情報論的学習理論テクニカルレポート 2009
Technical Report on Information-Based Induction Sciences 2009 (IBIS2009)

ベイズ確率文脈自由文法のための高速構文木サンプリング法
Split Position Slice Sampler

武井俊祐[*]　牧野貴樹[†]　高木利久[*†‡]
Shunsuke Takei[*]　Takaki Makino[†]　Toshihisa Takagi[*†‡]

Abstract: We propose a new tree sampling algorithm for Bayesian probabilistic context-free grammar (PCFG) called Split Position Slice Sampler. Split Position Slice Sampler is developed based on Beam Sampling method that is a fast MCMC sampling algorithm for Bayesian Hidden Markov Model, and adapted to Bayesian PCFG. This tree sampling method can be combined with Metropolis-Hastings sampler to constitute an efficient grammar sampling algorithm for Bayesian PCFG. Because this algorithm does not involve any approximation, more efficient inference is achieved without losing accuracy. We evaluate our approach by comparing with an existing method in a small artificial corpus.

Keywords: Bayesian inference, PCFG, MCMC

1 Introduction

本研究は，ベイズ拡張された確率文脈自由文法(PCFG)に対する効率的なサンプリングアルゴリズムを構築することで，高速かつ高精度な文法学習を可能にすることを目的とする．ベイズ PCFG モデルに対する精度の高いパラメータ推定手法としては，Johnson らの動的計画法を利用したマルコフ連鎖モンテカルロ法（MCMC）による手法 [1]が知られているが，モデルが複雑になるにつれて増大する計算コストが問題となる．そこで我々は，ベイズ拡張された隠れマルコフモデル（HMM）のための高速な MCMC 法として知られている Beam Sampling 法[2]を PCFG に応用することで Johnson らの手法を高速化する．隠れマルコフモデルにおける Beam Sampling 法は，動的計画法と Slice Sampling 法[3]を利用することにより，ベイズ隠れマルコフモデルの高速なパラメータ推定を可能にする手法であるが，これをそのままの形式で PCFG の枠組みへ応用しようとするとき，HMM において各時刻において割り当てられている補助変数を PCFG で利用される内側確率表に直接対応付けることができず，アルゴリズムが構築できない．本論文ではこれを内側確率表上の分割位置に補助変数を対応付ける形式として Split Position Slice Sampler という手法を新たに提案することで Beam Sampler の枠組みを拡張し，これを Johnson らの手法へ組み込むことによって高精度かつ高速なベイズ PCFG モデルのパラメータ推定法を構築する．

1.1. Motivation

ベイズモデルとは，統計的機械学習分野において学習モデルにベイズ統計の手法を導入するもので，ベイズ化された学習モデルは，与えられたデータに対するモデルパラメータの推定値と不確実性の範囲を確率分布として表現する．結果としてベイズ統計における学習は，従来の最尤法による点推定に比べ汎化性能に優れ，過学習の問題も生じにくい．また，あらかじめ推定したい対象に何らかの知識がある場合，それを事前確率分布という形式でモデルに導入することも可能であり，より良い推定ができる場合もある．さらに，近年注目を集めているベイズモデルの拡張であるノンパラメトリックベイズモデルでは，無限次元のパラメータ空間を仮定し，データを表現する最適なモデルを推定することでモデル選択の問題を解決しており，様々な統計学習モデルに導

*東京大学大学院　新領域創成科学研究科　情報生命科学専攻，〒277-8568　千葉県柏市柏の葉 5-1-5　総合研究棟 609, tel. 04-7136-3973,
e-mail: takei_shunsuke@cb.k.u-tokyo.ac.jp, Department of Computational Biology, Graduate School of Frontier Science, The University of Tokyo, General Research Building 609, 5-1-5 Kashiwa-no-ha, Kashiwa, Chiba, 277-8568 JAPAN
†東京大学　総括プロジェクト機構
‡情報・システム研究機構ライフサイエンス統合データベースセンター

入されており[4][5][6]，今後より広範な問題に応用されることが期待されている．

ベイズ学習モデルにおいて学習の対象である個々のパラメータは確率変数で表現されているため，従来の最尤法とは異なりしばしば計算コストの高い高次元の期待値計算を伴う方法でパラメータ推定を行う．特に高い精度が要求されるような場合においては高精度なパラメータ推定手法として知られるMCMC法が利用されるが，膨大なサンプルを必要とするため計算コストが高く，ベイズモデルの応用における問題となっている．

PCFG[9]は計算機科学分野だけでなくバイオインフォマティクス分野など[10]幅広い分野で活用されている一般性の高い確率モデルであり，そのベイズモデルについても期待が高い．近年，変分ベイズ法[7][8]のような近似に基づく高速パラメータ推定手法が提案され，さかんに研究されているものの，高精度な推定を必要とする場合は依然として膨大なサンプルを必要とする計算コストの高いMCMCのような手法が必要となる．

PCFGに対するMCMC法を構成するためには，単純にはGibbs Samplerのような手法を適用することが考えられるが，PCFGのパラメータ間に強い相互依存があり，またパラメータ空間も大きいことから，Gibbs Samplerでは収束が遅く効率が悪い．これに対しJohnsonらは構文木の確率に応じた構文木のサンプリングによるMetropolis-Hastingsアルゴリズムを構築し，より効率的なサンプリング手法を提案している[1]．しかしJohnsonらの手法は依然として計算コストが大きく，大規模データに適用するのには困難を伴う．

近年Bayesian HMMの高速なサンプリング手法としてBeam Samplerが提案された．この手法は近似などを伴わず，精度を保ったままBayesian HMMのパラメータ推定を高速に行うことに成功しており，幅広い応用が見込まれている．この手法は，Slice Samplingの手法を適用し，隠れ状態間の遷移確率分布をスライスで“間引く”ことで，近似などを伴わず，精度を保ったままBayesian HMMのパラメータ推定を高速に行うことに成功している．HMMにおける遷移確率分布は，PCFGにおいては書き換え規則の確率分布に対応するため，正しいスライス手法が与えられれば，同様な高速化を達成できることが期待できる．

そこで我々は，Johnsonらの手法にBeam Samplerの枠組みを導入することで高い精度を保ったままBayesian PCFGのパラメータ推定を高速に行うことのできるMCMC法を構築する．

2 Background

2.1. PCFG

文脈自由文法(CFG)は文の生成過程を明らかにするモデルであり，$G = (V_N, V_T, S, R)$ の4-タプルにより定義される．V_T は終端記号集合，V_N は非終端記号集合，R は生成規則の集合である，S は開始記号であり，$S \in V_N$ である．ここでは生成規則に関してチョムスキー標準形のみを扱う．すなわち，$A \to BC$ または $A \to s$ の形式の生成規則のみを持つ．ここで，$A, B, C \in V_N$，$s \in V_T$ である．

確率文脈自由文法(PCFG)は，CFGの各々のルールに対して確率値を割り振ったものであり，(G, θ) と定義される．$\boldsymbol{\theta}$ は$|R|$次元の実数ベクトルであり，$\boldsymbol{\theta}$ のそれぞれの要素は θ_r で表され，これは $r \in R$ の生成規則の確率値であることを表す．たとえば，$\theta_{A \to BC}$ ならば $A \to BC$ の規則の確率，$\theta_{A \to s}$ ならば $A \to s$ の規則の確率を表す．ここで，確率の定義から $\theta_r \geq 0$ かつ $\sum_{a \in V_T \cup V_N \times V_N} \theta_{A \to a} = 1$ である必要がある．

2.2. Bayesian PCFG

Bayesian PCFGでは，生成規則のパラメータ $\boldsymbol{\theta}$ を確率変数 $\mathrm{P}(\boldsymbol{\theta})$ として扱う．ベイズアプローチにおいて離散パラメータは，しばしばその扱いやすさからDirichlet分布のような共役な確率分布によって表される．本論文におけるBayesian PCFGについてもDirichlet分布を導入したPCFGを仮定している．具体的には，$\boldsymbol{\theta}$ の事前分布 $\mathrm{P}_D(\boldsymbol{\theta} \mid \boldsymbol{\alpha})$ は次のようなDirichlet分布で表現されるものとする．

Dirichlet分布を $\mathrm{P}_D(\cdot)$，Γ をガンマ関数，$A \to *$ の形式の文法規則を R_A，$\boldsymbol{\theta}$ における $A \to *$ の形式の文法規則についてのパラメータベクトル $\boldsymbol{\theta}_A$，Dirichlet分布のハイパーパラメータベクトルを $\boldsymbol{\alpha}$，$\boldsymbol{\alpha}$ における $A \to *$ の形式の文法規則についてのハイパーパラメータを $\boldsymbol{\alpha}_A$ で表現すると，

$$\mathrm{P}_D(\boldsymbol{\theta} \mid \boldsymbol{\alpha}) = \prod_{A \in V_N} \mathrm{P}_D(\boldsymbol{\theta}_A \mid \boldsymbol{\alpha}_A) \qquad (1)$$

ここで,

$$\mathrm{P}_D(\boldsymbol{\theta}_A \mid \boldsymbol{\alpha}_A) = \frac{1}{C(\boldsymbol{\alpha}_A)} \prod_{r \in R_A} \theta_r^{\alpha_r - 1} \tag{2}$$

$$C(\boldsymbol{\alpha}_A) = \frac{\prod_{r \in R_A} \Gamma(\boldsymbol{\alpha}_A)}{\Gamma(\sum_{r \in R_A} \boldsymbol{\alpha}_A)} \tag{3}$$

である.

一般的に，モデルのパラメータは未知であり，データからそれを推定しなければならない．ベイズアプローチにおける推定の対象はデータ観測後の事後確率分布であり,

$$\begin{aligned} \mathrm{P}(\boldsymbol{\theta} \mid \mathbf{w}) &\propto \mathrm{P}(\mathbf{w} \mid \boldsymbol{\theta}) \mathrm{P}(\boldsymbol{\theta} \mid \boldsymbol{\alpha}) \\ &= \prod_{i=1}^{n} \mathrm{P}(w_i \mid \boldsymbol{\theta}) \mathrm{P}(\boldsymbol{\theta} \mid \boldsymbol{\alpha}) \end{aligned} \tag{4}$$

の左項を求めることである．ここで,
$\mathbf{w} = (w_1, ..., w_n)$ は終端記号列からなるデータ集合であり， w_i が個別の終端記号列を意味する.

2.3. Gibbs Sampler

Dirichlet 分布は共役であるという性質上，事後確率分布も Dirichlet 分布となるが，Bayesian PCFG のように複数の Dirichlet 分布を組み合わせた場合は解析的積分が不可能であり，その推定には MCMC 法や変分ベイズなどの手法が用いられる． MCMC 法のもっとも単純な手法の 1 つである Gibbs Sampler を Bayesian PCFG のパラメータ推定法として用いようとする場合，パラメータ $\boldsymbol{\theta}$ と構文木 $\mathbf{t}$ を交互にサンプリングすることでアルゴリズムが構築される．ここで， $\mathbf{t} = (t_1, ..., t_n)$ であり，それぞれの t_i は終端記号列 w_i に対する構文木を意味する．詳しい導出は Johnson らの論文に譲るとして，ここでは具体的な Gibbs Sampler のアルゴリズムを示す.

構文木 t において文法規則 $r \in R$ が使われた回数を $f_r(t)$， $\mathbf{t}$ における $A \to *$ の形式の文法規則が使われた回数のベクトルを $\mathbf{f}_A(\mathbf{t})$ とすると，Bayesian PCFG の Gibbs Sampler は， $\mathbf{t}$ を固定して $\boldsymbol{\theta}$ についてのサンプリング,

$$\mathrm{P}(\boldsymbol{\theta} \mid \mathbf{t}, \mathbf{w}, \boldsymbol{\alpha}) = \prod_{A \in V_N} \mathrm{P}_D(\boldsymbol{\theta}_A \mid \mathbf{f}_A(\mathbf{t}) + \boldsymbol{\alpha}_A) \tag{5}$$

$\boldsymbol{\theta}$ を固定して $\mathbf{t}$ についてのサンプリング,

$$\mathrm{P}(\mathbf{t} \mid \boldsymbol{\theta}, \mathbf{w}, \boldsymbol{\alpha}) = \prod_{i=1}^{n} \mathrm{P}(t_i \mid w_i, \boldsymbol{\theta}) \tag{6}$$

を交互に繰り返すこととなる．パラメータ $\boldsymbol{\theta}$ についてのサンプリングは，与えられた構文木 $\mathbf{t}$ における各文法規則の使用回数から容易に計算できるが，構文木 $\mathbf{t}$ の具体的なサンプリングについては自明ではない.

Johnson らは構文木のサンプリングについて，動的計画法を用いた効率的な手法を提案している．まず,
ある終端記号列 $\mathbf{s} = (s_1, ..., s_n)$ が与えられたとき，その内側確率を

$$p_{k,k}^{A} = \theta_{A \to s_k} \tag{7}$$

$$p_{i,k}^{A} = \sum_{A \to BC \in R} \sum_{i \le j < k} \theta_{A \to BC} p_{i,j}^{B} p_{j+1,k}^{C} \tag{8}$$

のように計算する．ここで， $p_{i,k}^{A}$ は非終端記号 A が終端記号列 $s_i, ..., s_k$ を生成する確率を意味する.

次に作られた内側確率表の確率にしたがい構文木を再帰的にサンプリングする．これを疑似コードで示すと,

```
Function SAMPLE(A,i,k)
If i = k
    return TREE(A, s_k)
Else
    (j,B,C) = MULTI(A,i,k)
    return TREE(A,SAMPLE(B,i,j),SAMPLE(C,j,k))
```

となる．ここで，関数 $\mathrm{SAMPLE}(A,i,k)$ はある構文木のノードにおいて非終端記号 A が終端記号列 $s_i, ..., s_k$ を生成するとき，そのノード以下の構文木を確率的にサンプリングする関数である．また，関数 $\mathrm{MULTI}(A,i,k)$ はどの分割点で構文木が枝分かれするのか，その位置 j と子ノードの非終端記号 B,C を確率的に返す関数であり，その確率は

$$\mathrm{P}(j,B,C) = \frac{\theta_{A \to BC} p_{i,j}^{B} p_{j+1,k}^{C}}{p_{i,k}^{A}} \tag{9}$$

と表すことができる.

2.4. Metropolis-Hasting Sampler

PCFG における各々のパラメータ間には強い依存があるため，パラメータ $\boldsymbol{\theta}$ を個別にサンプリングする Gibbs Sampler は十分なサンプルを得るために多大な時間が必要になる．この問題に対し Johnson らは Bayesian PCFG のパラメータ $\boldsymbol{\theta}$ を積分消去し，構文木のサンプリングと Metropolis-Hastings の枠組みによるサンプルの確率的な受理から構成されるサンプリングアルゴリズムを提案している．この手法では，依存の強い $\boldsymbol{\theta}$ のサンプリングは回避されるため，Gibbs Sampler に比べ速い収束が見込まれ，さらに $\boldsymbol{\theta}$ のサンプリングにかかる計算コストも不要となり，効率的な Bayesian PCFG のサンプリングアルゴリズムであるといえる．

このアルゴリズムでは，Gibbs Sampler とは違いパラメータ $\boldsymbol{\theta}$ のサンプリングを行わず，構文木をサンプリングしたのち，そのサンプルを確率的に受諾もしくは拒否する．いま， $\mathbf{w} = (w_1,...,w_n)$ を終端記号列の $\mathbf{s}$ の集合とし，それぞれの終端記号列 w_i に対応する構文木を $\mathbf{t} = (t_1,...,t_n)$ で表す． t'_i が新しくサンプルされた構文木， $\mathbf{t}_{-i}$ が t_i を除いた構文木サンプル群とすると，そのサンプルの受理確率は

$$
\begin{aligned}
A(t_i, t'_i) &= \min\left\{1, \frac{P(t'_i \mid w_i, \mathbf{t}_{-i}, \alpha) P(t_i \mid w_i, \boldsymbol{\theta}')}{P(t_i \mid w_i, \mathbf{t}_{-i}, \alpha) P(t'_i \mid w_i, \boldsymbol{\theta}')}\right\} \\
&= \min\left\{1, \frac{P(t'_i \mid \mathbf{t}_{-i}, \alpha) P(t_i \mid w_i, \boldsymbol{\theta}')}{P(t_i \mid \mathbf{t}_{-i}, \alpha) P(t'_i \mid w_i, \boldsymbol{\theta}')}\right\}
\end{aligned}
\tag{7}
$$

となる．ここで，

$$
\mathrm{P}(t_i \mid \mathbf{t}_{-i}, \boldsymbol{\alpha}) = \prod_{A \in V_N} \frac{C(\boldsymbol{\alpha}_A + \mathbf{f}_A(\mathbf{t}))}{C(\boldsymbol{\alpha}_A + \mathbf{f}_A(\mathbf{t}_{-i}))} \tag{9}
$$

である． $\boldsymbol{\theta}'$ は θ_r' を要素に持つベクトルである． θ_r' は $\mathbf{t}_{-i}$ と $\boldsymbol{\alpha}$ が与えられたときのパラメータ θ_r の期待値で，

$$
\theta_r' = \frac{f_r(\mathbf{t}_{-i}) + \boldsymbol{\alpha}_r}{\sum_{r' \in R_A} f_{r'}(\mathbf{t}_{-i}) + \boldsymbol{\alpha}_{r'}} \tag{10}
$$

として計算される．

以上が Johnson らの Bayesian PCFG のための構文木サンプリングの枠組みである．この手法はパラメータ $\boldsymbol{\theta}$ と構文木 $\mathbf{t}$ を交互にサンプリングする Gibbs Sampler に比べ効率的ではあるが，パラメータ空間が大きい場合，内側確率の計算に多大な計算コストがかかり，依然として大規模データには不向きである．

2.5. Beam Sampler

Beam Sampler は，ノンパラメトリックベイズ化された HMM のための高速な MCMC アルゴリズムであり，HMM のパラメータ推定をする際しばしば用いられる動的計画法（この場合 Forward-Backward）と Slice Sampling 法をサンプリングに応用した，HMM のための高速な MCMC 法である．HMM のための Gibbs Sampler は，各時刻の隠れ状態を交互にサンプリングする形式で行われるが，Beam Sampler は各時刻の隠れ状態を個別にサンプリングするのではなく，動的計画法を利用して隠れ状態列をひとつのサンプルとして取得する形式で行われるため，Johnson らの手法同様パラメータ間の依存の問題が解決される．また， Slice Sampling に基づく補助変数（スライサー）を各時刻について導入し，

1.前回サンプルされた状態遷移列に従いスライサーをサンプリング
2.状態遷移分布をスライスすることで，各時刻における可能な状態遷移を間引
3.間引きされた分布にしたがい隠れ状態列をサンプル

という手順でサンプリングを行う．ここでスライスとは，対象となる確率分布において，スライサーの値以上のもののみを考慮するよう分布を“間引く”操作であり，スライスされた分布は等確率でサンプリングされる．この手順により各時刻で考慮すべき状態遷移数が減るために計算が高速に行われる．Beam Sampler における分布のスライスと，スライスされた分布のサンプリングという処理は近似処理ではなく，Slice Sampling を援用したものでありそれ自体がサンプリングの手続きとなっている．そのため Beam Sampler は高精度なパラメータ推定というマルコフ連鎖モンテカルロ法の性質を維持したまま高速化に成功した手法であるといえる．

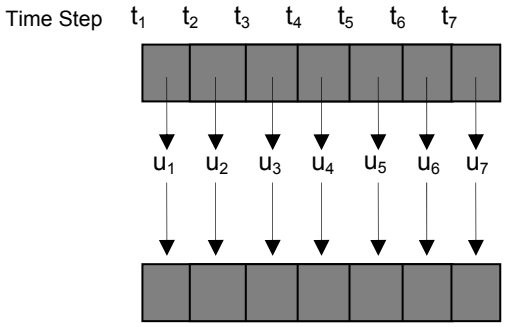

Figure 1.　HMM の Beam Sampler におけるスライサーの対応
u_iがスライサーを意味する．観測列は省略している．

3 Method

3.1. PCFG へのスライスサンプリングの導入

我々のアプローチは Johnson らの手法に Beam Sampler の枠組みを導入することで高速化を図るものである． Beam Sampler でサンプルとする隠れ状態列は PCFG における構文木にあたり，Beam Sampler で用いている動的計画法は Johnson らの手法では内側確率を計算し，構文木をサンプリングすることと等価である．すなわち，Slice Sampling 法を内側確率の計算に利用すればよいので，Johnson らのアルゴリズムが

1. 内側確率表を計算
2. 構文木をサンプリング
3. 構文木を Accept/Reject

という手順で構成されていたのに対し，提案手法は

1. スライサーをサンプリング
2. 生成確率をスライスして，内側確率表の各セルにおける文法規則適用分布を間引き
3. 構文木をサンプリング
4. 構文木を Accept/Reject

という手順で構成されることとなる．しかし PCFG と HMM のモデルの違いから Beam Sampler をそのまま適用しようとしてもアルゴリズムは構築できない．

問題となるのは，Beam Sampler における前回のサンプルにおける各時刻の状態遷移確率からスライサーの値をサンプリングするという手続きである．HMM では各時刻についてスライサーがそれぞれ割り当てられており，これを PCFG にそのまま適用しようとすると内側確率表の各セル，つまり考えうる全ての構文木のノード位置について，スライサーを導入するという形式となる．この場合前回のサンプルにあたるのが構文木であり，スライサーのサンプルに必要なのは，構文木の各ノードで用いられた文法規則の確率値である．しかしながらこの場合，前回のサンプル，すなわちひとつの構文木からすべてのセルのスライサーを作ることができない．なぜなら，サンプルとして得られた構文木は，内側確率表全てのセルに一対一対応するだけノード数を保持しておらず，従ってこのような配置でスライサーを導入しようとした場合，全てのスライサーをサンプリングするには情報が足りない．これは n を文中の単語数とした場合，構文木は 2n-1 個のノードしか持たず，一方内側確率表には n(n+1)/2 個のセルがあるということからも分かる．

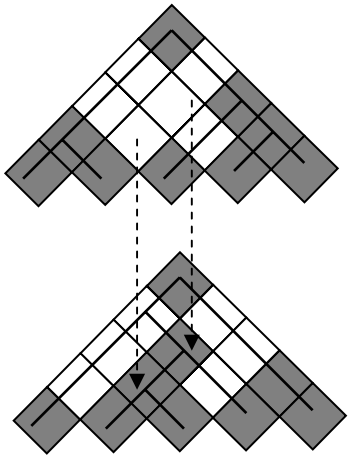

Figure 2.　内側確率表の各セルにスライサーを導入した場合
構文木の形が変わると対応がつかないセルが存在する
破線矢印は対応のつかないセルを意味する．

3.2. Split Position Slice Sampler

そのため我々はスライサーの配置に関する新たなルールとして，Split Position Slice Sampler という形式を提案する．この形式におけるスライサーは内側確率表において最終的に終端記号を出力するセルを対応付ける担当するスライサー **u** と，構文木の“分割点”を担当するスライサー **v** の 2 種類に分けて扱われる． **u** については Beam Sampler 同様，前回そのセルで使われた文法規則の確率値をもとに値を決める．これは，対応する内側確率表の場所が毎回必ず使われるためである．一方“分割点”を担当するスライサー **v** は，構文木において位置 j で木が分岐したとして，そのときの文法規則の確率値から作られる．

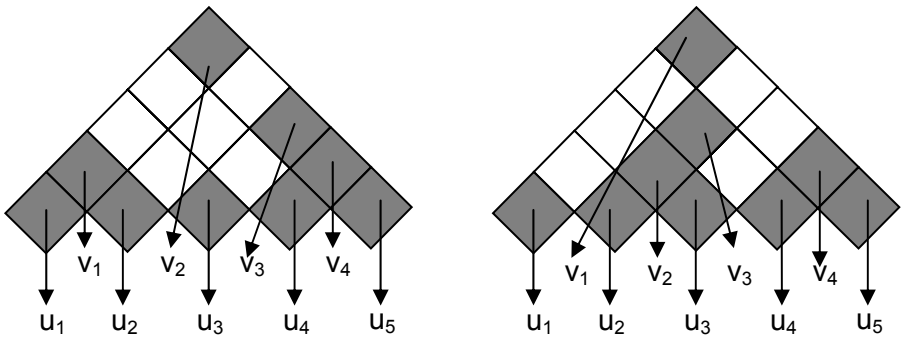

Figure 3. 内側確率表の最下列，および“分割点”にスライサーを導入した場合
構文木の形が変わってもスライサーの対応がついている

この手法では，内側確率計算の際，ひとつのセルでの計算に複数のスライサーが利用されることになるが，完成した構文木においては，ひとつのスライサーが二度以上利用されることは無い．また，導入すべきスライサーは最下列セル数 n，分割点数 n-1 であり，構文木の持つノード数と一致するため，前述のような問題は起こらず一対一対応が出来る．これを我々は Split Position Slice Sampler と呼ぶことにする．これにより内側確率表に対するスライサーが正しく定義され，アルゴリズムが構築可能となる．以下では Split Position Slice Sampler を利用した構文木のサンプリングアルゴリズムの具体的な手順を示す．

終端記号列 $\mathbf{s}$ に対する前回の構文木サンプルにおいて，s_k を導出した文法規則の確率値を π_k，構文木の分割点 j における文法規則の確率値を ρ_j と定義する．関数 $\mathrm{I}(C)$ は，条件 C が真のとき $\mathrm{I}(C)=1$，偽の時 $\mathrm{I}(C)=0$ となる関数である．

1. Sampling $\mathbf{u}, \mathbf{v}$ step

$$u_k \sim \mathrm{Uniform}(0, \pi_k) \tag{11}$$
$$v_j \sim \mathrm{Uniform}(0, \rho_j) \tag{12}$$

2. Inside-Filtering Step

$$p^A_{k,k} = \mathrm{I}(u_k < \theta_{A \to w_k}) \tag{13}$$
$$p^A_{i,k} = \sum_{A \to BC \in R} \sum_{i \le j < k} \mathrm{I}(v_j < \theta_{A \to BC}) p^B_{i,j} p^C_{j+1,k} \tag{14}$$

3. Tree-Sampling Step

Function SAMPLE(A,i,k)
If $i = k$
  return TREE(A, s_k)
Else
  $(j,B,C) = \mathrm{MULTI}(A,i,k)$
  return TREE$(A, \mathrm{SAMPLE}(B,i,j), \mathrm{SAMPLE}(C,j,k))$

関数 MULTI における分割点のサンプリング確率は

$$\mathrm{P}(j,B,C) = \frac{\mathrm{I}(\theta_{A \to BC}) p^B_{i,j} p^C_{j+1,k}}{p^A_{i,k}} \tag{15}$$

と置き換えられる．

以上のアルゴリズムは Johnson らの手法における構文木のサンプリングステップについての置き換えであり，その他の部分についての変更が無い．つまり，Gibbs Sampler に組み込んで $\boldsymbol{\theta}$ のサンプリングと組みあわせるか，Metropolis-Hastings Sampler と組み合わせてサンプリングされた構文木の確率的な受理をすることになる．

以上により， Beam Sampling 法の枠組みを PCFG へと適用した高速な MCMC 法である Split Position Slice Sampler が構築された．この手法は，単純に考えた場合に対応のつかない，形の違う構文木同士に対応をつけることで Beam Sampler の枠組みを拡張した．これにより，Beam Sampler の枠組みは Bayesian PCFG のためのサンプリングアルゴリズムに導入可能となり，高速なサンプラーが構築された．

4 Experiments

人工的な CFG 文法から小規模なコーパスを生成し，本手法と従来手法の比較を行った．コーパスは文法規則 S→xSy，S→zSw，S→SS，S→ ε から生成され，平均文長 9 単語の 100 文を訓練データ，平均文長 13 単語の 20 文をテストデータとして用い，従来の Johnson の手法と提案手法についてそれぞれ 50 回の実験を行い，計算速度とテストデータの対数尤度について評価を行った．

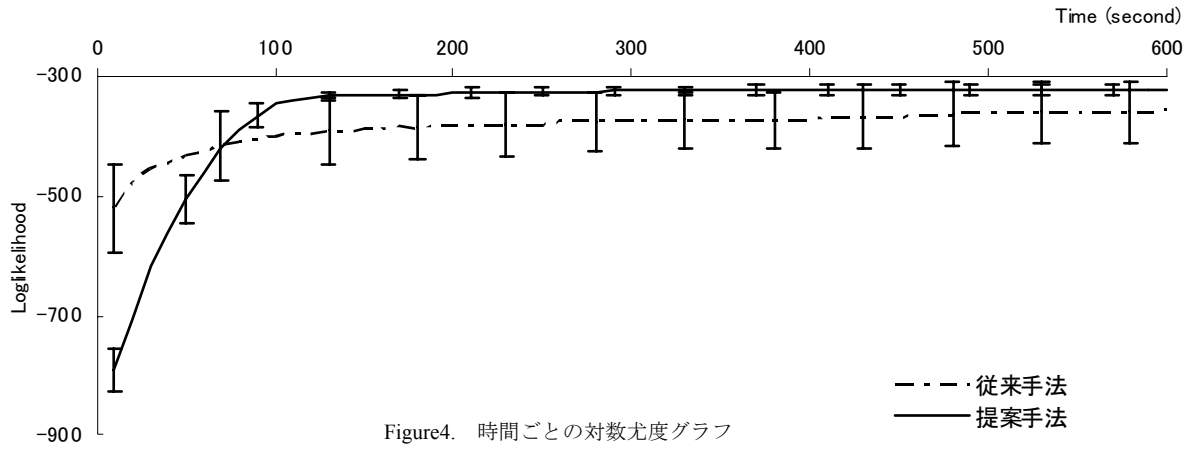

Figure4.　時間ごとの対数尤度グラフ

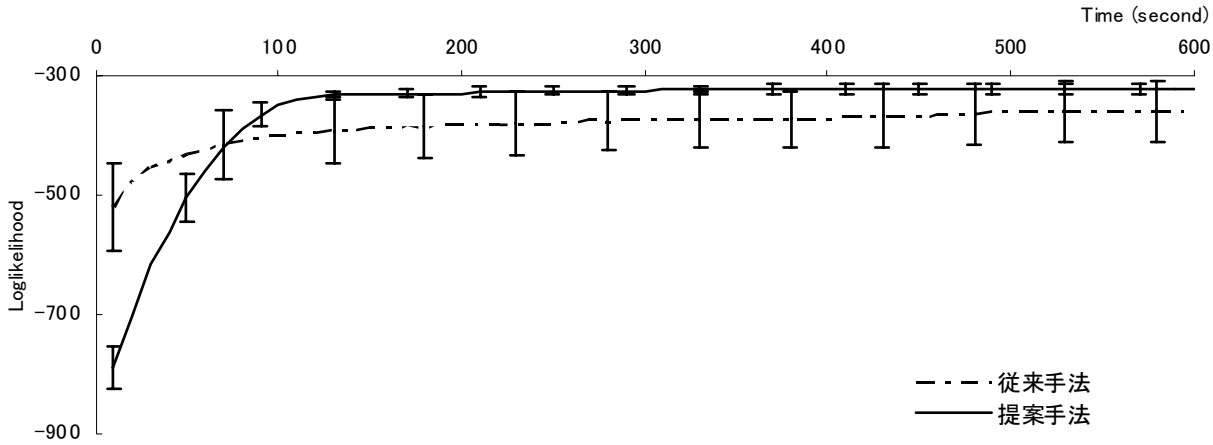

Figure5.　Sampling Step ごとの対数尤度グラフ

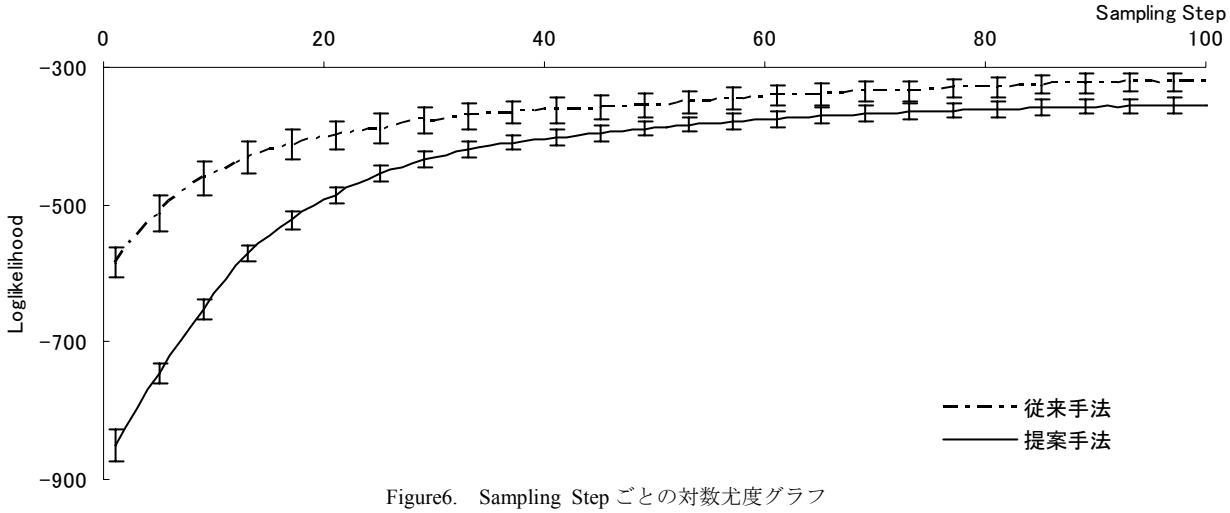

Figure6.　Sampling Step ごとの対数尤度グラフ

Figure2 はサンプリング 1 ステップごとにかかる時間，Figure3 は時間ごとに見たコーパスの対数尤度，Figure4 はサンプリング 1 ステップごとに見たテストデータの対数尤度である．グラフは 50 回の実験による結果の平均値であり，誤差棒は 95%の信頼区間を示している．

Figure2 より，提案手法は従来手法よりも急速に対数尤度が時間，サンプリングステップを追うごとに上昇しており，提案手法はより効率的に学習していることがわかる．

学習の経過をみると，Figure4 で明らかなように，1 ステップごとの対数尤度の上昇は従来手法の方が早い．これは，提案手法ではスライサーを通してサンプリング前後の構文木が依存しているためであると考えられる．しかしながら，Figure3 からは，1 ステップの計算速度が従来手法に比べて提案手法が大幅に高速化されているために，1 ステップあたりの学習効率の悪さを補って余りある高速化効果が得られ，その結果，従来手法に比べ計算速度の向上がなされていることがわかった．

5 まとめ

本論文では Beam Sampler の枠組みを Split Position Slice Sampler へと拡張し，それを従来手法に組み込むことにより，Bayesian PCFG のための高速なパラメータ推定手法を構築し，それを小規模な実験によって評価した．我々の手法では一切の近似を含まず，高い精度でのパラメータ推定が可能であるという MCMC 法の性質を保っているため，高速かつ高精度な手法であるといえる．このアルゴリズムは CKY アルゴリズムを伴えばどのような問題にも適用可能であり，ベイズ拡張された PCFG のスーパーセットのようなモデルにも適用可能であることが期待できる．

また，Beam Sampler が元々はノンパラメトリック拡張された HMM において提案されたものであるということから，ノンパラメトリック拡張された PCFG への適用の可能性もある．しかし，ノンパラメトリック HMM の場合とは違い，動的計画法の向きが逆であることから，スライサーで分布をスライスする際，対象となる分布のサーチに打ち切ることができないことから，単純な適用はできない．そのため，本手法をノンパラメトリック PCFG へ適用する場合，無限次元の分布を近似なしで推定する手法である Retrospective Sampling を併用する必要がある．

今後は PCFG のスーパーセットのためのサンプリングアルゴリズムへの拡張などのアルゴリズム的な応用や，実際の大規模データに対して適用するなどの実用面での応用の可能性を探るつもりである．

参考文献

[1] Johnson, M., Griffiths, T. L. & Goldwater, S. (2007). Bayesian Inference for PCFGs via Markov Chain Monte Carlo. In *Proceedings of North American Chapter of Association for Computational Linguistics – Human Language Technologies.*

[2] Gael, J. V., Saatci, Y. Teh, Y. W. & Gharamani. Z. (2008) Beam Sampling for the Infinite Hidden Markov Model. In *Proceedings of the 25th International Conference on Machine Learning.*

[3] Neal, R. M. (2003). Slice sampling. *The annals of Statistics*..

[4] Teh, Y. W., Jodan, M. I., Beal, M. & Blei, D. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association.*.

[5] Blei, M., Gharamani, Z. & Rasmussen, C. (2002). The infinite hidden Markov model. In *Advances in Neural Information Proceeding Systems.*

[6] Liang, P., Petrov. S., Jordan., M. I. & Klein. D. (2007) The infinite PCFG using hierarchical Dirichlet processes. In *Proceedings of the 2007 joint Conference on Empirical Methods in Natural language Processing and Computational Natural Language Learning.*

[7] Attias, H. (1999). Inferring Parameters and Structure of Latent Variable Models by Variational Bayes. In *Proceedings of Uncertainty in Artificial Intelligence.*

[8] Kenichi, K., Yoshitaka, K., Taisuke, S. (2004). Variational Bayesian Approach to Probabilistic Context-Free Grammar based on Dynamic Programing. *Journal of Information Processing Society, Japan.*

[9] Chaniak, E. (1996). Treebank grammars. *Association for the Advancement of Artificial Intelligence* .

[10] Sakakibara, Y.; Brown, M.; Hughey, R.; Mian, I. S.; Sjölander, K.; Underwood, R. C.; and Haussler, D. (1994). Stochastic Context-Free Grammars for tRNA Modeling. *Nuc. Acids Res.*