無限木構造隠れMarkovモデルによる階層的品詞の教師なし学習

持橋 大地[1,a)]　能地 宏[2,b)]

概要：隠れ Markov モデル (HMM) は情報科学の基本的なモデルであるが, たとえば自然言語の品詞にみられるような階層的な状態を学習できないという問題があった. 本論文ではこれに対し, 木構造 Stick-breaking 過程 (Adams+ 2010) をそれ自体階層化することで, 無限の深さと幅を持つ隠れた木構造上での状態遷移確率と階層的な出力確率を持つ無限木構造隠れ Markov モデル (iTHMM) を提案する. これにより, 原理的に無限の複雑度を持つ隠れた木から, データに合わせた適切な状態の階層を学習することが可能となる. 英語および日本語のテキストで実験を行った. 提案法は自然言語処理に限らず, 情報科学一般に適用できる隠れ Markov モデルの本質的な拡張であり, PCFG など隠れ状態を持つ多くのモデルへの適用が期待できる.

キーワード：木構造 Stick-breaking 過程, 隠れマルコフモデル, ノンパラメトリックベイズ, 教師なし学習

The Infinite Tree Hidden Markov Model for Unsupervised Hierarchical Part-of-speech Induction

Daichi Mochihashi[1,a)]　Hiroshi Noji[2,b)]

Abstract: Hidden Markov models (HMM) is widely used in statistics and machine learning. However, it cannot learn latent states where these states are actually structured. Extending the tree-structured stick-breaking processes (Adams+ 2010) hierarchically as from DP to HDP, this paper proposes an Infinite Tree Hidden Markov models (iTHMM) whose states constitute a latent hierarchy. Experimental results on natural language texts show the validity of the proposed algorithm.

Keywords: Tree-structured stick-breaking process, Hidden Markov models, Nonparametric Bayes, Unsupervised learning

1. はじめに

隠れ Markov モデル (HMM)[1] は情報科学の基本的な統計モデルであり, 自然言語処理だけでなく, 音声認識, 経済学, 生態学, ロボティクス, バイオインフォマティクスのような多くの領域で, モデル化の重要な方法となっている [2].

特に自然言語処理においては, HMM は単語列が隠れ状態として品詞列を持つような形態素解析のモデルであり, 実際に初期の形態素解析 (茶筌) は HMM の教師あり学習として定式化されていた. さらに, 品詞自体を単語列のみから学習する教師なし品詞学習は 90 年代前半に始まり [3][4], 2000 年代に入ってベイズ学習によって高精度化され [5], 特に無限隠れ Markov モデル [6][7] によって品詞数も学習できるようになった. 半教師あり学習は先に教師なし学習のモデルを必要とするため, HMM は半教師あり学習においても不可欠なモデルである [8]. 2010 年には [9] によって経過がまとめられ, 研究は一見収束したかのように見える.

HMM は K 個 (無限 HMM では K を学習する) の整数で表される状態を持ち, この系列が観測値の裏に隠れているとしてそれを学習するものであるが, 実際の京大コーパス等で使われている品詞は, “名詞–固有名詞–地名” のように階層化されている. しかし, 通常の HMM では, こうした階層的な隠れ状態を教師なし学習することはできない. なぜならば, 隠れ変数の下に隠れ変数を考える場合,

- 何個の分岐を考えればよいのか
- どの深さまで階層を考えるべきなのか

について無限の可能性を考える必要が生じ, これらを全て数え上げることは不可能だからである. 具体的には, 各状態 $s_k \in \{1..K\}$ について, その一段階の細分化は $1..M_k$ 個の可能性があり, この細分化の数 $M_1 \cdots M_K$ は未知な上に, すべての状態は $\prod_{k=1}^{K} M_k$ 個に達し, これをさらに細分化する場合…を考えると, 無数のモデル選択問題と状態数の指数的増加に直面することになる.

構文解析の分野では, シンボル細分化 [10][11] によって名詞句や動詞句といった既知の文法的カテゴリを細分化す

[1] 統計数理研究所 数理・推論研究系
The Institute of Statistical Mathematics
[2] 奈良先端科学技術大学院大学 情報科学研究科
Nara Institute of Science and Technology
[a)] daichi@ism.ac.jp
[b)] noji@is.naist.jp

ることで, より高精度な学習を可能にしている. しかし, この場合でも細分化は上で述べた問題から 1 段階に限られており, また既知の品詞体系を必要とする. 未知の言語を解析する場合や, たとえば動詞句と形容詞句がより上の階層で統合されるような可能性も考えると, 計算言語学の立場からは, こうした品詞階層自体を言語データから学習できる統計的枠組が求められているといえる.

そこで本論文では, ノンパラメトリックベイズ法の立場から上の問題をすべて解決し, 隠れ状態が無限の分岐と無限の深さをもつ木構造上で定義される無限木構造隠れ Markov モデル (iTHMM) および, それに基づいた階層的な品詞の教師なし学習法を提案する. 提案法はディリクレ過程が木の縦方向の深さおよび横方向のそれぞれの分岐に存在する木構造 Stick-breaking 過程 [12] をそれ自体無限木構造上で階層化したものであり, こうして得られる無限木構造上の状態遷移確率と, この上で拡散過程として生成される出力確率分布によって観測系列が生成される. この iTHMM は自然言語処理に限らず, 情報科学一般に適用できる HMM の本質的な拡張であり, 多くの分野での適用が期待できる.

以下, 2 章で提案法の基礎となる無限隠れ Markov モデルおよびディリクレ過程, その具体的実現である Stick-breaking 過程について説明する. 3 章では木構造 Stick-breaking 過程 (TSSB) とそのポリアの壺表現について説明し, 4 章で TSSB を階層化した階層的木構造 Stick-breaking 過程 (HTSSB) とそれに基づいた無限木構造隠れ Markov モデルと特別な MCMC 法による学習について述べる. 5 章で HHMM などの関連研究との違いについて述べた後, 6 章で日本語や英語のテキストに対して実験を行って優位性を示し, 特に半教師あり学習に用いることも可能であることを示す. 7 章で展望を示し, 全体をまとめる.

2. 無限隠れ Markov モデルと Stick-breaking 過程

HMM は図 1 のように, 観測列 $\mathbf{w} = w_1 w_2 \cdots w_T$ の背後に隠れ状態列 $\mathbf{s} = s_1 s_2 \cdots s_T$ があり, $\mathbf{s}$ から $\mathbf{w}$ が生成されたとする確率モデルである. 1 次の HMM では時刻 t での状態 s_t は一つ前の状態 s_{t-1} のみに依存すると考え, $\mathbf{w}$ と $\mathbf{s}$ が生成される同時確率は

$$p(\mathbf{w}, \mathbf{s}) = \prod_{t=1}^{T} p(w_t|s_t) p(s_t|s_{t-1}) \qquad (1)$$

で表される. ただし, s_0 は初期状態である. 隠れ状態を名詞や動詞のような品詞とみなすと, これは品詞学習のモデ

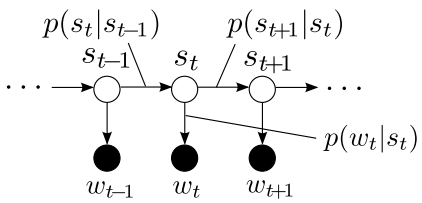

図 1 隠れ Markov モデルの構造. ●は観測値を, ○は未知の確率変数を表す.

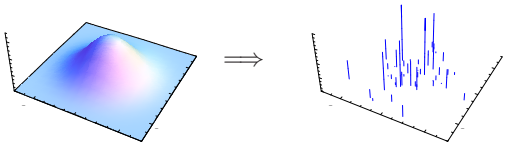

図 2 ディリクレ過程による基底測度 G_0 からの G の生成.

ルであり, HMM は初期の形態素解析 (茶筌) に使われたほか, 現在でも半教師あり学習に用いられている [8].

品詞の教師なし学習は最初は最尤推定 (EM アルゴリズム) によっており性能が低いとみなされていたが [3], Goldwater ら [5] はこれを MCMC 法によりベイズ推定することで, 局所解を避け, 高精度な解が得られることを示した. これらの研究では状態数＝品詞数 K は既知であるとしているが, この K も学習できるのが無限隠れ Markov モデル (Infinite HMM, iHMM)[6][13] である.

2.1 iHMM と HDP

まず, HMM では生成モデルから, 状態は状態遷移確率 $p(s_t|s_{t-1})$ によって生成されることに注意しよう. 通常の HMM では, これは決まった K 個の状態への確率分布となるが, iHMM では, これが可算無限個の要素を持つディリクレ過程から生成されたと考える. ディリクレ過程とは, 図 2 のように基底測度とよばれる親の分布 G_0 に似た無限次元の離散的な測度を生成する確率過程であり,

$$G \sim \mathrm{DP}(\alpha, G_0) \qquad (2)$$

と書かれる. 集中度パラメータ $\alpha > 0$ が大きいほど G は G_0 に似たものとなるが, 期待値は常に $E[G] = G_0$ である.

ただし, 各状態 k で別々にこの遷移確率 G_k を G_0 からサンプルすると, 他の状態との重なりが 0 になってしまい, HMM の状態が共有されなくなってしまう. そこで, iHMM ではまず全体の離散的な $G \sim \mathrm{DP}(\eta, H)$ をサンプルし, これを基底測度として各 $G_k \sim \mathrm{DP}(\alpha, G)$ $(k = 1 \cdots \infty)$ を生成する階層ディリクレ過程 (HDP) によって, 遷移する状態を共有し, その事前分布を G で与える. このとき α によって, G_k が事前分布 G と平均的にどれほど似ているかが制御されることになる.

2.2 Stick-breaking 過程と CDP 表現

上ではディリクレ過程およびそれに基づく iHMM の構成を測度論的に述べた. よく知られているように, ディリクレ過程に基づく $G \sim \mathrm{DP}(\alpha, G_0)$ からのサンプルは図 3 のような CRP(中国料理店過程) で表すことができる [14]. ここでは, G からのサンプル $x_1, x_2, \cdots, x_n$ が与えられたとき, 次の x_{n+1} のとる値の確率は G を積分消去することにより,

$$p(x_{n+1}|x_1 \cdots x_n) = \int p(x_{n+1}|G)\, p(G|x_1 \cdots x_n) dG \qquad (3)$$

$$= \begin{cases} n_k/(n+\alpha) & (k = 1, \cdots, K) \\ \alpha/(n+\alpha)\, G_0(x_{n+1}) & (k = K+1) \end{cases} \qquad (4)$$

となることを利用している．ここで K は $x_1 \cdots x_n$ の中での値の異なり数, n_k は k 番目の値が現れた回数である．これから, G からのサンプルを図 3 における客とみなし, (4) 式に従って k で番号づけられるテーブルに順番に着席する CRP が得られる．

CRP では G は積分消去されていたが, G は実際に, 次のような Stick-breaking(棒折り) 過程で明示的に生成することができる [15].

$$\gamma_k \sim \mathrm{Be}(1, \alpha) \tag{5}$$

$$\pi_k = \gamma_k \prod_{j=1}^{k-1} (1 - \gamma_j) \tag{6}$$

$$G = \sum_{k=1}^{\infty} \pi_k \delta(\theta_k), \qquad \theta_k \sim G_0 \ . \tag{7}$$

ここで $\mathrm{Be}(\alpha, \beta)$ はベータ分布, $\delta(x)$ は点 x のみで測度 1 となる離散測度を表す．これは図 4 のように, 長さ 1 の棒を次々と γ_k $(k = 1, 2, 3, \cdots)$ の割合で折り, その左端の長さ π_k の棒を基底測度 G_0 からランダムにサンプルした位置 θ_k に立てていったものが G であることを意味している (図 2). $\{\theta_k\}_{k=1}^{\infty}$ が定まれば, G を特徴づけるのは無限次元の多項分布 $\pi = (\pi_1, \pi_2, \cdots)$ であり, これを GEM 分布, あるいは本論文では SBP(α) とよぶ.

CDP 表現　SBP はベータ分布の確率変数 γ_k の積で定義されるから, π からの実現値 $\mathcal{D} = \{x_1, x_2, \cdots\}$ が与えられたとき, π の事後分布は各 γ_k の事後分布の積で表現することができる.

すなわち, (6) 式は k 番目の値が選ばれる確率 π_k は, 各 x_n が $1 \cdots k{-}1$ 番目まで折った棒の右側を選びつづけ, 最後に k 番目で左側を選んだ確率と等しいことを意味するから, γ_k の事後分布は $\mathcal{D}$ の中で k で止まった回数を $n_0(k)$, 止まらず折り続けた回数を $n_1(k)$ とすれば, $\mathrm{Be}(1{+}n_0(k), \alpha{+}n_1(k))$ であり, 期待値は

$$E[\gamma_k | \mathcal{D}] = \frac{1{+}n_0(k)}{1{+}\alpha{+}n_0(k){+}n_1(k)} \tag{8}$$

と計算できる．したがって, π_k の事後確率の期待値は

$$E[\pi_k | \mathcal{D}] = \frac{1{+}n_0(k)}{1{+}\alpha{+}n_0(k){+}n_1(k)} \prod_{j=1}^{k-1} \frac{\alpha{+}n_1(j)}{1{+}\alpha{+}n_0(j){+}n_1(j)} \tag{9}$$

となる.

この π_k は, 図 5 のように領域 1 の中に領域 2 があり, さらにその中に領域 3 が…と入れ子になっているとき, 各領域の入口に門番が立っており, これまでに門を通過した人数と止めた人数を数えて確率 (8) によってランダムに客を

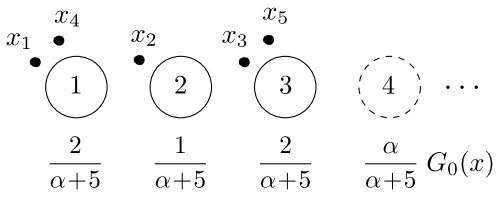

図 3　CRP(中国料理店過程) による客の配置.

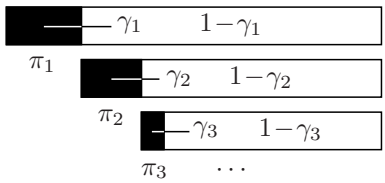

図 4　Stick-breaking 過程による $\pi = (\pi_1, \pi_2, \cdots)$ の生成. 長さ 1 の棒をベータ分布に従う $\gamma_k \sim \mathrm{Be}(1, \alpha)$ で次々と折り, 無限次元の多項分布 π を生成する.

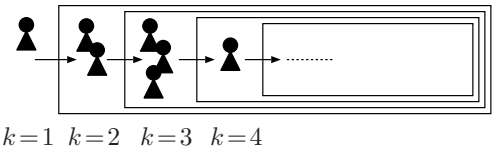

図 5　Chinese District Process (CDP) [16]. 無限にネストした各領域について, そこを通過した人数と止まった人数が数えられており, 番人が確率的に客を止める. これは SBP をポリアの壺として表現したものである.

止める場合に, 領域 k で止まる確率と等しい．このことから, 上の過程は Chinese District Process (CDP) と呼ばれており [16], これは SBP の CRP 表現であるといえる.

2.3　学習例と問題

SBP では G が明示的に表現されるため, HMM での取り扱いが簡単になるといった長所があり, 実際に [16] では CDP により HMM を表現し, Gibbs サンプリングおよび変分ベイズ法による学習を行っている．さらに, Gael らはスライスサンプリング [17] を用いることで, (7) 式の和を打ち切ることなく動的計画法によるサンプリングを可能にする無限隠れ Markov モデルの学習法を示した [7]. 図 1 に, この iHMM で『不思議の国のアリス』(1431 文, 26689 語) を学習した際の, 各潜在状態からの出力確率の上位語を示す.

ここではデータが小さいため, K はほぼ 7 と学習されている．図から, 状態 1=名詞, 状態 2=冠詞, 状態 3=動詞といった品詞が, まったく人手を介することなく自動的に学習されていることが見てとれる.

これらの方法はすべて, 品詞, すなわち HMM の状態が名詞, 動詞, 形容詞, …のようにフラットであることを前提にしている．しかし, 実際の品詞は京大コーパスにおいても「助動詞–ナ形容詞–語幹」のように階層化されており,[*1] しかも, こうした人手による階層が最適であることは何ら保証されていない. 名詞–固有名詞–地名という既存の分類以外にも, 名詞–抽象名詞–心理状態 (嬉しさ, 悲しさなど) といった分類も適切かもしれない．しかしながら, こうした階層を教師なしで学習するためには, 隠れ変数の下に隠れ変数があり, さらにその下に…という無限に続く統計モデルが必要であり, はじめに述べたように, この問題は通常の方法では解くことができない．これを可能にするのが, 木構造 Stick-breaking 過程 [12] である.

[*1] 状態が木構造で表現されるのではなく, 隠れた素性の組み合わせ, すなわちベータ過程 [18] によって表すことも考えられる．しかし, ベータ過程について AR 的でない任意の遷移を許す統計モデルはまだ提案されていない.

1		2		3	
she	432	the	1026	was	277
to	387	a	473	had	126
i	324	her	116	said	113
it	265	very	84	be	77
you	218	its	50	is	73
alice	166	my	46	went	58
and	147	no	44	were	56
they	76	his	44	see	52
there	61	this	39	could	52
he	55	an	37	know	50
that	39	your	36	thought	44
who	37	as	31	herself	42

4		5		6	
and	466	way	45	little	92
of	343	mouse	41	great	23
in	262	thing	39	very	22
said	174	queen	37	long	22
to	163	head	36	large	22
as	163	cat	35	right	20
that	125	hatter	34	same	17
for	123	duchess	34	good	17
at	122	well	31	white	11
but	121	time	31	other	11
with	114	tone	28	poor	10
on	83	rabbit	28	first	10

表 1　『不思議の国のアリス』で iHMM の隠れ状態に割り当てられた単語とその回数.

3. 木構造 Stick-breaking 過程とその学習

木構造 Stick-breaking 過程 (Tree-structured Stick-breaking process, TSSB) [12] は階層クラスタリングのために提案されたベイズ事前分布であり, 原理的に無限の深さと無限の分岐を持つ木構造を離散確率分布として生成する確率過程である. TSSB により, 深さや分岐の数が場所によって異なり, データによって決まる階層クラスタリングが可能になる. またこれは, 著者による無限 Markov モデル [19] の一般化ともみることができる.

階層クラスタリングのモデルで最も簡単なのは, 先の SBP で生成された無限個の棒 π_k をさらに SBP で分割し, それをさらに…と無限に分割していく方法であろう. これは Polya 木 [20] とよばれている. しかし, この方法ではデータは最も細分化された末端のカテゴリにだけ存在することになり, 中間の一般的なカテゴリに存在することはできない. Nested CRP [21][22] または Nested HDP [23] ではさらに木の深さに対して別の DP を事前分布とすることでこれを許しているものの, この事前分布は分割とは別であり木の場所によらないため, 現実のように一部のノードが特に深くなる様子を表現することができず, 状態数の指数的増加を抑えられないという問題がある.

これに対し, TSSB では棒を単に再帰的に分割するのではなく, 先に「そのカテゴリで止まる確率」を導入する. 具体的には, 長さ 1 の棒から始めて

$$\nu \sim \mathrm{Be}(1, \alpha) \tag{10}$$

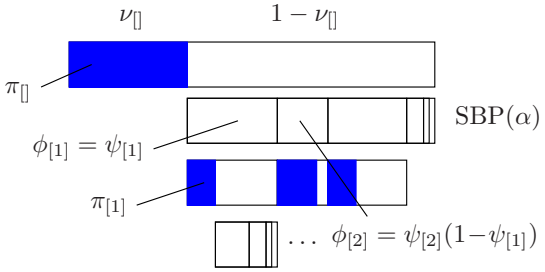

図 6　ベータ分布に従う確率変数 $\nu_{\mathbf{s}}, \psi_{\mathbf{s}}$ による TSSB π の構成.

で左端を折り, このノードに止まる確率を生成する. 止まらない場合は, 残った長さ $(1-\nu)$ の棒を SBP(γ) で分割し, 子供であるそれらの各棒に同じ操作を再帰的に繰り返す (図 7).

こうして得られる TSSB の各ノードは, 可変長の整数列 $\mathbf{s} = s_1 s_2 s_3 \cdots$ でインデックスされる. たとえば, ノード $\mathbf{s} = []$ (空列) は木構造の根ノードを, $\mathbf{s} = [2\ 4\ 1]$ は根から 2 番目の子供→ 4 番目の子供→最初の子供と順にたどったノードを表している. 木構造なので, 各分岐を表す整数の意味は木構造上の場所によって異なることに注意しよう. たとえば, 動詞の 3 番目の細分と名詞の 3 番目の細分の意味は, もちろん異なっている.

3.1 TSSB の定義

上の TSSB は, 次のようなポリアの壺で表すことができる. まず, 無限木構造のすべてのノード $\mathbf{s}$ について, 確率 $\nu_{\mathbf{s}} \sim \mathrm{Be}(1, \alpha)$ および $\psi_{\mathbf{s}} \sim \mathrm{Be}(1, \gamma)$ が生成される. 客が木の根ノード [] に到着すると, $\nu_{[]}$ の確率で表が出るコインを投げ, 表が出れば客はここに止まり, 裏が出れば子供に降りることにする. どの子供に降りるかは, SBP(α) で決定される. すなわち, CDP に従って子供を [1], [2], [3], $\cdots$ と順番に訪れ, $\psi_{[k]}$ のコインを投げて表が出ればその子供を選び, 裏が出れば次の子供に進む. こうして選ばれた子供 $[k]$ に降り, そこに止まるかどうか $\nu_{[k]}$ のコインを投げ…という操作を, この客が止まるまで再帰的に繰り返す.

いま, 根ノード [] には止まらず, 次に子供 [1], [2], [3] は通過して [4] で止まったとしよう. 次に, [4] で止まるかどうかを $\nu_{[4]}$ で決め, 止まらなければ, [4 1], [4 2] と順に訪れ, たとえば [4 2] で終わると次にここで止まるかを $\nu_{[4\ 2]}$ で決め, 表が出ればこの客は [4 2] に追加されることになる.

これから, 数学的には TSSB は下のように定義することができる. $\pi_{\mathbf{s}}$ を TSSB π においてノード $\mathbf{s}$ に止まる確率とし, $\mathbf{s}' \prec \mathbf{s}$ は木構造上で $\mathbf{s}'$ が $\mathbf{s}$ の親ノードにあることを表すものとすると,

$$\pi_{\mathbf{s}} = \nu_{\mathbf{s}} \phi_{\mathbf{s}} \prod_{\mathbf{s}' \prec \mathbf{s}} \phi_{\mathbf{s}'} (1 - \nu_{\mathbf{s}'}) \tag{11}$$

$$= \nu_{\mathbf{s}} \prod_{\mathbf{s}' \prec \mathbf{s}} (1 - \nu_{\mathbf{s}'}) \cdot \prod_{\mathbf{s}' \preceq \mathbf{s}} \phi_{\mathbf{s}'} \tag{12}$$

となり, ここで

$$\nu_{\mathbf{s}} \sim \mathrm{Be}(1, \alpha), \quad \psi_{\mathbf{s}k} \sim \mathrm{Be}(1, \gamma) \tag{13}$$

$$\phi_{\mathbf{s}k} = \psi_{\mathbf{s}k} \prod_{j=1}^{k-1} (1 - \psi_{\mathbf{s}j}) \tag{14}$$

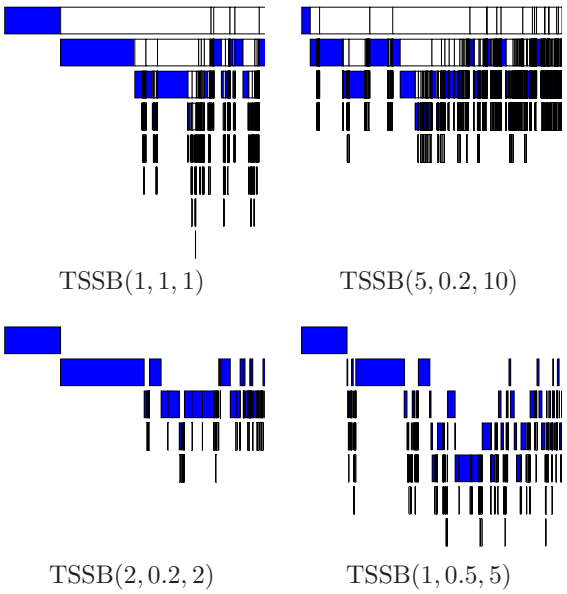

図 7　様々なパラメータから生成した TSSB($\alpha_0, \lambda, \gamma$). 2 行目では Stick-breaking の切れ目を省略した. 無限次元の離散分布が構造を持ち, かつ総和が 1 になっている様子が見てとれる.

である.

(12) 式および SBP の定義 (6) 式から, これは ν で定義される縦方向の SBP すなわちディリクレ過程と, ψ で定義される横方向のディリクレ過程の積になっていることがわかる. 図 7 に, こうしてランダムに生成された TSSB の例を示した. 実際には (12) 式だけでは木が深くなりすぎるため, ノードが深くなるほど止まる確率が上がるよう, (13) 式の α を [12] と同様に

$$\alpha(\mathbf{s}) = \alpha_0 \cdot \lambda^{|\mathbf{s}|} \tag{15}$$

とし, パラメータ $0 < \lambda \leq 1$ によって減衰率の事前分布をコントロールする. ここで, $|\mathbf{s}|$ はノード $\mathbf{s}$ の深さである. ただし (15) 式はあくまで平均的な事前確率であり, 実際にはノードは場所によって深くなることも浅くなることもあることに注意されたい.

全体として, TSSB のパラメータは $(\alpha_0, \lambda, \gamma)$ であり, この値によって図 7 のような様々な木構造が得られる.

3.2　TSSB の CDP 表現

TSSB は (12) 式より SBP の積となっているから, TSSB における客の追加は複数の CDP で表される. 前節のポリアの壺の議論から, 客があるノード $\mathbf{s} = s_1 s_2 \cdots s_n$ に到達

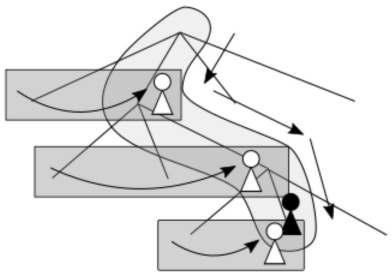

図 8　TSSB の CDP. 黒の客に対応する深さ方向の CDP と, 白の客に対応する各分岐の CDP にそれぞれ客を追加/削除する.

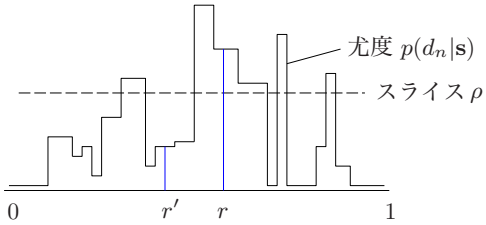

図 9　Slice sampling+Retrospective sampling による TSSB からの MCMC サンプリング. 一様乱数 r に対応する TSSB のノード $\mathbf{s}$ において, それぞれ尤度 $p(d_n|\mathbf{s})$ が存在する.

したことは

- $\mathbf{s}$ に至る横方向の各 CDP において $s_1, \cdots, s_n$ を選び,
- $\mathbf{s}$ から根に至る縦方向の CDP において $\mathbf{s}$ で止まった

ことを意味するから, これは図 8 のように, 深さを表す縦方向の CDP と各分岐を表す横方向の CDP にそれぞれ客, すなわちカウントが追加されたことを表す.

このとき, 縦の CDP でノード $\mathbf{s}$ を垂直に通過した客の数を $n_1(\mathbf{s})$, 止まった数を $n_0(\mathbf{s})$ とし, 横の CDP でノード $\mathbf{s}$ を水平に通過した客の数を $m_1(\mathbf{s})$, 止まった数を $m_0(\mathbf{s})$ とおけば, $\nu_{\mathbf{s}}$, $\psi_{\mathbf{s}k}$ の事後分布の期待値は同様に

$$E[\nu_{\mathbf{s}}|\mathcal{D}] = \frac{1+n_0(\mathbf{s})}{1+\alpha+n_0(\mathbf{s})+n_1(\mathbf{s})} \tag{16}$$

$$E[\psi_{\mathbf{s}k}|\mathcal{D}] = \frac{1+m_0(\mathbf{s})}{1+\gamma+m_0(\mathbf{s})+m_1(\mathbf{s})} \tag{17}$$

となり, これから (12) 式で $\pi_{\mathbf{s}}$ を計算することができる. 原論文 [12] では横方向の SBP を CRP として表現しているが, こうして全て CDP で表せることに注意されたい.

3.3　無限階層クラスタリング

TSSB によるベイズ無限階層クラスタリングでは, N 個のデータ $d_1 \cdots d_N$ が与えられたとき, それぞれの d_n を TSSB のどれかのノード $\mathbf{s}_n$ に割り当てる. これには Gibbs サンプリングにより, 図 10 のようなアルゴリズムで $p(\mathbf{s}_n|d_n) \propto p(d_n|\mathbf{s})\pi_{\mathbf{s}}$ に従った確率でランダムに $\mathbf{s}_n$ をサンプリングしていけばよい.

しかし, TSSB では通常の混合モデルと異なり, ノード $\mathbf{s}$ が木構造化されて無限に存在するため, 端からサンプルする対象を数え上げることはできない. ここで, TSSB の構成から図 7 のように, $\mathbf{s}$ はその確率 $\pi_{\mathbf{s}}$ の長さで $[0, 1)$ の中の区間を占めていることに注意しよう. ゆえに, $\pi_{\mathbf{s}}$ に従ってランダムに $\mathbf{s}$ をサンプルするには, まず一様乱数 $r = \mathrm{Unif}[0, 1)$ をサンプルしてから, TSSB の中で r に対応するノードを探せばよい. 先に乱数を決めてからそれに対応する候補を選ぶこの方法は, Retrospective sampling [24] とよばれている.

図 9 のように $p(d_n|\mathbf{s})\pi_{\mathbf{s}}$ に従って $\mathbf{s}_n$ をサンプルするためには, スライスサンプリングと併用すれば, まず現在のノード $\mathbf{s}_n$ での密度 $p(d_n|\mathbf{s}_n)\pi_{\mathbf{s}_n}$ と 0 の間の一様分布からサンプリングしてスライス ρ を作り, $p(d_n|\mathbf{s})\pi_{\mathbf{s}} > \rho$ となる $\mathbf{s}$ から一様に選べばよい. これは上の Retrospective sampling でまずランダムに $\mathbf{s}$ を選び, これがスライスより上になる

まで繰り返せば得られる. 実際には $[0,1)$ の間の二分探索に似た方法で効率的にサンプリングできるが, 詳細は後の図 12 または [12] を参照されたい.

4. 無限木構造隠れ Markov モデル

木構造 Stick-breaking 過程により, 無限の深さと分岐をもつ木構造上での階層クラスタリングを行うことができる. ここで木構造のノードは階層化されたクラスタを表しているから, これを時系列に展開して隠れ状態とみなせば, 無限木構造を状態空間にもつ隠れ Markov モデルが原理的に構成できるはずである.

4.1 木構造上の状態遷移確率

ただし, 時系列モデルの HMM とするためには, 状態から状態への遷移確率, すなわち木構造のノード間の遷移確率を定義しなければならない. K 個の状態からなる通常の HMM では, これは各行が次の K 個の状態への遷移確率分布からなる $K \times K$ の遷移行列で簡単に表すことができる. しかし, いま状態は木構造をなしているから, これは無限の木構造の各ノードに, 次の時刻の無限の木構造のノード上への遷移確率分布が必要となることを意味している. この分布は TSSB で表すことができるから, これはすなわち, TSSB の無限個の各ノード $\mathbf{s}$ にそれぞれ, 次のノードへの状態遷移確率を表す TSSB があることを示している. これを上つき添字を使って, $\pi^{\mathbf{s}}$ と書くことにしよう (図 11).

ただし, ノードは木構造をなしているから, 各ノードからの遷移を表す $\pi^{\mathbf{s}}$ は独立ではなく, 親子間の依存関係を持っているはずである. たとえば, ノード $\mathbf{s} = [2\ 3]$ が「名詞–固有名詞」に相当するノードであったとしよう. このとき, $[2\ 3]$ からの状態遷移 $\pi^{[2\ 3]}$ は親ノードである $[2]$, つまり「名詞」からの遷移確率 $\pi^{[2]}$ を反映しており, それはさらに状態全体の遷移の事前確率 $\pi^{[]}$ (冠詞には遷移しやすいが, 感動詞へは遷移しにくいなど) を反映しているはずである.

4.2 階層的 TSSB

そこで, 本研究では $\pi^{\mathbf{s}}$ を独立とするのではなく, 親の TSSB $\pi^{\mathbf{s}'}$ からそれ自体階層的に生成することを考える.

3.1 節で述べたように, TSSB は縦方向および横方向の無数の Stick-breaking 過程, すなわちディリクレ過程の積となっているから, これには $\pi^{\mathbf{s}}$ を構成するそれぞれの DP を, 対応する $\pi^{\mathbf{s}'}$ の DP から生成する階層ディリクレ過程を考えればよい. 具体的には, $\pi = \mathrm{SBP}(\gamma)$ で表されるディリクレ過程が SBP $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots)$ で表されるディリクレ過程から

1: **for** iter $= 1 \cdots$ iters **do**
2: **for** n in randperm$(1 \cdots N)$ **do**
3: $p(d_n|\mathbf{s}_n)$ から d_n, π から $\mathbf{s}_n$ を削除.
4: Draw $\mathbf{s}_n \propto p(d_n|\mathbf{s}_n)\pi_{\mathbf{s}_n}$
5: $p(d_n|\mathbf{s}_n)$ に d_n, π に $\mathbf{s}_n$ を追加.
6: **end for**
7: **end for**

図 10 TSSB による無限階層クラスタリングの Gibbs サンプリング.

$$\pi \sim \mathrm{DP}(\alpha, \boldsymbol{\beta}) \tag{18}$$

と生成されるとき, HDP の Stick-breaking 表現から, π を構成する確率変数 γ_k $(k = 1, 2, \cdots)$ の分布は

$$\gamma_k \sim \mathrm{Be}\Big(\alpha\beta_k, \alpha\Big(1 - \sum_{j=1}^{k} \beta_j\Big)\Big) \tag{19}$$

となるから [13], われわれの場合, ノード $\mathbf{s}$ での ν, ψ の分布は階層的に

$$\nu_{\mathbf{s}} \sim \mathrm{Be}\Big(\alpha\nu'_{\mathbf{s}}, \alpha\Big(1 - \sum_{\mathbf{u} \preceq \mathbf{s}_k} \nu'_{\mathbf{u}}\Big)\Big), \tag{20}$$

$$\psi_{\mathbf{s}k} \sim \mathrm{Be}\Big(\alpha\psi'_{\mathbf{s}k}, \alpha\Big(1 - \sum_{j=1}^{k} \psi'_{\mathbf{s}j}\Big)\Big) \tag{21}$$

と与えられる. ここで $\nu'_{\mathbf{s}}$, $\psi'_{\mathbf{s}k}$ は親の TSSB における $\nu_{\mathbf{s}}$, $\psi_{\mathbf{s}k}$ の値である. 根ノードでは親がないため, (13) 式によって $\nu_{\mathbf{s}}$, $\psi_{\mathbf{s}k}$ を生成する. このとき, 客が与えられた後の事後確率の期待値は (16) (17) 式と同様にして,

$$E[\nu_{\mathbf{s}}|\mathcal{D}] = \frac{\alpha\nu'_{\mathbf{s}} + n_0(\mathbf{s})}{\alpha(1 - \sum_{\mathbf{u} \prec \mathbf{s}} \nu'_{\mathbf{u}}) + n_0(\mathbf{s}) + n_1(\mathbf{s})} \tag{22}$$

$$E[\psi_{\mathbf{s}k}|\mathcal{D}] = \frac{\alpha\psi'_{\mathbf{s}k} + m_0(\mathbf{s}k)}{\alpha(1 - \sum_{j=1}^{k-1} \psi'_{\mathbf{s}j}) + m_0(\mathbf{s}k) + m_1(\mathbf{s}k)} \tag{23}$$

となる. 上の確率は親の TSSB の $\nu'_{\mathbf{s}}$, $\psi'_{\mathbf{s}k}$ の値に依存し, それはさらにその親の $\nu''_{\mathbf{s}}$, $\psi''_{\mathbf{s}k}$ に依存し…と再帰的な計算が必要となることに注意しよう. トップレベルの TSSB では, 値は (16)(17) 式で与えられる.

なお, HDP において (19) 式の事後確率の期待値を変形すると, $n = n_0 + n_1$, $\beta_k^l = \sum_{j=k}^{l} \beta_j$ として

$$E[\gamma_k|n_0, n_1] = \frac{\alpha\beta_k + n_0}{\alpha\beta_k^{\infty} + n} \tag{24}$$

$$= \frac{\alpha\beta_k^{\infty}}{\alpha\beta_k^{\infty} + n} \cdot \frac{\alpha\beta_k}{\alpha\beta_k^{\infty}} + \frac{n}{\alpha\beta_k^{\infty} + n} \cdot \frac{n_0}{n} \tag{25}$$

$$= \mu \cdot \hat{p} + (1 - \mu) \cdot \bar{p} \tag{26}$$

ただし

$$\hat{p} = \frac{n_0}{n}, \quad \bar{p} = \frac{\beta_k}{\beta_k^{\infty}} \tag{27}$$

$$\mu = \frac{n}{\alpha\beta_k^{\infty} + n} \tag{28}$$

と書けるから, これは現在のノードでの Bernoulli 分布の最尤推定値 $\hat{p}$ と親 TSSB での期待値 $\bar{p}$ を割合 μ で線形補間したものとみることができる. n が大きいほど μ の値は大きくなるから, (26) 式は現在のノードのカウントが大きいほどノードでの推定値を, 小さいほど親ノードでの期待値を使うベイズ的な適応補間になっていることがわかる. 同様の構造が提案法にもあり, このときさらに α によって, 親の情報をどれほど受け継ぐのかが制御される.

提案法では, こうして生成された π からさらに π' が生成され…と, 無限木構造上で π 自体が階層的に生成される. (16)(17) 式で定義されるこの過程を, 階層的木構造 Stick-breaking 過程 (HTSSB) [25][*2] と呼ぶことにし,

[*2] [25] で概略のみ提案されている方法では TSSB を構成するベータ分布を独立に扱っており, HDP に基づく本論文とは異なる.

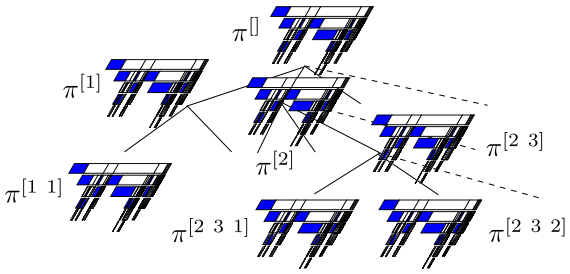

図 11 HTSSB の概念図. 無限個の分岐を持つ木構造の各ノードに次の時刻でのノードへの状態遷移を表す TSSB があり, 親から階層的に生成されている. この木構造自身と, TSSB の持っている木構造は自己同型になっている.

$$\pi \sim \mathrm{HTSSB}(\alpha, \pi_0) \tag{29}$$

と書くことにする. HTSSB に基づく無限木構造上の隠れ Markov モデルを HTSSB-HMM, または iTHMM (Infinite Tree HMM, 無限木構造隠れ Markov モデル) と呼ぶことにする.

iTHMM の生成モデル iTHMM では, 状態は次のようにして生成される. まず, トップレベルの $\pi^{[]} \sim \mathrm{TSSB}(\alpha_0, \lambda, \gamma)$ を生成する. 次にこれを親として, $\pi^{[1]} \sim \mathrm{HTSSB}(\alpha, \pi_{[]})$, $\pi^{[2]} \sim \cdots$が生成され, 次にそれらの子供である $\pi^{[1\ 1]} \sim \mathrm{HTSSB}(\alpha, \pi^{[1]})$, $\pi^{[1\ 2]} \sim \cdots$が無限に生成される.

次にある初期ノード $\mathbf{s}_0$ から始め, (1) 式の HMM の生成モデルに従って状態 $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \cdots$ 及び, それらからの出力 $w_1, w_2, w_3, \cdots$ が得られる.

なお, この HTSSB は各ノードの持っている π 自体が, ノードのなす無限木構造と同型であるという自己相似構造を持っていることに注意されたい. こうして TSSB 自体を階層的に生成することにより, HDP や階層 Pitman-Yor 過程と同様に, 現在の TSSB のノード $\mathbf{s}$ に信頼できる確率を計算できる充分なカウントがなくても, 親 TSSB での同じノードの確率と再帰的に混合することにより, より安定した推定値が得られることも利点の一つである.

4.3 iTHMM と HCDP

TSSB の事後確率は 3 章の CDP で求めることができる. それでは iTHMM, すなわち HTSSB の事後確率はどうやって求めればよいだろうか.

ここで HTSSB では, TSSB を構成する個々のディリクレ過程が対応する親から引き継いだ階層ディリクレ過程であったことを思い出そう. ディリクレ過程は CRP としても表せるから, たとえば垂直な ν の SBP において深さ k のノード $\mathbf{s}$ に客が追加されたとき, この客が自分の分布から生成されたか, それとも親の基底測度から生成されたかの確率は (4) 式で与えられる. ゆえに, $\mathbf{s}$ に着いた垂直の SBP の客の総数を前節の定義を使って $n(\mathbf{s}) = n_0(\mathbf{s}) + n_1(\mathbf{s})$ と表すと,

$$\left[\frac{n(\mathbf{s})}{n(\mathbf{s})+\alpha}, \frac{\alpha}{n(\mathbf{s})+\alpha} \nu'_{\mathbf{s}} \right] \tag{30}$$

に比例する Bernoulli 試行で後者が出た場合 [*3], 客を基底測度, すなわち親の TSSB の同じノードに追加すればよい.

同様にして水平な ψ の SBP も CRP とみなせるから, ノード $\mathbf{s}$ に水平の客が追加されたとき, $m(\mathbf{s}) = m_0(\mathbf{s}) + m_1(\mathbf{s})$ とおいて

$$\left[\frac{m(\mathbf{s})}{m(\mathbf{s})+\alpha}, \frac{\alpha}{m(\mathbf{s})+\alpha} \psi'_{\mathbf{s}} \right] \tag{31}$$

に比例する Bernoulli 試行を行い, 後者が出た場合に客を親の TSSB に追加する. これを HCDP(階層的 CDP) とよぶことにする.

このとき, 親の TSSB においても同様にして, さらにその親へと再帰的に客が追加される可能性があることに注意しよう. 上の Bernoulli 試行はカウント $n()$ が 0 のとき必ず後者を返すから, 初めてのノードに客が追加された際には, 自動的に上のノードも作成されて最初の客が追加されることになる. また, 削除の際には上の過程を逆にたどることで, 必要に応じて再帰的に客を TSSB から削除する.

4.4 iTHMM の学習

こうして無限木構造上の状態遷移確率を HTSSB から計算し, 更新できるようになったので, これに基づいて iTHMM の学習を行うことができる. iTHMM の隠れ状態 $\mathbf{s}_t$ は [4 2 3] のように構造化されているもの, HMM としての構造は図 1 と同じである. よって, Gibbs サンプリングを用いれば, 学習には観測値 (単語) w_t について, その隠れ状態 $\mathbf{s}_t$ を確率

$$p(\mathbf{s}_t | w_t, \mathbf{w}_{-t}, \mathbf{s}_{-t}) \propto p(w_t|\mathbf{s}_t)\, p(\mathbf{s}_{t+1}|\mathbf{s}_t)\, p(\mathbf{s}_t|\mathbf{s}_{t-1}) \tag{32}$$

に比例して次々とサンプルすればよい [5]. ここで第 1 項は後で述べるように状態 $\mathbf{s}_t$ から単語 w_t が生成される出力確率, 第 2 項と第 3 項は HTSSB で計算される状態遷移確率である. また, $\mathbf{w}_{-t}$ は w_t 以外のすべての観測値, $\mathbf{s}_{-t}$ は $\mathbf{s}_t$ 以外のすべての隠れ状態を表す.

ただし, iTHMM では $\mathbf{s}_t$ は $[s_1 s_2 s_3 \cdots]$ のように構造化された無限個の分岐と深さを持っており, 通常の HMM のように $1 \cdots K$ の有限個, あるいは iHMM のように確率的打ち切りにより簡単に数え上げられるわけではない. すなわち, $\mathbf{s}_t$ は図 7 にみるように $[0, 1)$ のある区間に対応するが, こうした区間は無限個の数があり, これを全て数え上げてその中から (32) 式の確率で選ぶことは不可能である.

そこで, 3.3 節と同様に Retrospective sampling と Slice sampling を組み合わせることで学習を行う. $\mathbf{s}_t$ は $[0, 1)$ の区間にあるから, 乱数 $r \sim \mathrm{Unif}[0, 1)$ をサンプルして対応するノードを求めれば, TSSB からランダムにノードをサンプルすることができる. ここで (32) 式に従ってランダムにサンプリングすることは, まず $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ からランダムに $\mathbf{s}_t$ を選び, そこから重み $p(w_t|\mathbf{s}_t)p(\mathbf{s}_{t+1}|\mathbf{s}_t)$ に従って選ぶことと同じであるから, これは 3.3 節の無限階層クラスタリングの学習において「尤度」が出力確率 $p(w_t|\mathbf{s}_t)$ だけでなく, 次の状態への遷移確率との積 $p(w_t|\mathbf{s}_t)p(\mathbf{s}_{t+1}|\mathbf{s}_t)$ となっている場合とみなすことができる. したがって, 同様にして

[*3] 実際には基底測度から出た数を管理するため, 後者が出た場合に新しいテーブルを用意し, 現在のテーブル数+1 の多項分布からサンプリングを行う.

```
function draw_state (s_{t-1}, s_t, s_{t+1}, w_t)
slice = p(w_t|s_t) p(s_{t+1}|s_t) · Unif[0, 1)
st := 0; ed := 1
while true do
    u := Unif[st, ed)
    s := s_{t-1} → TSSB → find_node(u)
    p := p(w_t|s) p(s_{t+1}|s)
    if p > slice then
        return s
    else
        if s < s_t then
            st := u
        else
            ed := u
        end if
    end if
end while
```

図 12　スライスサンプリングによる iTHMM の状態 $\mathbf{s}_t$ のサンプリング. $\mathbf{u} < \mathbf{s}$ は状態 $\mathbf{u}$ が辞書順で状態 $\mathbf{s}$ より前にあることを表す. TSSB→find_node(u) は $[0, 1)$ の実数 u に対応する TSSB のノードを返す関数であり, [12] を参照のこと.

(1) 現在の $\mathbf{s}_t$ について, スライス $\rho = p(w_t|\mathbf{s}_t)p(\mathbf{s}_{t+1}|\mathbf{s}_t) \cdot \mathrm{Unif}[0, 1)$ を作る. $st = 0, ed = 1$.

(2) $r \sim \mathrm{Unif}[st, ed)$ をサンプルし, $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ の TSSB からこれに対応するノード $\mathbf{s}'_t$ を求める.

(3) $p(w_t|\mathbf{s}'_t)p(\mathbf{s}_{t+1}|\mathbf{s}'_t) > \rho$ ならば受理. そうでなければ, st, ed を適切に変更して (2) に戻る.

というスライスサンプリングで $\mathbf{s}_t$ をサンプルすることができる. このアルゴリズムを図 12 に示した.

階層的出力確率　ここまでの議論ではノード $\mathbf{s}$ における単語の出力確率分布 $p(\cdot|\mathbf{s})$ については複雑さを避けるために特にふれなかったが, $\mathbf{s}$ は木構造をなしているから, 親子関係にある $p(\cdot|\mathbf{s}')$ と $p(\cdot|\mathbf{s})$ には依存関係があるのが自然である. 一般には [12] で述べられているように, ノード $\mathbf{s}$ における出力分布は例えばガウス分布であれば, 親のガウス分布の平均 $\mu_{\mathbf{s}'}$ を期待値とする拡散過程 $N(\mu_{\mathbf{s}'}, \sigma^2)$ などを考えればよい. いま, 我々の観測値は離散的な単語であるから, 本研究では $G_{\mathbf{s}} = \{p(\cdot|\mathbf{s})\}$ は階層 Pitman-Yor 過程 [26]

$$G_{\mathbf{s}} \sim \mathrm{HPY}(G_{\mathbf{s}'}, d_{|\mathbf{s}|}, \theta_{|\mathbf{s}|}) \tag{33}$$

を用いた.[*4] これにより, 下位のノードほど出力分布の尖った特別なカテゴリが学習されることになる.

EOS の取り扱い　実際の解析では, 文頭および文末に特別な状態 EOS を置くことで, 先頭または末尾であるという情報を表現することが多い. 状態が独立である通常の HMM では状態 0 を EOS とし, 状態 1 から先を学習すべき状態とすればよいが, 我々の iTHMM においてはノード [] はすべての状態遷移確率および出力確率の事前分布を表す特別なノードであり, EOS として用いることはできない.

このため, 本研究では EOS とそこからの状態遷移確率を表す単独の TSSB を用意することにした. このとき, 各状態 $\mathbf{s}$ について EOS を含む遷移確率の総和を 1 にするため, $\mathbf{s}$ ごとに EOS への遷移確率 $q_{\mathbf{s}} = p(\mathrm{EOS}|\mathbf{s})$ を別に計算する. $q_{\mathbf{s}}$ がベータ事前分布

$$q_{\mathbf{s}} \sim \mathrm{Be}(\tau_0, \tau_1) \tag{35}$$

に従うとすると, $\mathbf{s}$ から EOS へ遷移した回数を $c_0(\mathbf{s})$, それ以外の状態へ遷移した回数を $c_1(\mathbf{s})$ とすれば, $q_{\mathbf{s}}$ の事後確率は

$$E[q_{\mathbf{s}}|c_0(\mathbf{s}), c_1(\mathbf{s})] = \frac{\tau_0 + c_0(\mathbf{s})}{\tau_0 + \tau_1 + c_0(\mathbf{s}) + c_1(\mathbf{s})} \tag{36}$$

となる. 本研究では, $(\tau_0, \tau_1) = (1, 100)$ とした. 残った $(1 - q_{\mathbf{s}})$ の確率を TSSB によって分配し, 通常の状態への遷移確率として用いる. これは, 一種のディリクレツリー分布 [27] とみることができる.

以上をまとめると, iTHMM の学習アルゴリズムは図 13 のようになる. 上の (32) 式では 2 つの状態遷移確率 $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ と $p(\mathbf{s}_{t+1}|\mathbf{s}_t)$ を独立に計算しているが, 厳密には生成モデルに従えばこの 2 つの確率には依存関係があり, $\mathbf{s}_{t+1} = \mathbf{s}_t$ だった場合に $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ で 1 増えた頻度が $p(\mathbf{s}_{t+1}|\mathbf{s})$ に影響を与えるため, Metropolis-Hastings 法により補正する必要がある [28]. ただし, HCDP での確率の変化はきわめて複雑であり, 単純にカウントの ± 1 で MH に必要な正しい確率を求めることはできない. 本研究では実際に $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ の客を HCDP に追加してから $p(\mathbf{s}_{t+1}|\mathbf{s}_t)$ を計算し, 客を再び削除するという方法で正しい確率を計算することにした. 実験では, この補正による MH の受理確率は 99.99%以上であった.

5. 関連研究

「階層的」な HMM としては階層型 HMM (Hierarchical HMM, HHMM) [29] が知られており, その無限化も提案されている [30]. しかし, これは通常の HMM の潜在状態を上位の HMM の出力とみることで, 抽象度を上げて水平方

```
for iter = 1 ··· iters do
    for n in randperm(1 ··· N) do
        remove (w_t, s_{t-1}, s_t, s_{t+1})
        Draw s'_t = draw_state(w_t, s_{t-1}, s_t, s_{t+1})
        if MH-accept(s'_t, s_t) then
            s_t = s'_t
        end if
        add (w_t, s_{t-1}, s_t, s_{t+1})
    end for
end for
function add (w_t, s_{t-1}, s_t, s_{t+1})
    s_t →add_customer (w_t)
    s_{t-1} →add_customer(s_t)
    s_t →add_customer(s_{t+1})
function remove (w_t, s_{t-1}, s_t, s_{t+1})
    s_t →remove_customer (w_t)
    s_t →remove_customer(s_{t+1})
    s_{t-1} →remove_customer(s_t)
```

図 13　iTHMM の Gibbs サンプリングによる学習アルゴリズム.

[*4] 上の拡散過程において, 基底測度に κ の割合でノイズを加えた

$$G_{\mathbf{s}} \sim \mathrm{HPY}(\kappa G_0 + (1 - \kappa)G_{\mathbf{s}'}, d_{|\mathbf{s}|}, \theta_{|\mathbf{s}|}) \tag{34}$$

とした方がノードのもつ出力確率分布のバラエティが増える可能性があるが [12], ハイパーパラメータ d, θ の学習が困難になるため, 本研究では採用しなかった. G_0 は一様分布 $1/V$ などにとる.

表 2 『不思議の国のアリス』での予測精度. “iHMM” は提案法で木の最大の深さ M を 1 に制限したものである. iTHMM における λ の設定では, 木の最大の深さは ∞ である.

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

向 (時間軸) に隠れ状態を粗視化するものであり, 提案法のように時間軸の解像度を保持したまま, HMM で得られる隠れ状態自体を垂直方向に微視化するものとは異なる. また, 通常の HMM の延長である HHMM とは違い, これには本論文で述べたような新しいモデル化を必要とする.

この意味で, 提案法はむしろ Jordan らの隠れ Markov 決定木 (HMDT) [31] に似ている. ただし, HMDT や HHMM と異なり, 提案法は階層の深さが固定ではなく木の場所によって可変長であることや, HMDT と違い状態遷移が単純に深さ別にあるのではなく, 無限木構造上で自然に定義されること, また全体が統一された統計モデルとなっており, MCMC 法で近似なしに解かれるという特徴がある.

6. 実験

英語と日本語の標準的なコーパスで実験を行った. 実装は C++で 7000 行程度である. 学習するモデルの複雑さにもよるが, 現在の実装では Xeon 3.7GHz で 1 秒あたり数千語の隠れ状態をサンプリングすることができる.

6.1 教師なし学習とその性能

『不思議の国のアリス』のテキストで実験を行った. 最初の 1200 文を学習データ, 残りの 231 文をテストデータとした. 提案法の隠れ状態は木構造をなしているが, 確率は独立に計算できるため, 事前にすべてのノードの間の遷移確率を計算しておくことで, 前向き計算と Viterbi デコーディングは効率的に行うことができる.

表 2 にテストデータでの予測精度 (パープレキシティ) を示す. 木の高さが常に 1 の通常の HMM と比べて, 構造化された状態が適切にスムージングされる iTHMM は高い性能を見せることがわかる. 木の深さを無限まで取ると予測精度が落ちるが, これはデータ量が少ないせいもあると考えられ, 理由を探ってゆきたい. このとき学習された状態の一部を表 3 に示す. 木の根では確率分布がフラットになっており, 動詞や名詞を表すと思われる状態 2 や状態 4 の中で, さらに自動的に細分化が起きていることがわかる.

6.2 半教師あり学習

提案手法は教師あり学習だけでなく, 半教師あり学習も行うことができる. これには図 13 のアルゴリズムにおいて, 9 行目の後に教師データをモデルに加え, そのデータは更新しないようにすればよい. これにより, 既存の品詞体系と整合性を持ちつつ, 必要に応じて詳細化された品詞が得られると期待できる.

[]			[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
[2 3]			**[2 7]**		
know	69	0.1976	be	80	0.2478
think	41	0.1172	have	47	0.1451
say	20	0.0568	go	14	0.0397
wish	18	0.0489	remember	11	0.0322
wonder	16	0.0431	do	11	0.0296
tell	16	0.0453	get	11	0.0328
see	14	0.0343	take	10	0.0300
do	12	0.0357	talk	9	0.0266
[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

表 3 『不思議の国のアリス』で学習された状態と単語出力確率の例. 2 番目の数字はその単語が状態に割り当てられた回数を表している. 表 1 と比べて単語がよりクラスタ化されており, 動詞が [2] の下位カテゴリにそれぞれまとまるなど, 興味深い動作がみられる.

表 6 に, 京大コーパスにおいて 10000 文の品詞を教師データとした上で, 37400 文の教師なしデータを学習した半教師あり学習の結果の一部を示す. 教師データの品詞と隠れ状態の対応は, 品詞の頻度順に表 4 のようにした. ここでは細分類は与えていないが, 表 6 にみられるように, iTHMM が適切に細分類および新しい状態を学習していることがわかる.

6.3 未知の言語

最後に, 提案法はそもそも品詞体系が知られていない未知の言語に対して特に有益であろうと考えられる. 表 5 に, クリンゴン語で書かれた「ハムレット」[*5] を解析した様子

表 4 京大コーパスの半教師あり学習での潜在状態と品詞の対応.

状態	品詞		
0	名詞	8	判定詞
1	助詞	9	接頭辞
2	特殊	10	助動詞
3	動詞	11	接続詞
4	接尾辞	12	連体詞
5	形容詞	13	感動詞
6	副詞	14	未定義語
7	指示詞		

[*5] https://en.wikipedia.org/wiki/The_Klingon_Hamlet

表 5　クリンゴン語「ハムレット」の解析結果の一部.

[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
[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

を示す. これは 3,733 行, 19,927 語の小さなテキストである. クリンゴン語辞書によれば, DaH(=now), vaj(=then) といった言葉は間投詞を, ’el(=go), mej(=leave) といった言葉は動詞を表しており, これは統計モデルの結果とほぼ符合しているといえる.

7. まとめと展望

本論文では, ディリクレ過程を階層ディリクレ過程に拡張したのと同様に, ディリクレ過程の積である木構造 Stick-breaking 過程 [12] を階層化した階層的木構造 Stick-breaking 過程とその学習法を示し, 各状態がもつ出力分布を拡散過程として階層 Pitman-Yor 過程にとることで, 通常の HMM と異なり, 状態の隠れた階層構造も学習できる無限木構造隠れ Markov モデル (iTHMM) を提案した. 提案法は, TSSB による階層クラスタリングを時系列上で行うものととらえることができる.

学習には局所的な Gibbs サンプラーを用いたが, これは無限 HMM の Beam サンプラーと異なり, 無限個の状態を容易にスライスで有限化し, 前向き–後向き計算を行うことができないためである. しかし, 提案法は状態がすべて $[0, 1)$ の間の実数の区間で表されるという特徴があり, これを利用して状態をランダムに離散化してから前向き–後向き計算を行う Neal の Embedded HMM [32] が適用できる可能性がある. そうした効率的な学習法についても考慮していきたい.

提案法は隠れ状態を階層的にとらえるための最初のステップであり, ハイパーパラメータの学習や MCMC を用いても残る局所解の問題など, 課題は多く残されている. 自然言語処理内外の適用を含め, モデルの可能性をさらに探っていきたい.

参考文献

[1] Rabiner, L. R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257–286 (1989).
[2] Bishop, C. M.: *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer (2007).
[3] Merialdo, B.: Tagging English Text with a Probabilistic Model, *Computational linguistics*, Vol. 20, No. 2, pp. 155–171 (1994).
[4] Kupiec, J.: Robust part-of-speech tagging using a hidden Markov model., *Computer Speech & Language*, Vol. 6, No. 3, pp. 225–242 (1992).
[5] Goldwater, S. and Griffiths, T.: A Fully Bayesian Approach to Unsupervised Part-of-Speech Tagging, *Proceedings of ACL 2007*, pp. 744–751 (2007).
[6] Beal, M. J., Ghahramani, Z. and Rasmussen, C. E.: The Infinite Hidden Markov Model, *NIPS 2001*, pp. 577–585 (2001).
[7] Van Gael, J., Saatci, Y., Teh, Y. W. and Ghahramani, Z.: Beam sampling for the infinite hidden Markov model, *ICML 2008*, pp. 1088–1095 (2008).
[8] Suzuki, J. and Isozaki, H.: Semi-Supervised Sequential Labeling and Segmentation Using Giga-Word Scale Unlabeled Data, *ACL:HLT 2008*, pp. 665–673 (2008).
[9] Christodoulopoulos, C., Goldwater, S. and Steedman, M.: Two decades of unsupervised POS induction: How far have we come?, *EMNLP 2010*, pp. 575–584 (2010).
[10] Matsuzaki, T., Miyao, Y. and Tsujii, J.: Probabilistic CFG with latent annotations, *ACL 2005*, pp. 75–82 (2005).
[11] Shindo, H., Miyao, Y., Fujino, A. and Nagata, M.: Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing, *ACL 2012*, pp. 440–448 (2012).
[12] Adams, R. P., Ghahramani, Z. and Jordan, M. I.: Tree-Structured Stick Breaking for Hierarchical Data, *NIPS 2010*, pp. 19–27 (2010).
[13] Teh, Y. W., Jordan, M. I., Beal, M. J. and Blei, D. M.: Hierarchical Dirichlet Processes, *JASA*, Vol. 101, No. 476, pp. 1566–1581 (2006).
[14] Blackwell, D. and MacQueen, J. B.: Ferguson Distributions via Pólya Urn Schemes, *Annals of Statistics*, Vol. 1, No. 2, pp. 353–355 (1973).
[15] Sethuraman, J.: A Constructive Definition of Dirichlet Priors, *Statistica Sinica*, Vol. 4, pp. 639–650 (1994).
[16] Paisley, J. and Carin, L.: Hidden Markov models with stick-breaking priors, *IEEE Transactions on Signal Processing*, Vol. 57, pp. 3905–3917 (2009).
[17] Neal, R. M.: Slice sampling, *Annals of Statistics*, pp. 705–741 (2003).
[18] Hjort, N. L., Holmes, C., Müller, P. and Walker, S. G.: *Bayesian Nonparametrics*, Cambridge University Press (2010).
[19] Mochihashi, D. and Sumita, E.: The Infinite Markov Model, *Advances in Neural Information Processing Systems 20 (NIPS 2007)*, pp. 1017–1024 (2008).
[20] Mauldin, R. D., Sudderth, W. D. and Williams, S. C.: Polya Trees and Random Distributions, *Annals of Statistics*, Vol. 20, No. 3, pp. 1203–1221 (1992).
[21] Blei, D. M., Griffiths, T. L. and Jordan, M. I.: The Nested Chinese Restaurant Process and Bayesian Nonparametric Inference of Topic Hierarchies, *JACM*, Vol. 57, No. 2, pp. 1–30 (2010).
[22] Ahmed, A., Hong, L. and Smola, A.: Nested Chinese Restaurant Franchise Processes: Applications to User Tracking and Document Modeling, *ICML 2013*, pp. 1426–1434 (2013).
[23] Paisley, J., Wang, C., Blei, D. and Jordan, M. I.: Nested hierarchical Dirichlet processes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 37, No. 2, pp. 256–270 (2015).
[24] Papaspiliopoulos, O. and Roberts, G. O.: Retrospective

Markov chain Monte Carlo methods for Dirichlet process hierarchical models, *Biometrika*, Vol. 95, No. 1, pp. 169–186 (2008).

[25] Noji, H., Mochihashi, D. and Miyao, Y.: Hierarchical Tree-Structured Stick-Breaking Priors, *NIPS 2013 workshop: Modern Nonparametric Methods in Machine Learning* (2013).

[26] Teh, Y. W.: A Bayesian Interpretation of Interpolated Kneser-Ney, Technical Report TRA2/06, School of Computing, NUS (2006).

[27] Minka, T.: The Dirichlet-tree distribution (1999). http://research.microsoft.com/˜minka/papers/dirichlet/minka-dirtree.pdf.

[28] Johnson, M., Griffiths, T. L. and Goldwater, S.: Bayesian Inference for PCFGs via Markov Chain Monte Carlo, *Proceedings of HLT/NAACL 2007*, pp. 139–146 (2007).

[29] Fine, S., Singer, Y. and Tishby, N.: The Hierarchical Hidden Markov Model: Analysis and Applications, *Machine Learning*, Vol. 32, No. 1, pp. 41–62 (1998).

[30] Heller, K. A., Teh, Y. W. and Görür, D.: Infinite Hierarchical Hidden Markov Models, *AISTATS 2009*, pp. 224–231 (2009).

[31] Jordan, M. I., Ghahramani, Z. and Saul, L. K.: Hidden Markov decision trees, *Advances in Neural Information Processing Systems (1997)*, pp. 501–507 (1997).

[32] Neal, R. M., Beal, M. J. and Roweis, S. T.: Inferring state sequences for non-linear systems with embedded hidden Markov models, *Advances in Neural Information Processing Systems 16 (2004)*, pp. 401–408 (2004).

表 6　京大コーパスの半教師あり学習で導出された隠れ状態.

[ ]		
OOV	4702	0.0639
関係	74	0.0010
首相	50	0.0004
何	49	0.0004
代表	48	0.0004
建設	48	0.0004
推進	47	0.0004
支持	46	0.0004

[0]		
OOV	796	0.0581
日本	124	0.0082
それ	87	0.0056
選挙	66	0.0044
この	66	0.0042
外	52	0.0033
関係	52	0.0033
する	51	0.0034

[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

[0 1]		
OOV	658	0.1225
中	173	0.0322
こと	104	0.0194
問題	91	0.0170
声	67	0.0124
ため	66	0.0122
人	62	0.0114
責任	51	0.0095

[0 1 2]		
方	81	0.4144
者	36	0.1838
問題	30	0.1533
性	21	0.1070
OOV	9	0.0469
例	7	0.0353
規定	7	0.0353
データ	2	0.0096

[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

[5]		
OOV	385	0.1192
同	61	0.0186
大阪	56	0.0165
両	40	0.0124
東京	40	0.0118
関根	31	0.0090
神戸	30	0.0093
各	23	0.0066

[5 0]		
「	513	0.2102
その	262	0.1073
この	217	0.0888
OOV	158	0.0675
まだ	47	0.0193
同じ	37	0.0150
さらに	36	0.0146
こうした	32	0.0129

[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

[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