

# HHMM変換を用いた左非循環PCFGの高速推論

若林 啓<sup>1,a)</sup>

受付日 2014年9月20日, 採録日 2014年12月15日

**概要:** 確率的文脈自由文法 (PCFG) は, 系列データの統語構造を推定する機械学習モデルであり, 自然言語処理をはじめとしてプラン認識やアクセスログ解析など広範な分野で利用されている. しかし, PCFGの代表的な推論アルゴリズムである Inside-outside algorithm は, 計算量が系列長に対して3乗のオーダーであり, 長い系列を含むデータセットの解析は現実的でないという問題がある. 本研究では, PCFGを定義する文法が左非循環文法 (Left Acyclic Grammar; LAG) と呼ぶ制約を満たすことを条件として, PCFGを階層型隠れマルコフモデル (Hierarchical Hidden Markov Model; HHMM) に等価変換することにより, 線形時間で等価な PCFG の推論を行う手法を提案する. 実験により, 提案手法によって Inside-outside algorithm と等価な事後確率分布の推論が可能であることを確認し, 特に長い系列を含むデータセットにおいて大幅な高速化が可能になることを示す.

**キーワード:** 確率文脈自由文法, 階層型隠れマルコフモデル, 左非循環文法

## Efficient Inference of Left Acyclic PCFG Using HHMM Transformation

KEI WAKABAYASHI<sup>1,a)</sup>

Received: September 20, 2014, Accepted: December 15, 2014

**Abstract:** Probabilistic Context Free Grammar (PCFG) is a machine learning model for estimating latent syntactic structures of sequence data, which has a wide application area such as natural language processing, plan recognition and log analysis. However, the standard inference method for PCFGs, inside-outside algorithm, has cubic-time complexity for input sequence that makes it impractical to apply to long sequence data. In this study, we propose a linear-time inference method for PCFGs by using equivalent transformation into Hierarchical Hidden Markov Model (HHMM) under a condition of grammar named Left Acyclic Grammar (LAG). We give the experimental results that demonstrate our proposal method estimates exactly identical posterior distribution with inside-outside algorithm does, and show the execution time is dramatically improved especially for long sequence data.

**Keywords:** probabilistic context free grammar, hierarchical hidden Markov model, left acyclic grammar

### 1. はじめに

確率的文脈自由文法 (Probabilistic Context Free Grammar; PCFG) は, 系列データの潜在構造を構文木として推定する確率モデルであり, 自然言語文章の構文解析をはじめとして形態素解析 [8] やプラン認識 [5], [14], アクセスログ解析 [20] などに応用されている. また, 近年では, パラメータの事前分布を考慮した PCFG のベイズ推定手法 [8], [19]

や, ノンパラメトリックベイズ推定手法 [11], Pitman-Yor 過程による部分木のキャッシュを考慮した PCFG の拡張モデル [7] などの提案によって, PCFG の教師なし学習手法に新たな展開が期待されている.

PCFG は, 構文木  $t$  を潜在変数として含んだ, シンボル系列  $w$  の生成モデルである. 与えられた  $w$  について, 潜在変数の事後周辺確率分布<sup>\*1</sup>を求めることを, PCFG のモデル推論と呼ぶ. モデル推論を行うことで, EM アルゴリ

<sup>1</sup> 筑波大学図書館情報メディア系  
Faculty of Library, Information and Media Science, University of Tsukuba, Tsukuba, Ibaraki 305-8550, Japan

<sup>a)</sup> kwakaba@slis.tsukuba.ac.jp

<sup>\*1</sup> PCFG は生成モデルであるため, 構文木  $t$  の確率  $p(t)$  を陽に求めることができる. 与えられた  $w$  について, 構文木の事後確率分布とは条件付き確率  $p(t|w)$  を指し, その事前確率分布は  $p(t)$  に対応する.

ズムなどのモデルパラメータ推定や構文木のサンプリング、MCMC法の適用などを実現できることから、モデル推論はPCFGを機械学習モデルとして利用するために重要な役割を果たす。Inside-outsideアルゴリズムは、動的計画法に基づいたPCFGの効率的なモデル推論手法として知られている[9]。しかし、Inside-outsideアルゴリズムの計算量はシンボル系列の長さ $T$ に対して $O(T^3)$ であり、系列が長くなると計算時間が急激に増加する。プラン認識やアクセスログ解析、音素列解析[7]などのPCFGの応用は一般に長い系列を解析対象に含むが、この計算量の問題のため、現実的には短い系列に制限する必要がある。

一般に、PCFGの事後確率推論の計算量は、準3乗オーダーよりも小さくすることが困難であると指摘されている[10]。しかし、文法に制約を加えることによって、線形時間でモデル推論が行える場合がある。たとえば、Right-linear grammar (RLG)に制約されたPCFGは、生成する木構造がつねに線形鎖と見なせるため、隠れマルコフモデル (Hidden Markov Model; HMM)と同様の動的計画法による推論アルゴリズムを用いることで、系列長に対して線形時間で事後確率推論が実現できることが知られている[13]。しかし、応用上の観点からはRLGは強すぎる制約といえる。

本研究では、階層型隠れマルコフモデル (Hierarchical Hidden Markov Model; HHMM)への変換を利用して、PCFGの事後確率分布の推論を線形時間で行う手法を提案する。HHMMは、HMMの状態が別のHMMを階層的に生成する過程を表現した確率モデルであり、様々な粒度の系列パターンを扱うことのできる特性から、プラン認識[1]や固有表現抽出[17]、自然言語文章の構文解析[16]といった問題に応用されている。HHMMは線形時間の事後確率分布の推論アルゴリズムが知られており[18]、PCFGよりも高速に推論を実行できるというメリットがある。しかし、HHMMでは有限の階層が仮定されることから、文法の制約のないPCFGをHHMMに等価変換することはできない。

本研究の貢献は、HHMMに等価変換可能なPCFGの文法の十分条件を提案し、この条件の下でInside-outsideアルゴリズムと等価な結果を保証するPCFGの線形時間推論手法を明らかにすることである。本稿では、左非循環文法 (Left Acyclic Grammar; LAG)と呼ぶ文法のクラスを定義し、これがPCFGのモデル推論を線形時間で実行できることの十分条件になっていることを示す。

本稿の構成は以下のとおりである。2章でPCFGと既存の推論手法を紹介し、3章でHHMMの定義と推論アルゴリズムについて述べる。4章では、本研究で提案する左非循環文法を定義し、HHMMへの変換アルゴリズムおよび等価推論手法について論じる。5章で実験結果を示し、6章で結論とする。

## 2. 確率文脈自由文法

文脈自由文法 (Context Free Grammar; CFG)は、4つ組 $(N, \Sigma, S, R)$ によって定義される。ここで、 $N$ は非終端記号の集合、 $\Sigma$ は終端記号の集合、 $S \in N$ は開始記号、 $R$ は生成規則の集合である。生成規則 $r \in R$ は、 $r = X \rightarrow \gamma$  (ただし、 $X \in N$ 、 $\gamma \in (N \cup \Sigma)^*$ )の形式で表され、 $X$ を $r$ の左辺、 $\gamma$ を右辺と呼ぶ。 $X$ を左辺を持つ生成規則の集合を $R_X \subseteq R$ で表す。確率文脈自由文法 (PCFG)は、5つ組 $(N, \Sigma, S, R, \theta)$ で定義される。 $\theta$ は長さ $|R|$ のベクトルであり、 $\theta_r$ は生成規則 $r \in R$ の適用確率を表す。すべての $X \in N$ について、 $\sum_{r \in R_X} \theta_r = 1$ を満たす。PCFG $G$ による構文木 $t$ の生成確率は以下のように定義される。

$$P_G(t|\theta) = \prod_{r \in R} \theta_r^{f_r(t)}$$

$f_r(t)$ は構文木 $t$ における生成規則 $r$ の適用回数である。 $t$ の産出 (yield)  $y(t)$ を、生成された構文木の葉に対応する終端記号の系列とする。PCFGにおける終端記号系列 $o_{1:T}$ の生成確率は以下のように与えられる。

$$P_G(o_{1:T}|\theta) = \sum_{t: y(t)=o_{1:T}} P_G(t|\theta)$$

### 2.1 チョムスキー標準文法

生成規則の形式に制約がない場合、PCFGの推論は容易ではない。チョムスキー標準形 (Chomsky Normal Form; CNF) またはチョムスキー標準文法 (Chomsky Normal Grammar; CNG)は、 $R$ に含まれるすべての生成規則が以下のいずれかの形式に従うCFGである。

- (1)  $S \rightarrow X$
- (2)  $X \rightarrow YZ$
- (3)  $X \rightarrow v$

ただし、 $S$ は開始記号、 $X, Y, Z \in N$ 、 $v \in \Sigma$ とする。任意のCFGはCNGに等価に変換できることが知られており、CNGの仮定により一般性は失われない。

ここでは、CNGに従うPCFGを確率チョムスキー標準文法 (Probabilistic CNG; PCNG)と呼ぶ。Inside-outsideアルゴリズムは、PCNGについて推論を行うアルゴリズムである[9]。Inside-outsideアルゴリズムは、終端記号系列 $o_{1:T}$ のすべての部分系列についての動的計画法であり、系列長 $T$ 、生成規則の数 $|R|$ に対して $O(T^3|R|)$ の計算量である。このため、 $T$ が大きくなると急速に計算時間が増加する。

### 2.2 右線形文法

右線形文法 (Right Linear Grammar; RLG)は、 $R$ に含まれるすべての生成規則が以下のいずれかの形式に従うCFGである。

- (1)  $S \rightarrow X$

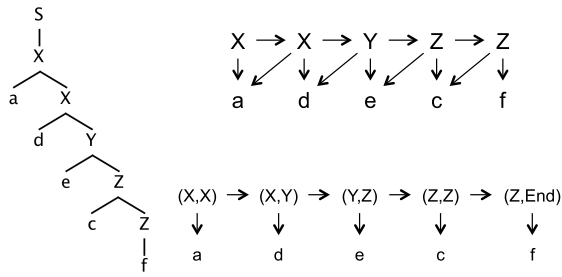


図 1 (左) PRLG によって生成される構文木. (右上) 変数間の確率依存関係. (右下) 等価な HMM 表現

Fig. 1 (left) Parse tree of PRLG. (top-right) Probabilistic dependency diagram in graphical model fashion. (bottom-right) Equivalent HMM representation.

(2)  $X \rightarrow v Y$

(3)  $X \rightarrow v$

ただし,  $S$  は開始記号,  $X, Y \in N$ ,  $v \in \Sigma$  とする. RLG が受理する構文木は, 線形鎖を成す木に限られる (図 1 (左)).

RLG に従う PCFG を, 確率右線形文法 (Probabilistic RLG; PRLG) と呼ぶ. Ponvert ら [13] は, PRLG の推論を  $O(T)$  で行う手法を提案している. ここでは, PRLG を系列終了確率付きの隠れマルコフモデル (HMM) に変換する. HMM は 5 つ組  $(\Pi, \Sigma, A, B, \pi)$  で定義される.  $\Pi$  は状態の有限集合,  $\Sigma$  は終端記号の集合,  $A_i(j)$  は状態  $i$  から  $j$  への遷移確率,  $B_i(v)$  は状態  $i$  からの終端記号  $v$  の出力確率,  $\pi(i)$  は状態  $i$  の初期状態確率である.

PRLG が生成する構文木は線形鎖をなすため, 構文木に出現するそれぞれの非終端記号に対して, 開始記号からの順番を数えることができる. 開始記号の次の非終端記号から数えて,  $t$  番目に生成される非終端記号を表す確率変数を  $Q_t$  とし,  $Q_t$  が生成する終端記号を  $O_t$  とする. この定義より,  $\theta_{X \rightarrow v Y} = P(O_t = v, Z_{t+1} = Y | Q_t = X)$  がいえる. この確率を以下のように分解する.

$$P(O_t = v, Q_{t+1} = Y | Q_t = X) = P(Q_{t+1} = Y | Q_t = X) P(O_t = v | Q_t = X, Q_{t+1} = Y)$$

この分解は, 構文木に現れる記号を確率変数と見なしたときの確率依存関係が, 図 1 (右上) で表現できることを意味している.

$Q_t$  と  $Q_{t+1}$  のペア  $(Q_t, Q_{t+1})$  を新たに状態として定義することで, HMM 表現が得られる. HMM の状態集合  $\Pi$  を以下のように定義する.

$$\Pi = \{(X, Y) : \forall X \rightarrow v Y \in R\} \cup \{(X, End) : \forall X \rightarrow v \in R\}$$

ただし,  $End$  は PRLG の非終端記号には存在しない記号とする. 終端記号出力確率  $B$ , 初期状態確率  $\pi$ , 状態遷移確率  $A$  は以下のように定義する.

$$B_{(X,Y)}(v) = \frac{\theta_{X \rightarrow v Y}}{\sum_{v' \in \Sigma} \theta_{X \rightarrow v' Y}}$$

$$B_{(X,End)}(v) = \frac{\theta_{X \rightarrow v}}{\sum_{v' \in \Sigma} \theta_{X \rightarrow v'}}$$

$$\pi((X, Y)) = \theta_{S \rightarrow X} \sum_{v \in \Sigma} \theta_{X \rightarrow v Y}$$

$$\pi((X, End)) = \theta_{S \rightarrow X} \sum_{v \in \Sigma} \theta_{X \rightarrow v}$$

$$A_{(X,Y)}((Z, W)) = \delta(Z = Y) \sum_{v \in \Sigma} \theta_{Z \rightarrow v W}$$

$$A_{(X,Y)}((Z, End)) = \delta(Z = Y) \sum_{v \in \Sigma} \theta_{Z \rightarrow v}$$

ここで,  $\delta(Z = Y)$  は  $Z = Y$  のとき 1, それ以外のとき 0 になるディラックのデルタ関数である. この変換によって得られる HMM が元の PRLG と等価な確率分布を与えることは, 以下のように示せる.  $X \rightarrow v Y$  の形式の生成規則を  $T-1$  回行う構文木を生成する確率は, 元の PRLG  $G$  では以下になる.

$$P_G(Q_{1:T}, O_{1:T}) = \theta_{S \rightarrow Q_1} \times \left( \prod_{t=1}^{T-1} \theta_{Q_t \rightarrow O_t Q_{t+1}} \right) \times \theta_{Q_T \rightarrow O_T}$$

変換された HMM での生成確率は以下のようになり, PRLG による生成確率と等価になる. ただし, 紙面の節約のため, 状態  $(Q_t, Q_{t+1})$  を  $Q_{t,t+1}$ , 状態  $(Q_t, End)$  を  $Q_{t,End}$  と表記する.

$$\begin{aligned} & \pi(Q_{1,2}) B_{Q_{1,2}}(O_1) \prod_{t=2}^{T-1} A_{Q_{t-1,t}}(Q_{t,t+1}) B_{Q_{t,t+1}}(O_t) \\ & \times A_{Q_{T-1,T}}(Q_{T,End}) B_{Q_{T,End}}(O_T) \\ & = \theta_{S \rightarrow Q_1} \theta_{Q_1 \rightarrow O_1 Q_2} \prod_{t=2}^{T-1} \theta_{Q_t \rightarrow O_t Q_{t+1}} \times \theta_{Q_T \rightarrow O_T} \\ & = \theta_{S \rightarrow Q_1} \times \left( \prod_{t=1}^{T-1} \theta_{Q_t \rightarrow O_t Q_{t+1}} \right) \times \theta_{Q_T \rightarrow O_T} \\ & = P_G(Q_{1:T}, O_{1:T}) \end{aligned}$$

HMM の推論は前向き後向きアルゴリズム [15] により  $O(T|R|)$  で行うことができる. このため, PRLG の推論は線形時間で実行できる. しかし, RLG は一般の CFG に対して強い制約であり, 応用上の有用性は限定的であるといえる.

### 3. 階層型隠れマルコフモデル

本研究では, HMM よりも表現能力の高い階層型隠れマルコフモデル (Hierarchical Hidden Markov Model; HHMM) を用いて PCFG の線形時間推論を行う手法を提案する. ここでは, HHMM を以下の 7 項組で定義する.

- (1)  $\Pi$  状態の有限集合.
- (2)  $\Sigma$  終端記号の有限集合.
- (3)  $A_i(j)$  状態  $i$  から  $j$  への遷移確率.
- (4)  $\pi_i(j)$  状態  $i$  からの  $j$  の起動確率.
- (5)  $B_i(v)$  状態  $i$  からの終端記号  $v$  の出力確率.
- (6)  $\sigma$  開始状態.

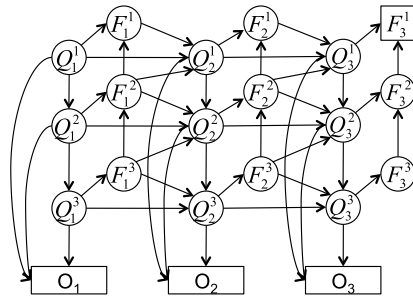


図 2 HHMM のグラフィカルモデル  
Fig. 2 Graphical model of HHMM.

(7)  $I(i) \in \{0, 1\}$  状態  $i$  の生成状態フラグ.

HMM ではすべての状態が終端記号を出力するのに対して, HHMM では別の状態を出力すること (ここでは状態起動と呼ぶ) が認められる\*2. 終端記号出力を行う状態を生成状態, 状態起動を行う状態を内部状態と呼ぶ. 状態  $i$  について, 生成状態フラグ  $I(i)$  はモデル構造として与えられる値であり,  $I(i) = 1$  のとき状態  $i$  は生成状態, 0 のとき内部状態とする.

図 2 に, HHMM のグラフィカルモデルを示す\*3. HHMM は, 時刻  $t$ , 階層  $d$  ごとに隠れ状態  $Q_t^d \in \Pi$  を潜在変数として持つ.  $d = 1$  は最上層の階層を表し,  $d$  が大きいほど下層の階層を表す. また, HHMM は, 終了フラグと呼ぶ 2 値の確率変数  $F_t^d \in \{0, 1\}$  を潜在変数として持つ.  $F_t^d$  が 1 のとき, 階層  $d$  のマルコフ連鎖が時刻  $t$  で終了することを表す. HHMM では, 階層  $d$  の状態は, 階層  $d + 1$  のマルコフ連鎖が終了していない限り, 別の状態に遷移できないように制約される (すなわち,  $Q_t^d = Q_{t-1}^d$  if  $F_{t-1}^{d+1} = 0$ ). マルコフ連鎖の終了は, 特別な擬似状態  $End$  への状態遷移として形式化される.  $Q, F, O$  の確率分布は以下のよう

$$\begin{aligned}
 &P(Q_t^d = j | Q_{t-1}^d = i, F_{t-1}^{d+1} = b, F_{t-1}^d = f, Q_{t-1}^{d-1} = k) \\
 &= \begin{cases} \delta(j = i) & (\text{if } b = 0) \\ A_i(j) & (\text{if } b = 1, f = 0) \\ \pi_k(j) & (\text{if } b = 1, f = 1, I(k) = 0) \\ End & (\text{if } b = 1, f = 1, I(k) = 1) \end{cases} \\
 &P(F_t^d = 1 | Q_t^d = i, F_t^{d+1} = b) \\
 &= \begin{cases} 0 & (\text{if } i \neq End, b = 0) \\ A_i(End) & (\text{if } i \neq End, b = 1) \\ 1 & (\text{if } i = End) \end{cases} \\
 &P(O_t = v | Q_t^d = i) = B_i(v) \quad (d : I(Q_t^d) = 1, I(Q_{t-1}^{d-1}) = 0)
 \end{aligned}$$

\*2 ここでは詳細を省くが, HHMM の確率オートマトンとしての記号系列の生成過程の動作については文献 [4] や [12] を参照されたい.

\*3 文献 [12] や [18] の定義では, 状態  $Q_t^d$  が  $Q_t^1$  から  $Q_t^{d-1}$  までのすべての状態に依存しているが, ここでは  $Q_{t-1}^{d-1}$  のみに依存する定義を示している. このことにより, 階層の最大深さ  $D$  がモデルパラメータに明示的に含まれなくなるが, 後述の APSS の構築により  $D$  は決定的に計算されるため, APSS 構築後の議論にこの変更は影響しない.

ただし, 便宜上  $I(End) = 1$  とする.  $Q_t^d$  の確率分布の定義において  $f = 1$  の場合は, 階層  $d$  のマルコフ連鎖が時刻  $t - 1$  で終了している場合を意味していることから, 時刻  $t$  の状態  $Q_t^d$  が当該階層の新たなマルコフ連鎖の最初の状態として, 状態起動確率  $\pi$  に依存して決まることを示している.

HHMM の効率的な推論手法として, 前向き後向き活性化 (Forward-Backward Activation; FBA) アルゴリズムが知られている [18]. FBA アルゴリズムでは,  $Q^1$  から  $Q^d$  までの状態の可能なすべての組合せを列挙するために, 絶対経路状態空間 (Absolute Path State Space; APSS) を構築する. APSS 上の識別子を用いた系列  $O_{1:T}$  の分解に基づいて, 効率的な推論を行うことができる.

### 3.1 APSS の構築

まず, 状態  $q$  から状態遷移によって到達可能な状態の集合  $Reach_q$  を, 以下のように再帰的に定義する.

- $q \in Reach_q$
- $q' \in Reach_q$  if  $\exists q'' \in Reach_q$  s.t.  $A_{q''}(q') > 0$   
変数  $Q^d$  の値が  $q$  のとき, 直下の階層の変数  $Q^{d+1}$  がとりうる状態の集合  $Under_q$  を, 以下のように定義する.

$$Under_q = \bigcup_{q': \pi_q(q') > 0} Reach_{q'}$$

APSS は, 状態をノードラベルに持つ木構造であり, 根ノードは開始状態  $\sigma$  をラベルに持つ. ラベル  $q$  を持つノードは,  $|Under_q|$  個の子ノードを持ち, それぞれの子ノードが持つラベルが  $Under_q$  の要素に対応する. APSS のそれぞれのノードを絶対経路状態 (absolute path state; APS) と呼ぶ. APS の根からの深さは, HHMM のグラフィカルモデル表現における状態  $Q^d$  の階層に対応している. 階層  $d$  の APS の集合を  $\Omega_d$  とし, 根から最も遠い葉ノードまでの距離を  $D$  とする. APSS の構築手順をアルゴリズム 1 に示す.

図 6 は, 図 5 に対応する HHMM から構築した APSS を示している. APSS は, HHMM がとりうるすべての階層の状態の組合せの最小表現である. つまり, 図 5 の各時刻において, 状態を最上層から最下層までたどったとき, APSS の木構造の中に対応する葉ノードを必ず見つけることができ, かつ APSS の中に存在しない状態の組合せをとることはできない.

### 3.2 FBA アルゴリズム

時刻  $t$  において, 深さ 1 から  $d$  までの状態の組合せが APS  $z \in \Omega_d$  に対応する事象を  $Z_t^d = z$  と記述する. FBA アルゴリズムでは, 図 2 における状態  $Q_t^d$  の代わりに, APS  $Z_t^d$  を識別子として推論を行う. この識別子に合わせて, モデルパラメータについて以下の記法を定義する.

**Algorithm 1** APSS の構築

```

1: 根ノード  $r$  を作成し,  $r.label = \sigma$  とする
2:  $\Omega_0 = \{r\}$ 
3:  $d = 0$ 
4: while  $\Omega_d \neq \phi$  do
5:    $\Omega_{d+1} = \phi$ 
6:   for each  $z \in \Omega_d$  do
7:     for each  $q \in Under_{z.label}$  do
8:        $z$  の子ノードとして  $c$  を作成し,  $c.label = q$  とする
9:        $\Omega_{d+1} = \Omega_{d+1} \cup \{c\}$ 
10:    end for
11:  end for
12:   $d = d + 1$ 
13: end while
14:  $D = d$ 

```

$$A_{zy} = A_{L(z)}(L(y)), A_{zEnd} = A_{L(z)}(End)$$

$$\pi_z = \pi_{L(pa(z))}(L(z)), B_{zv} = B_{L(z)}(v), I_z = I(L(z))$$

ただし,  $L(z)$  はノード  $z$  のラベルを返す関数,  $pa(z)$  はノード  $z$  の親ノードを返す関数とする.

APS の親は一意に決まるため,  $Z_t^d$  が所与のとき,  $d$  より上位のすべての階層  $d'$  について  $Z_t^{d'}$  も所与である. また,  $F_t^{d+1} = 1$  であれば, それより下位の階層のマルコフ連鎖はすべて終了していることが保証される. このことから, APS の事後確率は以下のように分解できる.

$$P(Z_t^d = z, F_t^{d+1} = 1 | O_{1:T}, F_T^1 = 1) = \frac{P(Z_t^d = z, F_t^{d+1} = 1, O_{1:t})P(O_{t+1:T}, F_T^1 = 1 | Z_t^d = z, F_t^{d+1} = 1)}{P(O_{1:T}, F_T^1 = 1)}$$

下位階層のマルコフ連鎖が終了することは, 状態遷移を行うための条件である. このことに基づいて, 同時事象  $\{Z_t^d = z, F_t^{d+1} = 1\}$  を遷移活性化事象  $e_t^d = z$  と定義する. また, 遷移活性化と対になる事象として, 同時事象  $\{Z_t^d = z, F_{t-1}^{d+1} = 1\}$  を出力活性化事象  $b_t^d = z$  と定義する. 上記の分解に基づいて, 前向き活性化確率と後向き活性化確率を以下のように定義する.

$$\alpha_{et}^d(z) = P(e_t^d = z, O_{1:t})$$

$$\beta_{et}^d(z) = P(O_{t+1:T}, F_T^1 = 1 | e_t^d = z)$$

$$\alpha_{bt}^d(z) = P(b_t^d = z, O_{1:t-1})$$

$$\beta_{bt}^d(z) = P(O_{t:T}, F_T^1 = 1 | b_t^d = z)$$

FBA アルゴリズムは,  $\alpha$  と  $\beta$  を以下のように逐次的に求める.

$$\alpha_{b1}^1(z) = \pi_z$$

$$\alpha_{bt}^d(z) = \alpha_{bt}^{d-1}(pa(z))\pi_z + \sum_{y \in sib(z)} \alpha_{et-1}^d(y)A_{zy}$$

$$\alpha_{et}^d(z) = \alpha_{bt}^d(z)B_{zO_t}\delta(I_z = 1) + \sum_{c \in ch(z)} \alpha_{et}^{d+1}(c)A_{cEnd}$$

$$\beta_{eT}^1(z) = A_{zEnd}$$

$$\beta_{et}^d(z) = \beta_{et}^{d-1}(pa(z))A_{zEnd} + \sum_{y \in sib(z)} \beta_{bt+1}^d(y)A_{zy}$$

$$\beta_{bt}^d(z) = \beta_{et}^d(z)B_{zO_t}\delta(I_z = 1) + \sum_{c \in ch(z)} \beta_{bt}^{d+1}(c)\pi_c$$

ただし,  $ch(z)$  は APSS において  $z$  の子ノードの集合を返す関数,  $sib(z) = ch(pa(z))$  は  $z$  の兄弟ノードの集合を返す関数である.

FBA アルゴリズムの計算量は, APSS の構造に依存するが, 系列長  $T$  に対しては線形である. APSS のノードの集合を  $\Omega$ , APS  $z$  から状態遷移可能な APS の数の最大値を  $M = \max_{z \in \Omega} sib(z)$  とすると, FBA アルゴリズムの計算量は  $O(TM|\Omega|)$  である.  $\Omega$  の大きさの上界は  $\sum_{d=1}^D M^d$  であり, 状態数と比較して一般にオーダが大きくなる.

求めた  $\alpha$  と  $\beta$  を用いて, 状態の事後確率は以下のように求めることができる.

$$P(Z_t^d = z, F_t^{d+1} = 1 | O_{1:T}, F_T^1 = 1) \propto \alpha_{et}^d(z)\beta_{et}^d(z)$$

また, 前向き活性化確率と後向き活性化確率に基づいて, 状態遷移, 状態起動, 終端記号出力の期待回数をそれぞれ求めることができ, この期待回数を用いて EM アルゴリズムを構成できる [18].

## 4. 左非循環文法

本章では, 左非循環文法 (Left Acyclic Grammar; LAG) を定義し, LAG に従う PCFG を HHMM に変換することで線形時間で推論を行う手法について述べる.

### 4.1 LAG の定義

$R$  に含まれるすべての生成規則が以下のいずれかの形式に従う CFG を, 準チョムスキー標準文法 (Semi-CNG) と呼ぶ.

- (1)  $S \rightarrow X$
- (2)  $X \rightarrow YZ$
- (3)  $X \rightarrow Y$
- (4)  $X \rightarrow v$

ただし,  $S$  は開始記号,  $X, Y, Z \in N$ ,  $v \in \Sigma$  とする. Semi-CNG  $G$  について, 非終端記号  $X$  の右線形到達可能な非終端記号の集合  $Line_X$  を以下のように再帰的に定義する.

- (1)  $X \in Line_X$
  - (2)  $X' \in Line_X$  if  $\exists X'' \in Line_X$  s.t.  $X'' \rightarrow YX' \in R$
- Semi-CNG  $G$  の左到達可能グラフ  $LR_G = (V_{LR_G}, E_{LR_G})$  を以下の手順で構築されるグラフとして定義する.
- (1) 頂点集合を非終端記号の集合とする ( $V_{LR_G} = N$ ).
  - (2)  $X \rightarrow YZ \in R$  または  $X \rightarrow Y \in R$  ならば,  $E_{LR_G}$  に有向辺  $(X, Y)$  と  $(X, X')$ :  $\forall X' \in Line_Y$  を加える.

**定義 1** CFG  $G$  が Semi-CNG であり, かつ  $LR_G$  が非循環グラフであるとき,  $G$  は左非循環文法 (LAG) という.

**例 1** 表 1 の CFG は Semi-CNG の条件を満たす. この

表 1 左非循環文法の例

Table 1 Example of left acyclic grammar.

$S \rightarrow S'$ (1.0)	$Nom \rightarrow Adj\ Nom$ (0.45)	$VP \rightarrow VO\ S'$ (0.2)
$S' \rightarrow NP$ (0.1)	$Nom \rightarrow bear$ (0.35)	$VP \rightarrow VO$ (0.5)
$S' \rightarrow NP\ VP$ (0.9)	$Nom \rightarrow trout$ (0.2)	$VP \rightarrow V$ (0.3)
$NP \rightarrow Det\ Nom$ (0.7)	$Adj \rightarrow little$ (0.33)	$VO \rightarrow V\ NP$ (1.0)
$NP \rightarrow Nom$ (0.3)	$Adj \rightarrow fine$ (0.44)	$V \rightarrow saw$ (1.0)
$Det \rightarrow the$ (1.0)	$Adj \rightarrow fat$ (0.23)	

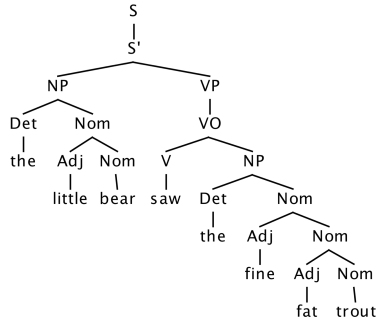


図 3 PLAG の構文木

Fig. 3 Parse tree of PLAG.

文法の左到達可能グラフは以下になる.

$$V_{LRG} = \{S, S', NP, Det, Nom, Adj, VP, VO, V\}$$

$$E_{LRG} = \{(S, S'), (S, VP), (S', NP), (S', Nom), (NP, Det), (NP, Nom), (Nom, Adj), (VP, VO), (VP, NP), (VP, Nom), (VP, V), (VO, V)\}$$

このグラフは有向閉路を持たない. よってこの CFG は LAG である.

$LRG$  が非循環グラフであることを, 文法  $G$  が左非循環性を持つという. 図 3 に, 例 1 の LAG が受理可能な構文木を示す. LAG は RLG とは異なり, 階層構造のある構文木を表現できる. 形態素解析 [8] や音素列解析 [7], プラン認識 [5] などで提案されている文法は, 複雑な再帰性を持たないため, LAG で表現できると考えられる.

#### 4.2 PLAG から HHMM への等価変換

LAG に従う PCFG を, 確率左非循環文法 (Probabilistic LAG; PLAG) と呼ぶ. PLAG に含まれる生成規則の形式は制限されていることから, 構文木に含まれる記号は以下の規則に従って格子状に並べることができる.

- (1)  $S \rightarrow X$  の生成規則によって生成された  $X$  を位置  $(t, d) = (1, 1)$  に置く.
- (2) 位置  $(t, d)$  の非終端記号  $X$  について  $X \rightarrow Y$  の形式の規則が適用される場合,  $Y$  を位置  $(t, d + 1)$  に置く.
- (3) 位置  $(t, d)$  の非終端記号  $X$  について  $X \rightarrow v$  の形式の規則が適用される場合,  $v$  を位置  $(t, d + 1)$  に置く.
- (4) 位置  $(t, d)$  の非終端記号  $X$  について  $X \rightarrow YZ$  の形式

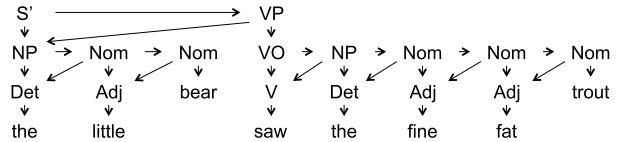


図 4 図 3 の構文木に現れる記号間の確率依存関係

Fig. 4 Probabilistic dependency diagram.

の規則が適用される場合,  $Y$  を位置  $(t, d + 1)$  に置く. この  $Y$  を根とする部分木に含まれる非終端記号の位置  $(t', d')$  のうち, 最も大きい  $t'$  に 1 を加えた値を  $s(t)$  とする.  $Z$  を位置  $(s(t), d)$  に置く.

図 3 の構文木について, 上記の規則に従って記号を配置した図を図 4 に示す. 位置  $d$  に配置される非終端記号と位置  $d + 1$  に配置される非終端記号の間には,  $LRG$  において有向辺が存在する.  $LRG$  は非循環グラフであるため, 可能なすべての構文木における位置  $d$  の最大値は有限になる.

位置  $(t, d)$  に配置された非終端記号を  $Q_{t,d}$  と記述すると,  $\theta_{X \rightarrow YZ} = P(Q_{t,d+1} = Y, Q_{s(t),d} = Z | Q_{t,d} = X)$  がいえる. この確率を以下のように分解する.

$$P(Q_{t,d+1} = Y, Q_{s(t),d} = Z | Q_{t,d} = X) = P(Q_{s(t),d} = Z | Q_{t,d} = X) \times P(Q_{t,d+1} = Y | Q_{t,d} = X, Q_{s(t),d} = Z)$$

この分解は, 構文木に現れる非終端記号を確率変数と見なしたときの確率依存関係が, 図 4 で表現できることを意味している.  $Q_{t,d}$  と  $Q_{s(t),d}$  のペアを新たに状態として定義することで, HHMM 表現が得られる. HHMM の状態集合  $\Pi$  を以下のように定義する.

$$\Pi = \{(X, Z) : \forall X \rightarrow YZ \in R\} \cup \{(X, End) : \forall X \rightarrow Y \in R\} \cup \{(X, End) : \forall X \rightarrow v \in R\}$$

ただし,  $End$  は PLAG の非終端記号には存在しない記号とする. 開始状態を  $\sigma = (S, End)$  とする.

終端記号出力確率  $B$ , 状態起動確率  $\pi$ , 状態遷移確率  $A$  は以下のように定義する.

$$B_{(X, End)}(v) = \frac{\theta_{X \rightarrow v}}{\sum_{v' \in \Sigma} \theta_{X \rightarrow v'}}$$

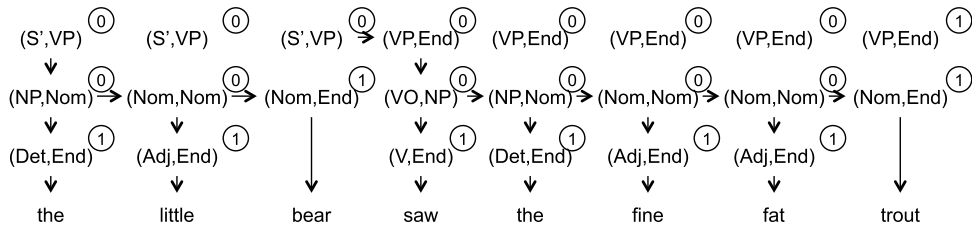


図5 図3の構文木のHHMM表現. 円は  $F_t^d$  の値を示す  
 Fig. 5 Equivalent HHMM representation. The circles denote value of  $F_t^d$ .

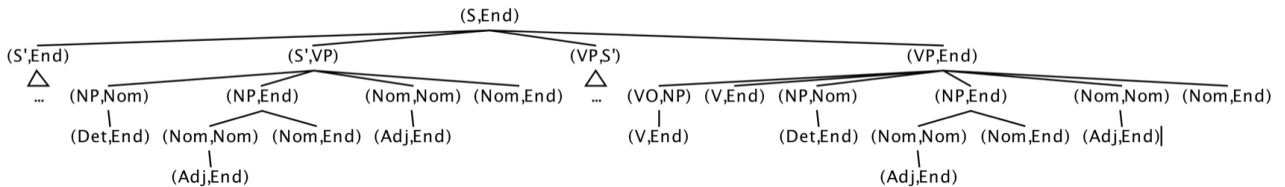


図6 絶対経路状態空間. 一部の部分木を省略して示している  
 Fig. 6 Absolute path state space.

$$\begin{aligned} \pi_{(X,Y)}((Z,W)) &= \frac{\theta_{X \rightarrow ZY}}{\sum_{\chi \in N} \theta_{X \rightarrow \chi Y}} \sum_{\chi \in N} \theta_{Z \rightarrow \chi W} \\ \pi_{(X,Y)}((Z, End)) &= \frac{\theta_{X \rightarrow ZY}}{\sum_{\chi \in N} \theta_{X \rightarrow \chi Y}} \sum_{\chi \in N \cup \Sigma} \theta_{Z \rightarrow \chi} \\ \pi_{(X, End)}((Z,W)) &= \frac{\theta_{X \rightarrow Z}}{\sum_{\chi \in N \cup \Sigma} \theta_{X \rightarrow \chi}} \sum_{\chi \in N} \theta_{Z \rightarrow \chi W} \\ \pi_{(X, End)}((Z, End)) &= \frac{\theta_{X \rightarrow Z}}{\sum_{\chi \in N \cup \Sigma} \theta_{X \rightarrow \chi}} \sum_{\chi \in N \cup \Sigma} \theta_{Z \rightarrow \chi} \\ A_{(X,Y)}((Z,W)) &= \delta(Z=Y) \sum_{\chi \in N} \theta_{Z \rightarrow \chi W} \\ A_{(X,Y)}((Z, End)) &= \delta(Z=Y) \sum_{\chi \in N \cup \Sigma} \theta_{Z \rightarrow \chi} \\ A_{(X, End)}(End) &= 1 \\ I((X,Y)) &= 1 \\ I((X, End)) &= \begin{cases} 0 & \text{if } \sum_{v \in \Sigma} \theta_{X \rightarrow v} > 0 \\ 1 & \text{otherwise} \end{cases} \end{aligned}$$

$$\begin{aligned} A_{(NP, Nom)}((Nom, End)) &= 0.55 \\ A_{(NP, End)}(End) &= 1.0, A_{(Nom, End)}(End) = 1.0 \\ B_{(Nom, End)}(bear) &= 0.636, B_{(Nom, End)}(trout) = 0.364 \end{aligned}$$

### 4.3 推論とEMアルゴリズム

図5に図3の構文木のHHMM表現を, 図6に構築したAPSSを示す. 与えられたPLAGを上記の手順に従ってHHMMに変換し, FBAアルゴリズムを用いて推論を行うことで, PLAGの推論を線形時間で行うことができる.

ただし, HHMMのパラメータ集合はPCFGとは異なるため, EMアルゴリズムの結果をInside-outsideアルゴリズムに基づく手法と等価にするためには, 生成規則の期待適用回数を明示的に求める必要がある. 位置  $(t, d)$  における生成規則  $X \rightarrow YZ$  の適用確率は, 以下と等しい.

$$\begin{aligned} & \sum_W P(Z_t^d = (X, Z), Z_t^{d+1} = (Y, W), F_{t-1}^{d+1} = 1 | O_{1:T}, F_T^1 = 1) \\ &= \sum_W \alpha_{bt}^d((X, Z)) \pi_{(X,Z)}((Y, W)) \beta_{bt}^{d+1}((Y, W)) \end{aligned}$$

このことから, 生成規則の期待適用回数  $\mu$  は以下で求まる.

$$\begin{aligned} \mu(X \rightarrow YZ) &= \sum_{t,d} \sum_W \alpha_{bt}^d((X, Z)) \pi_{(X,Z)}((Y, W)) \beta_{bt}^{d+1}((Y, W)) \\ \mu(X \rightarrow Y) &= \sum_{t,d} \sum_W \alpha_{bt}^d((X, End)) \pi_{(X, End)}((Y, W)) \beta_{bt}^{d+1}((Y, W)) \\ \mu(X \rightarrow v) &= \sum_{t: o_t = v} \sum_d \alpha_{bt}^d((X, End)) B_{(X, End)}(o_t) \beta_{bt}^d((X, End)) \end{aligned}$$

得られた  $\mu$  を用いて, 以下によって  $\theta$  を更新する.

例2 表1の文法を等価なHHMMに変換する. 状態集合  $\Pi$  は以下ようになる.

$$\begin{aligned} \Pi = \{ & (S, End), (S', End), (S', VP), (NP, Nom), \\ & (NP, End), (Det, End), (Nom, Nom), (Nom, End), \\ & (Adj, End), (VP, S'), (VP, End), (VO, NP), (V, End) \} \end{aligned}$$

上式に従って  $B, \pi, A$  を求める. たとえば, 状態  $(NP, Nom), (NP, End), (Nom, End)$  については以下のようになる.

$$\begin{aligned} \pi_{(NP, Nom)}((Det, End)) &= 1.0 \\ \pi_{(NP, End)}((Nom, Nom)) &= 0.45 \\ \pi_{(NP, End)}((Nom, End)) &= 0.55 \\ A_{(NP, Nom)}((Nom, Nom)) &= 0.45 \end{aligned}$$

$$\bar{\theta}_{X \rightarrow \gamma} = \frac{\mu(X \rightarrow \gamma)}{\sum_{r \in R_X} \mu(r)}$$

#### 4.4 PLAG における Inside-Outside アルゴリズム

本研究では、提案手法と Inside-outside アルゴリズムの実行時間の比較を行うが、PLAG は  $X \rightarrow Y$  の形式の規則 (Unit 規則) を含むため、直接 Inside-outside アルゴリズムを適用できない。しかし、文法が左非循環性を持つ場合、Unit 規則に順序付けを行うことができるため、以下のように修正を加えることで Inside-outside アルゴリズムを実行できる。修正された Inside アルゴリズムでは、非終端記号  $X$  を根としたときに部分系列  $o_{i:k}$  が生成される確率  $\alpha_{i,k}(X)$  をボトムアップに以下のように求める。

$$\begin{aligned} \alpha_{k,k}(X) &= \theta_{X \rightarrow o_k} + \sum_{X \rightarrow W \in R} \theta_{X \rightarrow W} \alpha_{k,k}(W) \\ \alpha_{i,k}(X) &= \sum_{X \rightarrow Y} \sum_{Z \in R_X} \sum_{i \leq j \leq k} \theta_{X \rightarrow Y Z} \alpha_{i,j}(Y) \alpha_{j,k}(Z) \\ &+ \sum_{X \rightarrow W \in R} \theta_{X \rightarrow W} \alpha_{i,k}(W) \end{aligned}$$

Outside アルゴリズムでは、非終端記号  $X$  が  $o_{i:k}$  を生成することを所与としたときに、 $o_{1:i-1}$  と  $o_{k+1:T}$  が  $G$  から生成される確率  $\beta_{i,k}(X)$  をトップダウンに以下のように求める。

$$\begin{aligned} \beta_{1,T}(S) &= 1 \\ \beta_{i,k}(X) &= \sum_{Y \rightarrow Z} \sum_{X \in R_1} \sum_{1 \leq j \leq i-1} \theta_{Y \rightarrow Z X} \alpha_{j,i-1}(Z) \beta_{j,k}(Y) \\ &+ \sum_{Y \rightarrow X} \sum_{Z \in R} \sum_{k+1 \leq j \leq T} \theta_{Y \rightarrow X Z} \alpha_{k+1,j}(Z) \beta_{i,j}(Y) \\ &+ \sum_{W \rightarrow X} \theta_{W \rightarrow X} \beta_{i,k}(W) \end{aligned}$$

これらの計算式を用いて、すべての可能な  $i, k$  の組について  $\alpha$  と  $\beta$  を求める。生成規則の期待適用回数  $\mu$  は以下のように求まる。

$$\begin{aligned} \mu(X \rightarrow o_k, k, k) &= \beta_{k,k}(X) \theta_{X \rightarrow o_k} \\ \mu(X \rightarrow Y Z, i, k) &= \sum_{j=i}^k \beta_{i,k}(X) \theta_{X \rightarrow Y Z} \alpha_{i,j}(Y) \alpha_{j,k}(Z) \\ \mu(X \rightarrow Y, i, k) &= \beta_{i,k}(X) \theta_{X \rightarrow Y} \alpha_{i,k}(Y) \end{aligned}$$

得られた  $\mu$  を用いて、以下によって  $\theta$  を更新する。

$$\bar{\theta}_{X \rightarrow \gamma} = \frac{\sum_{i,k} \mu(X \rightarrow \gamma, i, k)}{\sum_{r \in R_X} \sum_{i,k} \mu(r, i, k)}$$

## 5. 実験

本実験では、提案手法による推論と Inside-outside アルゴリズムによる推論および EM アルゴリズムが等価な結果を与えることを確認し、実行時間の比較を行う。提案手法 (「HHMM w/  $\mu$ 」とする) は、PCFG との等価性を保つ

表 2 データセット

Table 2 Dataset.

Corpus	#Sequences	Av. Length	#Vocabulary
Alice	1658	21.43	2584
br-phono	9790	9.79	50
br-text	9790	3.41	1380

表 3 実験に用いた文法

Table 3 Grammars in the experiment.

Grammar	N	$\Pi$	D	$max_d \Omega_d$	APSS 構築時間
JG1	29	45	13	1920	204 ms
JG2	12	18	8	48	26 ms
JG3	5	7	4	4	8 ms
ExG1	7	12	4	19	10 ms
ExG2	18	44	2	130	28 ms

ために、EM イテレーションごとに  $\mu$  の計算とパラメータの等価変換計算を行う。これらの計算のオーバーヘッドの大きさを検証するため、既存の HHMM の EM アルゴリズム (「HHMM w/o  $\mu$ 」とする) との実行時間の比較も行う。それぞれの手法は Java 言語で実装した。使用するデータセットを表 2 に示す。本実験では、Bernstein-Ratner コーパスを用いる [2], [3]。Bernstein-Ratner コーパスは、子供に向けた親の発話を書き起こしたデータセットであり、9,790 の発話からなる。「br-phono」は、音素を終端記号とした系列、「br-text」は書き起こされた単語を終端記号とした系列であり、それぞれ平均系列長は 9.79, 3.41 である。データセット「Alice」は不思議の国のアリスの全文であり、単語を終端記号とした 1,658 系列からなる。平均系列長は 21.43 と、他の 2 つのデータセットよりも長くなっている。

表 3 に、本実験で使用する文法を示す。JG1, JG2, JG3 は、Johnson ら [7] の論文および著者のホームページ<sup>\*4</sup>で公開されているソフトウェアに含まれる、音素系列を終端記号として生成する文法である。これらは、単語を表す非終端記号から、より細かい音素単位を表す非終端記号を階層的に生成する左循環のない文法であり、LAG の条件を満たしている。JG1 は最も生成規則の数が多く、JG3 は最も少ない。ExG1 は表 1 の文法の非終端記号 Det, Nom, Adj, V に、対象コーパスに含まれるすべての終端記号を生成する規則を (品詞とは無関係に) 追加した文法である。ExG2 は ExG1 をもとに作成した文法であり、非終端記号の数が ExG1 よりも大きい、APSS の階層数が小さい文法である。ExG2 は、APSS の構造が提案手法の実行時間に与える影響を調べるために用いる。表において、|N| は非終端記号の数、| $\Pi$ | は HHMM に変換した際の状態数、D

<sup>\*4</sup> <http://web.science.mq.edu.au/~mjohnson/Software.htm>  
「NAACL 2009 paper」の実験データとされている圧縮ファイルに含まれる、ファイル名がそれぞれ「colloc3-syllablesIF.lt (JG1)」「syllables.lt (JG2)」「unigram.lt (JG3)」の文法を用いた。



表 4 EM アルゴリズムによる対数尤度の変化  
Table 4 Log-likelihood for each EM iteration.

Iteration	(JG1,br-phono)		
	HHMM w/o $\mu$	HHMM w/ $\mu$	Inside-outside
1	-438616.217	-438616.217	-438616.217
2	-351445.043	-352087.565	-352087.565
3	-346561.799	-346976.950	-346976.950
4	-344904.561	-345410.794	-345410.794
5	-344412.819	-344982.431	-344982.431

表 5 1 系列あたりの推論の実行時間 (ms)  
Table 5 Execution time for inference per sequence.

Grammar	Corpus	HHMM	HHMM	Inside-outside
		w/o $\mu$	w/ $\mu$	
JG1	br-phono	21.653	9.253	10.235
JG2	br-phono	0.555	0.261	2.755
JG3	br-phono	0.050	0.042	0.661
ExG1	Alice	0.347	0.301	16.073
ExG1	br-text	0.046	0.038	0.075
ExG2	Alice	1.792	1.823	61.096
ExG2	br-text	0.233	0.158	0.265

は構築される APSS の深さである。  $max_d \Omega_d$  は 1 階層に含まれる APS の数の最大値であり、図 6 に示すとおり、上位階層からの状態の組合せの数の大きさを表す。また、それぞれの文法で APSS の構築にかかった時間は、いずれもデータセット全体を推論する時間と比較すれば十分小さいといえる。

事後確率の推論が等価であることを確認するため、提案手法と Inside-outside アルゴリズムそれぞれに基づく EM アルゴリズムを実行し、各反復における対数尤度を比較した結果を表 4 に示す。「HHMM w/  $\mu$ 」と「Inside-outside」は、EM アルゴリズムによる更新を行っても対数尤度が一致していることから、等価な事後確率分布の推論およびパラメータ更新を行っていることが確認できる。HHMM のパラメータ数は PCFG のパラメータ数よりも大きいため、「HHMM w/o  $\mu$ 」の対数尤度はより小さい値となっている。

表 5 に、EM アルゴリズムの 1 イテレーションの実行時間を系列数で割って求めた、系列あたりの実行時間を示す。実行時間は、100 回の試行の平均実行時間である。「HHMM w/  $\mu$ 」の実行時間には、PCFG のパラメータを更新し、再度等価変換を行う時間も含まれている。HHMM の方が PCFG よりもパラメータの数が多くなるため、「HHMM w/o  $\mu$ 」は「HHMM w/  $\mu$ 」よりも実行時間が大きくなる傾向がある。ただし、ExG2 の文法では終端記号を生成する規則が多く含まれており、語彙数の多い Alice データセットではパラメータ数の差が大きいため、この条件では  $\mu$  の計算および再変換を行わない分実行時間が小さくなっていると考えられる。

$\mu$  の計算を行う HHMM 手法では、すべての場合で Inside-

outside 手法よりも実行時間が小さい。系列長の違いに着目すると、Inside-outside では、平均系列長の短い br-text と比較して平均系列長の長い Alice の実行時間は大幅に増加している。Alice の平均系列長は br-text の約 6 倍であるが、Inside-outside では br-text と Alice で 200 倍以上の実行時間の差がある。一方、HHMM では、1 系列あたりの実行時間は br-text と Alice で約 7.5~12 倍程度の差であり、平均系列長に対するスケラビリティが大きく改善されていることが確認できる。

また、文法の大きさに着目すると、br-phono に対する HHMM の推論時間において、JG1 と JG2 の実行時間の差が 40 倍以上と大きいことが分かる。表 3 に示すとおり、JG1 と JG2 では階層の深さは大きく変わらないものの、非終端記号の組合せが爆発的に増加するため、 $max_d \Omega_d$  が JG1 と JG2 では大きく異なっており、このことが実行時間を大幅に増加させていると考えられる。また、ExG2 は ExG1 よりも階層数が小さいにもかかわらず実行時間が大きいことから、HHMM の推論の計算量は階層数よりも  $\Omega_d$  の大きさに強く依存していることが示唆される。

## 6. 結論

本研究では、新たに導入した左非循環文法を制約として、PCFG の推論および EM アルゴリズムを線形時間で行う手法を提案した。提案手法は、PCFG の等価 HHMM 変換に基づいて、潜在変数の事後確率分布の推論を行う。実験から、提案手法は特に長い系列長を含むデータセットにおいて大幅な実行時間の短縮が可能であることを明らかにした。

従来、センサデータ解析などの長い系列を対象とするドメインには PCFG の応用が困難であったが、提案手法によって応用範囲が広がる可能性が期待される。また、これまで HHMM は、状態空間の記述法がなくモデル構造の探索が容易ではなかったが、提案手法により CFG の記述法を用いて状態空間を構成できることから、系列ラベリングや系列セグメンテーションへの HHMM の応用の可能性が広がると考えられる。

謝辞 本研究の一部は、JSPS 科研費(課題番号 24800004, 25280110, 25540159) および筑波大学図書館情報メディア系プロジェクト研究 (Research Projects of Faculty of Library, Information and Media Science) の助成によって行われた。

## 参考文献

- [1] Avrahami-Zilberbrand, D. and Kaminka, G.: Incorporating Observer Biases in Keyhole Plan Recognition (Efficiently!), *Proc. Association for the Advancement of Artificial Intelligence* (2007).
- [2] Bernstein-Ratner, N.: The phonology of parent-child speech, *Children's Language*, Vol.6, pp.159-174 (1987).
- [3] Brent, M. and Cartwright, T.: Distributional regularity

- and phonotactic constraints are useful for segmentation, *Cognition*, Vol.61, pp.93-125 (1996).
- [4] Fine, S., Singer, Y. and Tishby, N.: The Hierarchical Hidden Markov Model: Analysis and Applications, *Machine Learning* (1998).
- [5] Geib, C. and Goldman, R.: Recognizing Plans with Loops Represented in a Lexicalized Grammar, *Proc. Association for the Advancement of Artificial Intelligence* (2011).
- [6] Huang, L. and Sagae, K.: Dynamic Programming for Linear-Time Incremental Parsing, *Proc. Association for Computational Linguistics* (2010).
- [7] Johnson, M. and Goldwater, S.: Improving nonparametric Bayesian inference: Experiments on unsupervised word segmentation with adaptor grammars, *Proc. Human Language Technologies: North American Chapter of the Association for Computational Linguistics* (2009).
- [8] Johnson, M., Griffiths, T. and Goldwater, S.: Bayesian Inference for PCFGs via Markov chain Monte Carlo, *Proc. North American Chapter of the Association for Computational Linguistics* (2007).
- [9] Lari, K. and Young, S.: The estimation of stochastic context-free grammars using the Inside-Outside algorithm, *Computer Speech and Language*, Vol.4, pp.35-56 (1990).
- [10] Lee, L.: Fast context-free grammar parsing requires fast Boolean matrix multiplication, *J. ACM*, Vol.49, No.1, pp.1-15 (2002).
- [11] Liang, P., Petrov, S., Jordan, M. and Klein, D.: The Infinite PCFG using Hierarchical Dirichlet Processes, *Proc. Empirical Methods on Natural Language Processing* (2007).
- [12] Murphy, K. and Paskin, M.: Linear Time Inference in Hierarchical HMMs, *Proc. Neural Information Processing Systems* (2001).
- [13] Ponvert, E., Baldrige, J. and Erk, K.: Simple unsupervised grammar induction from raw text with cascaded finite state models, *Proc. Association for Computational Linguistics: Human Language Technologies* (2011).
- [14] Pynadath, D. and Wellman, M.: Probabilistic State-Dependent Grammars for Plan Recognition, *Uncertainty in Artificial Intelligence* (2000).
- [15] Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition, *Proc. IEEE* (1989).
- [16] Schuler, W. and AdbelRahman, S.: Toward a Psycholinguistically-Motivated Model of Language Processing, *Proc. International Conference on Computational Linguistics* (2008).
- [17] Skounakis, M., Craven, M. and Ray, S.: Hierarchical Hidden Markov Models for Information Extraction, *Proc. International Joint Conference on Artificial Intelligence* (2003).
- [18] Wakabayashi, K. and Miura, T.: Forward-Backward Activation Algorithm for Hierarchical Hidden Markov Models, *Proc. Neural Information Processing Systems*, Vol.25 (Dec. 2012).
- [19] Wang, P. and Blunsom, P.: Collapsed Variational Bayesian Inference for PCFGs, *Proc. Computational Natural Language Learning* (2013).
- [20] 小島諒介, 佐藤泰介: アクセスログ分析における接頭部分列からのプラン認識, *人工知能学会論文誌*, Vol.29, No.3, pp.301-310 (2014).



若林 啓 (正会員)

1984年生。2012年法政大学大学院博士課程修了。博士(工学)。同年筑波大学図書館情報メディア系助教。機械学習, 自然言語処理の研究に従事。電子情報通信学会, 日本データベース学会, ACM各会員。

(担当編集委員 宮尾 祐介)