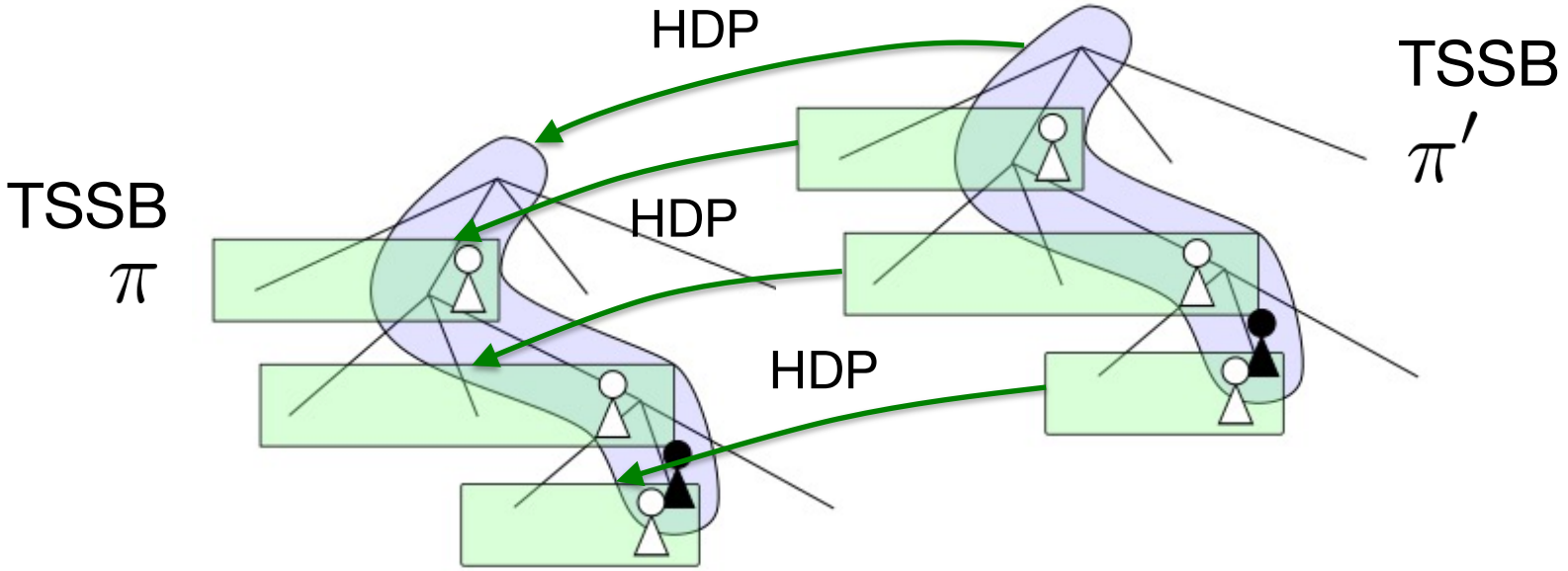
TSSB

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

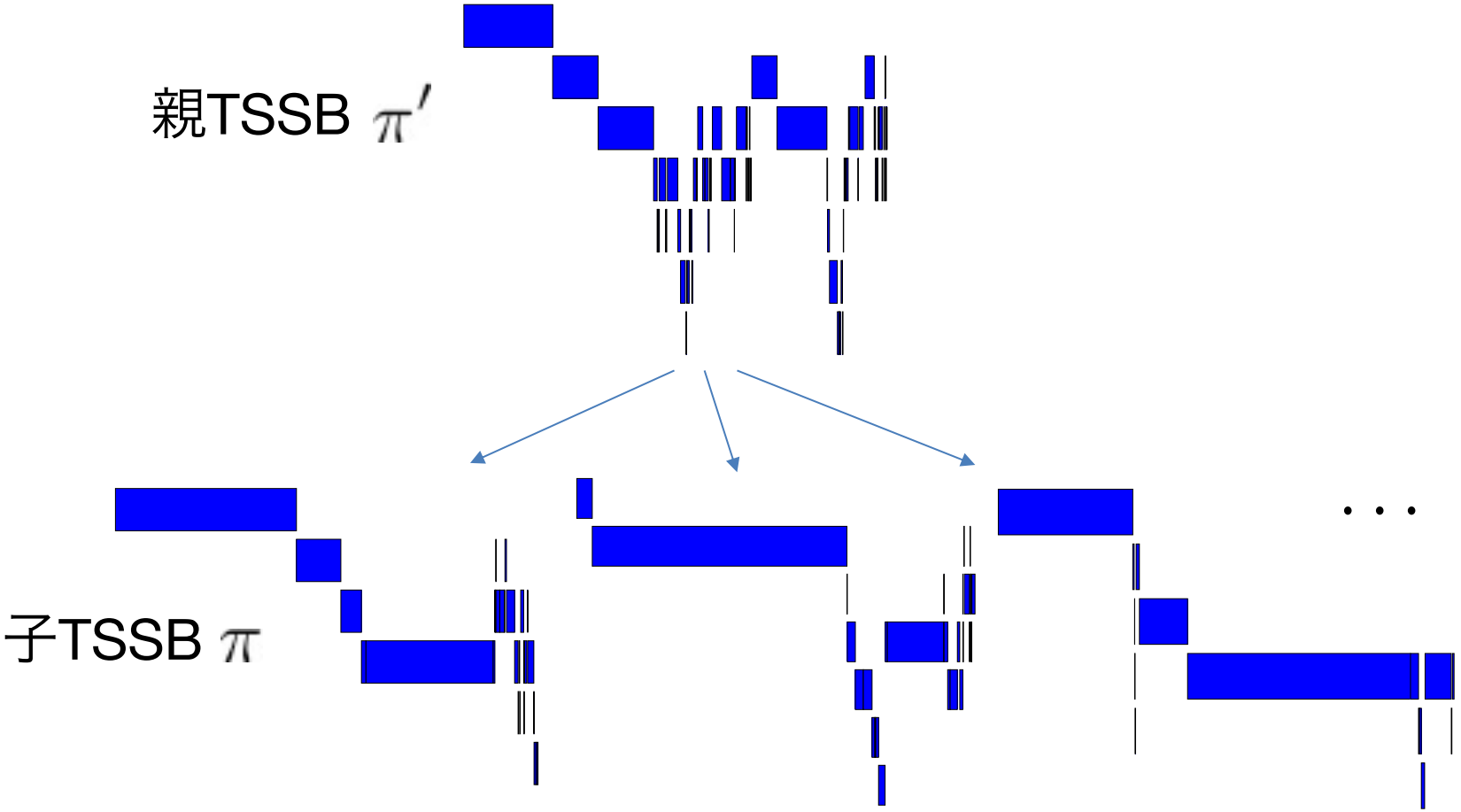
# 階層的木構造Stick-breaking過程 (HTSSB)

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

- $\pi$ 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)



# Random draws from HTSSB



# HTSSB (2)

- HDPのStick-breaking表現より、データDの下で

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

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

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

# HTSSB (3)

- $E[\nu_{\mathbf{s}}|D], E[\psi_{\mathbf{s}}|D]$ がわかれば、$\pi$ の各要素sでの確率は

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

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

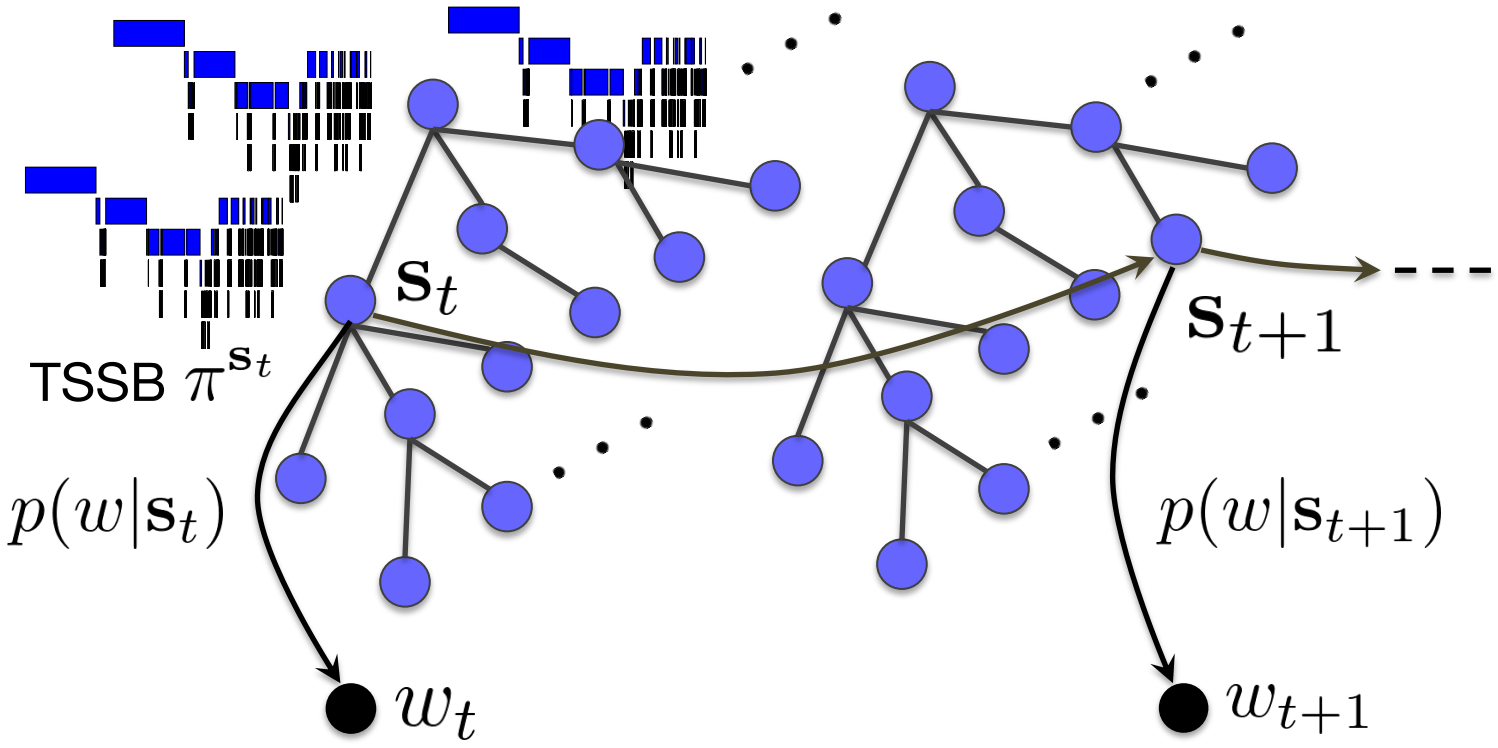
# HTSSBの学習

- TSSB ←→ CDPで事後分布

- HTSSB ←→ 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

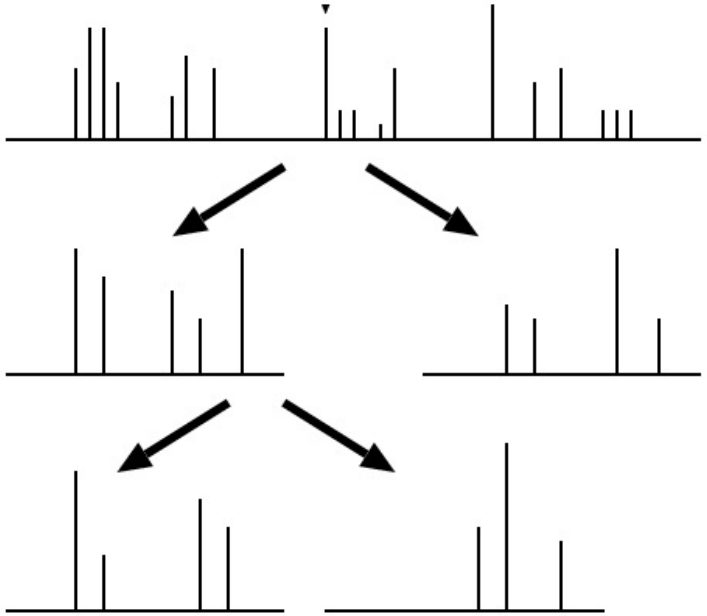
# 無限木構造HMM (iTHMM)

- 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 $p(w|\mathbf{s})$ と $p(w|\mathbf{s}')$ 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

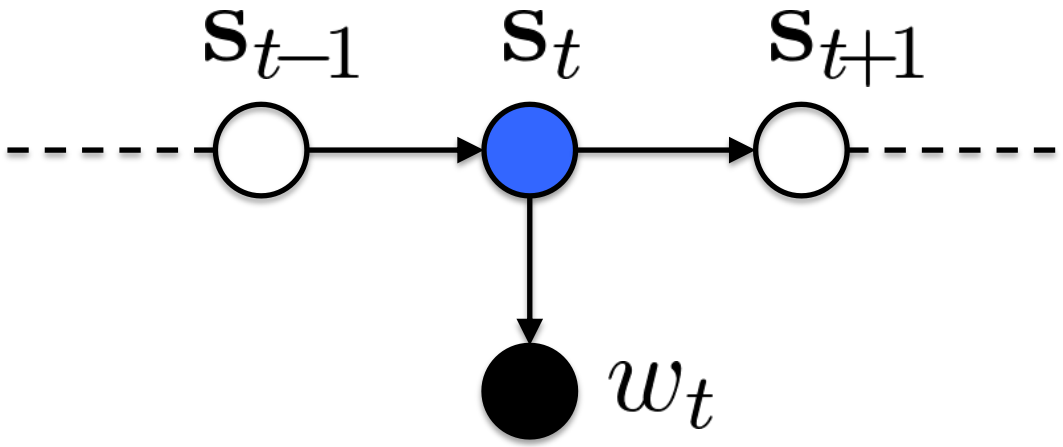


- ハイパーパラメータ $d$,$\theta$も自動推定
- カウントの追加/削除で、木構造上の分布が自動的に更新

# 無限木構造HMMの生成モデル

- iTHMMの生成モデル
  (1) TSSB $\pi^{[]} \sim \mathrm{TSSB}(\eta, \gamma, \lambda)$ 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 $\pi^{\mathbf{s}}$ を親の $\pi^{\mathbf{s}'}$ から
  $$\pi^{\mathbf{s}} \sim \mathrm{HTSSB}(\alpha, \pi^{\mathbf{s}'})$$
  と生成.
  (b) 単語出力確率分布 $G_{\mathbf{s}}$ を親の$G_{\mathbf{s}'}$から
  $$G_{\mathbf{s}} \sim \mathrm{HPY}(d_{|\mathbf{s}|}, \theta_{|\mathbf{s}|}, G_{\mathbf{s}'})$$
  と生成.
- BOSから始めて、隠れ状態列 $\mathbf{s}_1, \mathbf{s}_2, \cdots$と単語列 $w_1, w_2, \cdots$ を生成.

# iTHMMの学習




- Gibbsサンプリング (Goldwater+2007)

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

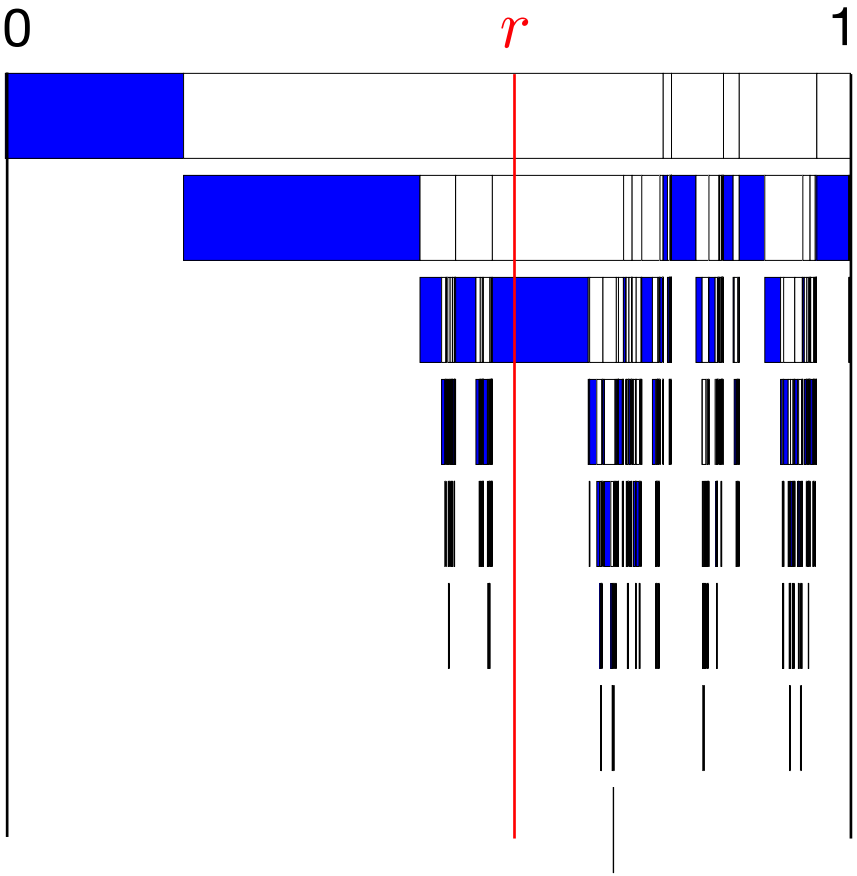
  - $\mathbf{s}_t$ を次々とサンプリング→正しい値に収束

# iTHMMの学習 (2)

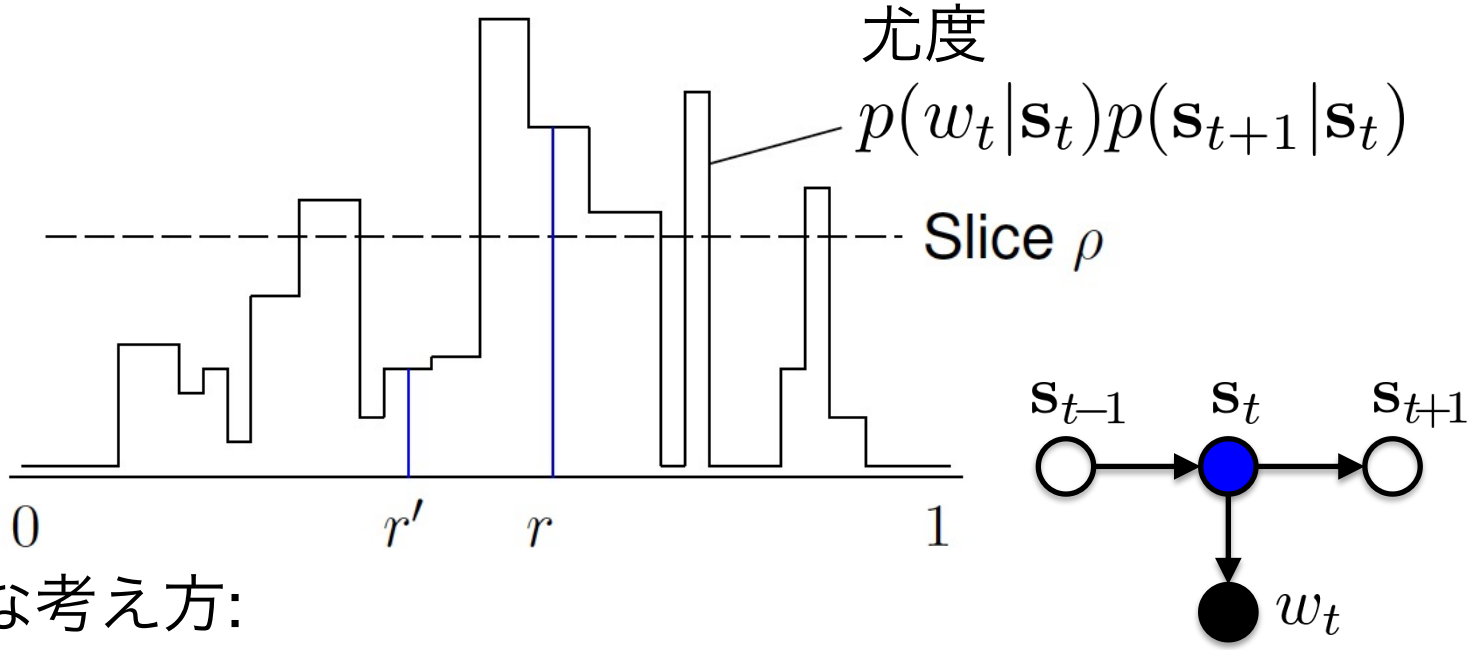
- 問題: $\mathbf{s}_t$ を数え上げられない!
  - $\mathbf{s}_t$ =[], [1 1], [1 1 2], [2 4 3], [17 5 3], $\cdots\cdots$
    と無限に候補が存在
  - iHMMのように、確率的に右側を切り落とすことはできない
  - どうするか?

# iTHMMの学習：注意

- TSSBの全てのノードは、[0,1) に含まれる区間のどれかに対応している→[0,1)の乱数rでノードを選べる



# iTHMMの学習 (3)

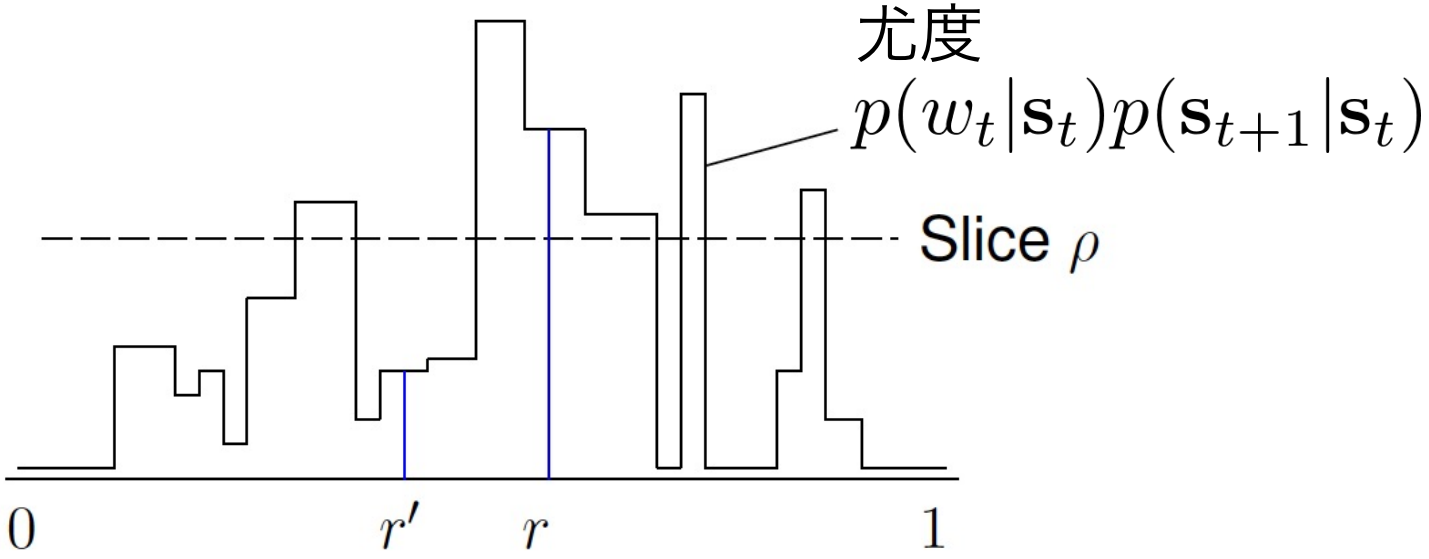



- 基本的な考え方:

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

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

# iTHMMの学習 (4)




- 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 $p = p(w_t|\mathbf{s}_t)p(\mathbf{s}_{t+1}|\mathbf{s}_t)$ から、スライス $\rho = p \cdot \mathrm{Unif}[0,1)$ を作る

(2) 一様乱数 r〜Unif(0,1] を引いて、対応するノード $\mathbf{s}$ を求める
- $\mathbf{s}$ が存在しなければ作成

(3) $p(w_t|\mathbf{s})p(\mathbf{s}_{t+1}|\mathbf{s}) > \rho$ なら $\mathbf{s}$ 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 $\mathbf{s}_{t-1}$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 $\pi^{\mathbf{s}}$ 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] | | | [2 7] | |
|---|---|---|---|---|
| know | 69 | 0.1976 | be | 80 |
| think | 41 | 0.1172 | have | 47 |
| say | 20 | 0.0568 | go | 14 |
| wish | 18 | 0.0489 | remember | 11 |
| wonder | 16 | 0.0431 | do | 11 |
| tell | 16 | 0.0453 | get | 11 |
| see | 14 | 0.0343 | take | 10 |
| do | 12 | 0.0357 | 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 | $M=3$ | **302.336** |
| | $\lambda=0.1$ | 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'wI'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”

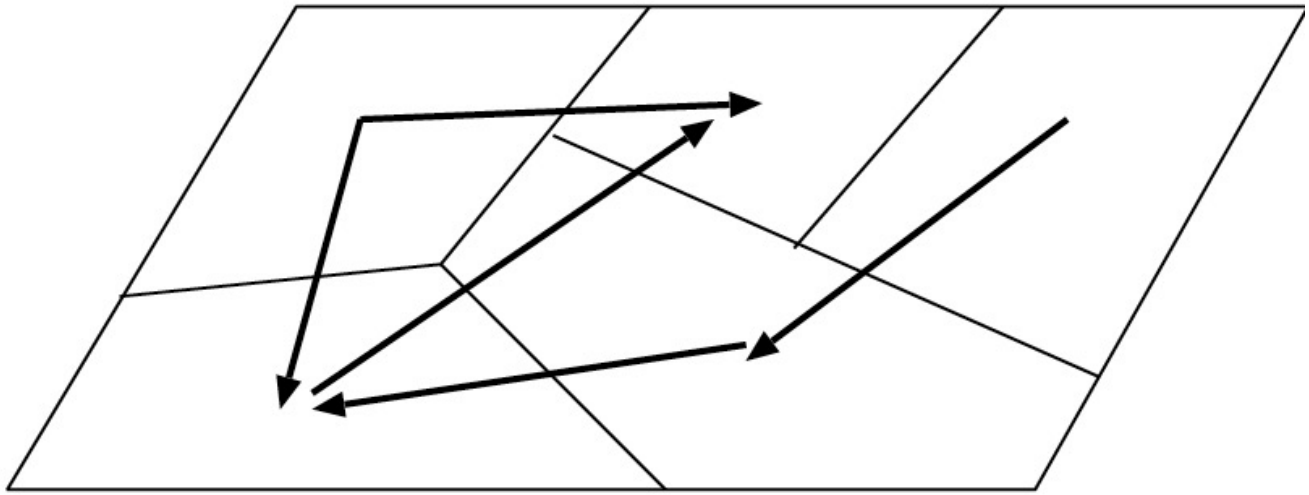
# 実験 (3)

| [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"

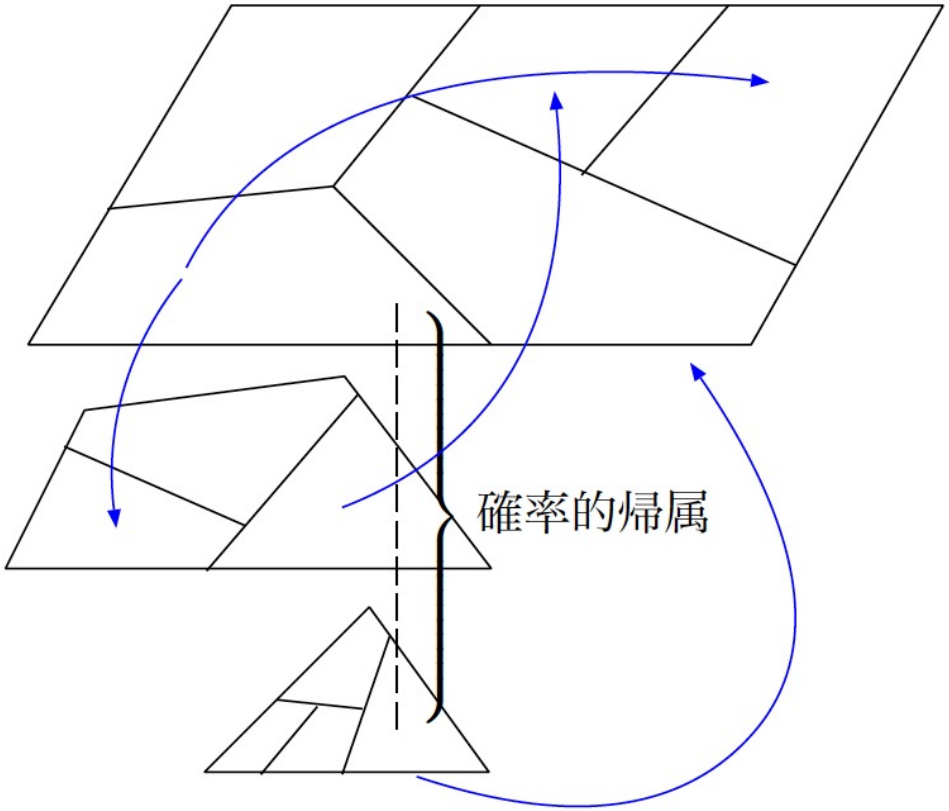
# 測度の空間と分割

- 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

- 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 (研究中)