JMLR: Workshop and Conference Proceedings 34:153–166, 2014 Proceedings of the 12th ICGI

Inferring (k, l)-context-sensitive probabilistic context-free grammars using hierarchical Pitman-Yor processes

Chihiro Shibata SHIBATACHH@STF.TEU.AC.JP
School of Computer Science, Tokyo University of Technology

Editor: Alexander Clark, Makoto Kanazawa and Ryo Yoshinaka

Abstract

Motivated by the idea of applying nonparametric Bayesian models to dual approaches for distributional learning, we define (k, l)-context-sensitive probabilistic context-free grammars (PCFGs) using hierarchical Pitman-Yor processes (PYPs). The data sparseness problem that occurs when inferring context-sensitive probabilities for rules is handled by the smoothing effect of hierarchical PYPs. Many possible definitions or constructions of PYP hierarchies can be used to represent the context sensitivity of derivations of CFGs in Chomsky normal form. In this study, we use a definition that is considered to be the most natural as an extension of infinite PCFGs defined in previous studies. A Markov Chain Monte Carlo method called blocked Metropolis-Hastings (MH) sampling is known to be effective for inferring PCFGs from unsupervised sentences. Blocked MH sampling is applicable to (k, l)-context-sensitive PCFGs by modifying their so-called inside probabilities. We show that the computational cost of blocked MH sampling for (k, l)-context-sensitive PCFGs is $O(|V|^{l+3}|s|^3)$ for each sentence s, where V is a set of nonterminals. This cost is too high to iterate sufficient sampling times, especially when $l \neq 0$, thus we propose an alternative sampling method that separates the sampling procedure into pointwise sampling for nonterminals and blocked sampling for rules. The computational cost of this sampling method is $O(\min\{|s|^l, |V|^l\}(|V||s|^2 + |s|^3))$.

Keywords: nonparametric Bayesian model, distributional learning, Gibbes sampling

1. Introduction

Methods that divide sentences into contexts and constituents, and then construct grammars by grouping or aligning their common parts are generally called *distributional learning*. The idea of distributional learning has been studied and implemented in deterministic systems (van Zaanen, 2000) and generative models (Klein and Manning, 2002).

In recent studies, theoretical aspects of distributional learning have been developed in deterministic learning frameworks such as learning in the limit and the Minimally Adequate Teacher (MAT) model. A series of subclasses of context-free grammars (CFGs) have been shown to be learnable using distributional approaches. Distributional learning are formally divided into *primal* and *dual* approaches (Yoshinaka, 2011, 2012). Roughly speaking, the difference between the primal and dual approaches is that the nonterminal symbols in the inferred CFGs are represented by substrings or contexts, respectively. A subclass of CFGs have the q-FCP (Finite Context Property) if their nonterminals are characterized by q substrings. CFGs that have the q-FCP can be learned by dual approaches.

In the context of probabilistic learning, several positive theoretical results have been obtained for distributional approaches. Clark (2006) showed that non-terminally separated (NTS) languages are PAC(probably approximately correct)-learnable from positive samples if the target distributions satisfy properties represented by several parameters, such as μ-distinguishability. All NTS languages are able to generated from CFGs that have the 1-FKP.

Other fundamental classes that can be learned by primal and dual approaches have also been shown to be learnable in a PAC sense (Shibata and Yoshinaka, 2013a). These results are useful steps from the viewpoint of understanding the learnable subclasses of CFGs, but their algorithms are not efficient in terms of required amount of data if the goal is to learn grammars with high accuracy from real-world data. In this paper, we apply nonparametric Bayesian methods, which were developed recently, to learning CFGs according to an idea of dual approaches for distributional learning.

We are interested in defining and learning the subclasses of probabilistic CFGs using nonparametric Bayesian models that are analogous to dual approaches. Let $P_{\mathrm{substr}}(\cdot|(l,r))$ be a distribution over the substrings that a context (l,r) accepts, which can be generated from a Pitman-Yor Process (PYP, as described in Sec. 3.1):

$$P_{\mathrm{substr}}(\cdot|(l,r)) \sim PYP(\theta, d, H_A),$$

where A is a nonterminal and the base measure H_A depends only on A and not on (l,r) [1]. H_A is analogous to $L(A)$ in the deterministic dual approach, i.e., after a set of substrings are drawn from H_A, the probabilities over the set depend on (l,r). Intuitively, the advantage of the definition given above is that the distribution over $L(A)$ can depend on (l,r), unlike probabilistic context-free grammars (PCFGs). In PCFGs, the probabilities of contexts and of substrings are independent for every nonterminal A: $P(S \overset{*}{\Rightarrow} lAr \overset{*}{\Rightarrow} lur) = P(S \overset{*}{\Rightarrow} lAr)P(A \overset{*}{\Rightarrow} u)$.

There are two problems with the generative model described above. One problem is the sparsity of the substrings that each context accepts. To avoid this, we can take advantage of the hierarchies of the base measures of PYPs by assuming that each context generates rules instead of substrings with probability $P_{\mathrm{rule}}(\cdot|lAr)$ and:

$$P_{\mathrm{rule}}(\cdot|lAr) \sim PYP(\theta, d, H_{\pi(lAr)}), \tag{1}$$

where π is an arbitrary function that decreases the length of l or r from the outside, i.e., $\pi(lAra) = (l,r)$ or $\pi(al,r) = (lAr)$ for any letter a. $H_{\pi(lAr)}$ is obtained recursively.

Another problem is that the generative model defined in Eq. 1 fails to define the probabilities for generated sentences, unless all the rules are linear. Eq. 1 assumes that the context (l,r) of A is known before a rule $A \to \alpha$ is generated. The contexts for rules should be pairs of sequences that include nonterminals around the nonterminal, to which the objective rule is applied. The Details of the definition of the generative model with PYPs are described in Sec. 4.2.

1. For example, if the target grammar is a c-deterministic CFG (Shirakawa and Yokomori, 1993), A is uniquely identified from (l,r).

2. Related Work

Recent studies have proposed nonparametric Bayesian models, which are related to CFGs and can relax the property of independence for the respective rules of PCFGs. Cohn et al. (2010); Shindo et al. (2012) showed that learning tree substitution grammars (TSGs) from parse tree-annotated data yielded high accuracy when parsing sentences. The rules of TSGs are elementary trees or tree fragments, instead of the production rules used in CFGs. TSGs can capture the dependencies between contexts and constituents. The adaptor grammars proposed by Johnson et al. (2007a) comprise a general framework that can weaken the independence between contexts and constituents in PCFGs. Pitman-Yor adaptor grammars were defined as a subclass of adaptor grammars, where the production probabilities depend on the number of subtrees in the derivation trees.

3. Preliminaries

A CFG is a tuple $G = \langle \Sigma, V, R, S \rangle$, where Σ is the set of terminal symbols, V is the set of nonterminal symbols, R is the set of production rules, and S is the initial symbol. We denote the empty string by λ.

3.1. Hierarchical PYPs

Let X be a countable set and $PYP(\theta, d, H)$ be a PYP over X, where θ and d are the parameters of the Poisson-Dirchlet distribution and H is an arbitrary distribution over X, which is called the *base measure* (Pitman and Yor, 1997; Ishwaran and James, 2003; Teh, 2006). $PYP(\theta, d, H)$ is a distribution over distributions over X. A sequence of real values $(p_1, p_2, \cdots)$ drawn from the Poisson-Dirchlet distribution gives a distribution over natural numbers. Each natural number t, called a *table*, is also assigned to some $x \in X$ with a probability $H(x)$. Thus, the probability that an element x is drawn c times via table t from the sample distribution mentioned above is $H(x){p_t}^c$. In the Chinese restaurant process (CRP) representation, the probability that a table and an element of X is sampled is given as follows by marginalizing all p_t:

$$P(x_{m+1}, t_{m+1} | x_1, \cdots x_m, t_1, \cdots, t_m) = \frac{1}{m+\theta} \cdot \begin{cases} (\theta + Td)H(x_{m+1}) & \text{if } t_{m+1} \text{ is a new table,} \\ c(x_{m+1}, t_{m+1}) - d & \text{otherwise,} \end{cases} \tag{2}$$

where T is the number of different tables in $t_1, \cdots, t_m$ and $c(x, t)$ is the number of occurrences of (x, t) in $(x_1, t_1), \cdots, (x_m, t_m)$.

If the base measure H, which is a distribution over X, is defined as being sampled from some other PYP recursively, $P(\theta, d, H)$ is called a hierarchical PYP.

If some distribution over X is drawn from a PYP, it is often the case that this distribution is never drawn from that PYP subsequently. In this case, because θ and d are identified by and are not related, other than by a distribution P that is sampled only once, we can omit these parameters in this study:

$$P \sim PYP(H).$$

3.2. Hierarchies of Base Measures for Sets of Finite Sequences

Let X and Y be some countable set such as Σ and V. Let P be a conditional distribution over X^m given $y = b_1 \cdots b_n \in Y^n$, which is taken from a PYP where the base measure is H_y:

$$P(\cdot|y) \sim PYP(H_y).$$

To estimate P from data, it is necessary to count every occurrence of $x = a_1 \cdots a_m \in X^m$ given y. A smoothing method needs to be applied because this is too sparse to obtain an appropriate estimate if n or m is large. Hierarchical PYPs with appropriate hierarchies of base measures allow good smoothing. Two methods are often used for constructing hierarchical PYPs: decreasing the length of y and decreasing the length of x.

For the former method, we refer to the base measure H_y as an *aggregation* of y if

$$H_y \sim PYP(H_{y'}),$$

where y' is some substring of y with a fixed position, i.e., $y' = b_i \cdots b_j$. For the latter method, we refer to the base measure H_y as a *decomposition* of X^m if

$$H_y(a_1 \cdots a_m) = \prod_{i=1}^{m} H_{y,i}(a_i), \quad \text{where} \quad H_{y,i} \sim PYP(K_y),$$

and K_y is some distribution over X given y.

4. Generative Models for CFGs using Hierarchical PYPs

For simplicity, CFGs are restricted to Chomsky normal form (CNF) in this study. In addition, we assume that for each nonterminal A, the length of α is fixed such that $A \to \alpha$.

4.1. PCFGs using Hierarchical PYPs

Let A be a nonterminal, the rules of which have two nonterminals on the left. We consider an infinite number of rules in $R_A = \{A \to BC | B, C \in V\}$, where the probabilities are taken from a PYP assigned to each A:

$$P(A \to \cdot\,\cdot) \sim PYP(H_A)$$

where H_A is the base measure of the PYP, the domain of which is $V \times V$. We can define H_A in two different manners by selecting whether the decomposition of $V \times V$ or the aggregation of A occurs first.

In the former case, the base measure H_A is defined as follows. B and C are assumed to be taken independently from $H_{A,1}$ and $H_{A,2}$, i.e., $H_A(BC) = H_{A,1}(B)\ \ H_{A,2}(C)$. Then, $H_{A,i}$ is recursively defined for each i as

$$H_{A,i} \sim PYP(H_{\lambda,i}).$$

A base measure $H_{\lambda,i}$ is again taken recursively from a PYP, the base measure of which is the uniform distribution:

$$H_{\lambda,i} \sim PYP(\text{Uniform}).$$

Liang et al. (2007) defined infinite PCFGs using the hierarchy described above.

In the latter case, H_A is given by the aggregation of A first. H_A is assumed to be taken from a PYP, where the base measure H_λ is shared for all $A \in V$:

$$H_A \sim PYP(H_\lambda).$$

Then, H_λ is defined as a decomposition of $V \times V$, which is the domain of H_λ: $H_\lambda(BC) = H_{\lambda,1}(B)H_{\lambda,2}(C)$. $H_{\lambda,i}$ are again sampled recursively from a PYP:

$$H_{\lambda,i} \sim PYP(\text{Uniform}).$$

In the case where a nonterminal A has rules with a terminal $a \in \Sigma$ on the right, $|\Sigma|$ is often too large to assume that $P(A \to \cdot)$ has a uniform prior distribution. The distribution from which a is generated is assumed to be sampled from a PYP:

$$P(A \to \cdot) \sim PYP(H_A),$$

where H_A is the uniform distribution [2].

4.2. (k,l)-context-sensitive Probabilistic CFGs using Hierarchical PYPs

4.2.1. context-sensitive Probabilities for CFGs

Suppose that a sentence w is derived with the rules $r_1, \cdots, r_k$ in a CFG G.

$$S \Rightarrow^{r_1} \alpha_1 \Rightarrow^{r_2} \cdots \Rightarrow^{r_m} \alpha_m = w$$

In probabilistic CFGs, the probability of w with the above derivation is given by the product of the probabilities for all rules, $P(r_1) \cdots P(r_m)$. Here, we define the context-sensitive probabilities for a rule $A \to \beta$ in G,

$$P(\alpha_{\mathtt{L}} A \alpha_{\mathtt{R}} \Rightarrow \alpha_{\mathtt{L}} \beta \alpha_{\mathtt{R}}) = P(A \to \beta | (\alpha_{\mathtt{L}}, \alpha_{\mathtt{R}})).$$

The probability of w with the above derivation is given by $P(r_1|(\varepsilon,\varepsilon)) \cdots P(r_m|(\alpha_{l,m-1}, \alpha_{r,m-1}))$, where $\alpha_{l,i}, \alpha_{r,i}$ is defined as: $\alpha_i = \alpha_{l,i}\beta_i\alpha_{r,i}$ and $r_i = A_i \to \beta_i$. We refer to a context-sensitive probabilistic CFG as (k,l)-*context-sensitive* if for all contexts $(\alpha_{\mathtt{L}}, \alpha_{\mathtt{R}})$ and rules $A \to \beta$, $P(A \to \beta|(\alpha_{\mathtt{L}}, \alpha_{\mathtt{R}})) = P(A \to \beta|(\gamma_{\mathtt{L}}, \gamma_{\mathtt{R}}))$, where $\gamma_{\mathtt{L}}$ is a suffix of $\alpha_{\mathtt{L}}$ with a length of no more than k and $\gamma_{\mathtt{R}}$ is a prefix of $\alpha_{\mathtt{R}}$ with a length of no more than l.

4.2.2. Construction of Hierarchies of Base Measures

The definition of $P(A \to \beta|(\alpha_{\mathtt{L}}, \alpha_{\mathtt{R}}))$ requires different parameters for each $(\alpha_{\mathtt{L}}, \alpha_{\mathtt{R}})$. Recent studies show that nonparametric Bayesian models such as hierarchical PYPs have excellent smoothing capacities in this situation. As described in Section 3.2, the definitions of hierarchies of PYPs have many variations, which depend on the order of the aggregation of contexts $(\alpha_{\mathtt{L}}, \alpha_{\mathtt{R}})$ and the decomposition of the right-hand side of the rule β. In

2. In the case that Σ is a set of words, H_A is often assumed to be a character-level language model used to represent morphological information (Clark, 2003). For example, Blunsom and Cohn (2011) used a bigram model to define H_A. Mochihashi et al. (2009) proposed to use a hierarchical Pitman-Yor language model (Teh, 2006) as H_A.

the following, we first aggregate $\alpha_{\mathtt{R}}$ and $\alpha_{\mathtt{L}}$ recursively and then decompose β. We write $P(A \to \beta|(\alpha_{\mathtt{L}}, \alpha_{\mathtt{R}}))$ as $P(\beta|(A, \alpha_{\mathtt{L}}, \alpha_{\mathtt{R}}))$, since $\sum_{\beta} P(A \to \beta|(\alpha_{\mathtt{L}}, \alpha_{\mathtt{R}})) = 1$. $P(\cdot|(A, \alpha_{\mathtt{L}}, \alpha_{\mathtt{R}}))$ is a distribution over $V \times V$ or Σ in CNF.

$$P(\cdot|(A, \alpha_{\mathtt{L}}, \alpha_{\mathtt{R}})) \sim PYP(H_{A,\pi(\alpha_{\mathtt{L}},\alpha_{\mathtt{R}})}).$$

$\pi(\alpha_{\mathtt{L}}, \alpha_{\mathtt{R}})$ is defined as follows:

$$\pi(\alpha_{\mathtt{L}}, \alpha_{\mathtt{R}}) = \begin{cases} (\pi_{\mathtt{L}}(\alpha_{\mathtt{L}}), \alpha_{\mathtt{R}}) & \text{if } |\alpha_{\mathtt{L}}| > |\alpha_{\mathtt{R}}|, \\ (\alpha_{\mathtt{L}}, \pi_{\mathtt{R}}(\alpha_{\mathtt{R}})) & \text{otherwise,} \end{cases}$$

where $\pi_{\mathtt{L}}(a\beta) = \beta$ and $\pi_{\mathtt{R}}(\beta a) = \beta$ for $\beta \in (V \cup \Sigma)^*$ and $a \in V \cup \Sigma$. The base measure $H_{A,(\alpha_{\mathtt{L}},\alpha_{\mathtt{R}})}$ is defined recursively using π as:

$$H_{A,\pi^i(\alpha_{\mathtt{L}},\alpha_{\mathtt{R}})} \sim PYP(H_{A,\pi^{i+1}(\alpha_{\mathtt{L}},\alpha_{\mathtt{R}})}),$$

for $i = 0, \cdots, |\alpha_{\mathtt{L}}\alpha_{\mathtt{R}}|$. After these recursive definitions of base measures are complete, we define

$$H_{A,(\lambda,\lambda)} \sim PYP\left(\Pi_{i=1}^{|\beta|} J_{A,i}\right), \quad \text{where} \quad J_{A,i} \sim PYP(J_i),$$

and $J_i \sim PYP(\text{Uniform})$.

4.2.3. Order of Derivation

With context-sensitive probabilities, different orders of derivation give different probabilities for a single derivation tree. For example, the two derivations $d_1 : S \Rightarrow AB \Rightarrow aB \Rightarrow ab$ and $d_2 : S \Rightarrow AB \Rightarrow Ab \Rightarrow ab$ have the same probability $P(S \to AB)P(A \to a)P(B \to b)$ in PCFG, but not in a CFG with context-sensitive probabilities:

$$\Pr(d_1) = P(S \to AB|(\varepsilon, \varepsilon))P(A \to a|(\varepsilon, B))P(B \to b|(a, \varepsilon)),$$
$$\Pr(d_2) = P(S \to AB|(\varepsilon, \varepsilon))P(A \to a|(\varepsilon, b))P(B \to b|(A, \varepsilon)).$$

Thus, we assume that all of the derivations are leftmost derivations in the following. Note that $\alpha_{\mathtt{L}}$ is in Σ^* and $\alpha_{\mathtt{R}}$ is in V^* in the leftmost derivation. $\alpha_{\mathtt{L}}$ is known directly from the sample sentence whereas $\alpha_{\mathtt{R}}$ is not. This implies that from a computational costs viewpoint, the length of $\alpha_{\mathtt{R}}$ is problematic whereas the length of $\alpha_{\mathtt{L}}$ is not, as described in Sec. 5.1.

5. Inference Method

Gibbs sampling is a representative Markov chain Monte Carlo (MCMC) algorithm, which is known to give relatively high accuracy approximations as a method for the Bayesian inference of probabilistic models where the marginalization of all unknown parameters is unfeasible, such as HMMs and PFAs (Shibata and Yoshinaka, 2013b). Gibbs sampling can be roughly divided into two types: pointwise sampling and blockwise sampling, based on how many variables are changed each time. Generally speaking, blockwise sampling is known to be less trapped in a local optimum while its computational cost is relatively high.

It is not straightforward to apply pointwise sampling for inferring the derivation trees of PCFGs in CNF because we have to consider many possible derivation trees. To apply

pointwise Gibbs sampling, a single rule that is used to generate some sentence is replaced whereas the other rules remain fixed. However, if the length of the right-hand side of the rule is changed by the replacement, this replacement forces the subsequent rules to change because the sequence of nonterminals is shifted. Thus, to ensure that pointwise Gibbs sampling is achieved successfully, rule replacements have to be limited so the shape of derivation tree is not changed. However, these limited replacements fail to sample the derivation trees of the given sentences.

The blocked Metropolis-Hastings (MH) sampling method proposed by Johnson et al. (2007b) is blockwise Gibbs sampling, but with a slight modification to ensure the correctness of sampling. Their method replaces a derivation tree immediately. The new derivation tree is sampled in the following steps. 1) The expected production probabilities are calculated according to the counts of their occurrence. 2) The inside probabilities (Lari and Young, 1990) are calculated. 3) The proposed derivation tree is generated randomly from S using the inside probabilities and rejected with some probability in order to compensate for the difference from the true probability.

5.1. Blocked Sampling for (k,l)-Context-Sensitive Probabilistic CFGs

For CFGs with (k,l)-context-sensitive probabilities, blocked MH sampling can be applied by modifying the definitions of the inside probabilities. Thus, we now review the definitions of inside probabilities for PCFGs. Let $s(i,j)$ denote a substring $a_i \cdots a_j$ of a sentence $s = a_1 \cdots a_m$. When $j < i$, let $s(i,j) = \lambda$. In PCFGs, an inside probability P_{in} for a given nonterminal A and a substring $s(i,j)$ is defined as: $P_{\text{in}}(A|i,j) = \Pr(A \overset{*}{\Rightarrow} s(i,j))$. The probability of the derivation $A \Rightarrow BC \overset{*}{\Rightarrow} yC \overset{*}{\Rightarrow} yz$ is calculated by $P(A \to BC) \Pr(B \overset{*}{\Rightarrow} y) \Pr(C \overset{*}{\Rightarrow} z)$. Thus, $P_{\text{in}}(A|i,j)$ is calculated recursively as:

$$P_{\text{in}}(A|i,j) = \sum_{A \to BC \in R_A} \sum_{k=i}^{j-1} P(A \to BC) P_{\text{in}}(B|i,k) P_{\text{in}}(C|k+1,j).$$

In (k,l)-context-sensitive CFGs, the inside probability P_{in} for a given nonterminal A, a substring $w = a_i \cdots a_j$ in a sentence $s = a_1 \cdots a_m$, and a sequence of nonterminals $\alpha \in V^*$ such that $|\alpha| \leq l$ is defined as follows:

$$P_{\text{in}}(A|i,j,\alpha) = \Pr(xA\alpha \overset{*}{\Rightarrow} xw\alpha),$$

where x is a suffix of $a_1 \cdots a_{i-1}$ and $|x| \leq k$. For (k,l)-context (x,α), the probability of the derivation $xA\alpha \Rightarrow xBC\alpha \overset{*}{\Rightarrow} xyC\alpha \overset{*}{\Rightarrow} xyz\alpha$ is given as $\Pr(xA\alpha \Rightarrow xBC\alpha \overset{*}{\Rightarrow} xyC\alpha \overset{*}{\Rightarrow} xyz\alpha) = P(A \to BC|(x,\alpha)) \Pr(xBC\alpha \overset{*}{\Rightarrow} xyC\alpha) \Pr(xyC\alpha \overset{*}{\Rightarrow} xyz\alpha)$. Consequently, $P_{\text{in}}(A|i,j,\alpha)$ is calculated recursively as:

$$P_{\text{in}}(A|i,j,\alpha) = \sum_{A \to BC \in R_A} \sum_{k=i}^{j-1} P(A \to BC|(x,\alpha)) P_{\text{in}}(B|i,k,C\alpha') P_{\text{in}}(C|k+1,j,\alpha), \quad (3)$$

where α' is a prefix of α such that $|\alpha'| \leq l-1$.

Note that x is identified uniquely by (i,j) and a given sentence s. Thus, we do not require any left-hand side context x as a condition of the inside probabilities. This means

that the length of the left-hand side context, or k, has no effect on the computational cost when the inside probabilities are calculated. By contrast, the length of the right-hand side context affects both the memory requirements and the computational cost. As shown by Eq. 3, the inside probabilities require an array with a size of $|V|^{l+1}|s|^2$ for each sentence s. Since $|V|^2|s|$ additions are required for each element of the array, the computational cost required to build all the inside probabilities for each sentence is $|V|^{l+3}|s|^3$.

5.2. Fast Sampling Method for (k,l)-Context-Sensitive Probabilistic CFGs

Building the table of inside probabilities is the main computational cost that is incurred during blocked MH sampling. As shown in the previous section, the cost of CFG with (k,l)-context-sensitive probabilities is too high to allow the iteration of samples for all sentences, unless l is sufficiently small or zero. In practice, even if $l = 0$, for example, if both $|s|$ and $|V|$ are 10 and the number of sentences is 10,000, 10G $\times$ (some constant) computational steps are required for each sampling iteration. It may be difficult to assume that l is nonzero in practical problems. The inside probabilities P_{in} shown in Eq. 3 require $|V| \times |V|^l$ values for each substring $s(i,j)$. In order to reduce the computational costs and memory requirements, we propose the combination of: 1) blocked MH sampling for derivation meshes; and 2) pointwise sampling for nonterminals in each substrings, where the derivation meshes are defined as follows.

Suppose that a derivation tree is given for the sentence s. A substring $s(i,j)$ is referred to as *constituent* if $s(i,j)$ is derived from one nonterminal and *distuent* (Klein and Manning, 2002) otherwise.

Each constituent substring has a parent substring and two children substrings can be identified naturally from a derivation tree. By contrast, if a substring $s(i,j)$ is distuent, $s(i,j)$ does not represent a node of the derivation tree. For the purposes of sampling, a triplet of a nonterminal, a parent substring, and children substrings are assigned to each distuent substring. We refer to these as a *pseudo-nonterminal*, a *pseudo-parent*, and *pseudo-children* respectively. Note that, since the grammar is in CNF for both a constituent and a distuent, a parent of $s(i,j)$ is either $s(i',j)$ s.t. $i' < i$ or $s(i,j')$ s.t. $j < j'$, and the children are $s(i,h)$ and $s(h,j)$ s.t. $i \leq h \leq j$.

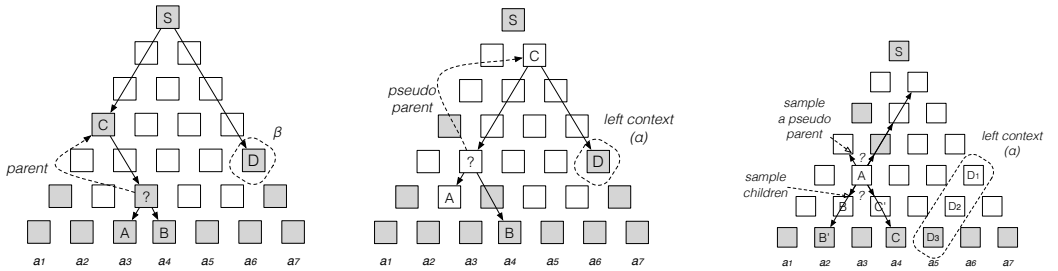
We refer to a map from each constituent or distuent substring $s(i,j)$ to a triplet mentioned above as a *derivation mesh* in this paper. A derivation mesh is equivalent to a derivation tree if all the distuent substrings are removed from the domain.

The probabilities that pseudo nonterminals are assigned to distuent substrings are required to be taken appropriately since the distuent substrings become constituent substrings when a new derivation tree or mesh is resampled.

5.2.1. Pointwise sampling for nonterminals

Assume that the derivation mesh is fixed. Let $A(i,j)$ be a nonterminal assigned to $s(i,j)$. For a constituent substring $s(i,j)$, since the derivation tree is fixed, it is easy to identify α such that $S \overset{*}{\Rightarrow} s(1,j-1)A(i,j)\alpha$. The parent and children are also identified from the derivation tree. $A(i,j)$ is updated using pointwise Gibbs sampling.

For example, let B, D, and E be the parent and children of $A(i,j)$, i.e., $S \overset{*}{\Rightarrow} xB\alpha \Rightarrow xAC\alpha \Rightarrow xDEC\alpha$, where $x = s(1,j-1)$ and $A = A(i,j)$. First, the values in the counting



(a) Pointwise sampling for constituent nonterminals

(b) Pointwise sampling for distuent nonterminals

(c) Blocked sampling for derivation meshes

Figure 1: Combination of pointwise and blocked sampling: (a)(b) sampling nonterminals, (c) sampling derivation meshes.

table are decreased for the rule $B \to AC$ with context (x, α) and the rule $A \to DE$ with context $(x, C\alpha)$. Next, a new nonterminal is sampled instead of A, according to the probabilities obtained from the CRP (Eq. 2).

For a distuent substring $s(i,j)$, we identify the left pseudo-context α by tracing its pseudo-ancestors. Let the sequence of ancestors of $s(i,j)$ from the side of $s(i,j)$ be $s(i_1,j_1)$, $s(i_2,j_2)$, $\cdots$, $s(i_n,j_n)$, where $s(i_1,j_1) = s(i,j)$ and $s(i_n,j_n) = s$. Let $j'_1, \cdots, j'_{n'}$ be the sequence obtained by removing j_h such that $j_h = j_{h-1}$ from $j_1, \cdots, j_n$. Since we assume that the derivations are leftmost, $A(j'_1+1, j'_2)A(j'_2+1, j'_3)\cdots A(j'_{n-1}+1, j'_{n'})$ is the left context. After identifying α, the remaining sampling of the new nonterminals is similar to that of the constituent substrings.

5.2.2. Blocked sampling for derivation meshes

Suppose that a nonterminal $A(i,j)$ is fixed for each substring $s(i,j)$. First, we remove the counts of pairs of rules and the (k,l)-contexts used in the derivation tree of the sentence s from the counting table of the hierarchical PYPs. For any (k,l)-context (x,α) and nonterminals A, $P(A \to \cdot\,\cdot\,|(x,\alpha))$ and $P(A \to \cdot|(x,\alpha))$ can easily be calculated as a posterior of the hierarchical PYPs using CRP(Eq. 2).

Recall that $A(\cdot,\cdot)$ is fixed in this sampling step. We define the modified inside probabilities restricted by $A(\cdot,\cdot)$ as follows:

$$P^{+}_{\text{in}}(i,j,\alpha) = \Pr(s(1,i-1)A(i,j)\alpha \Rightarrow^{A(\cdot,\cdot)} s(1,i-1)s(i,j)\alpha),$$

where $A(i,j) \Rightarrow^{A(\cdot,\cdot)} s(i,j)$ means that the nonterminals are restricted by $A(\cdot,\cdot)$, i.e., if a substring $s(i',j')$ of $s(i,j)$ is derived from one nonterminal, that nonterminal is $A(i',j')$. We can calculate $P^{+}_{\text{in}}(i,j,\alpha)$ for all i,j,$\alpha \in V^l$ in a bottom-up manner, as follows:

$$P^{+}_{\text{in}}(i,j,\alpha) = \sum_{h=i}^{j-1} P\left(A(i,j) \to A(i,h)A(h+1,j)|(s(1,i-1),\alpha)\right) P^{+}_{\text{in}}(i,h,A(h{+}1,j)\alpha) P^{+}_{\text{in}}(h{+}1,j,\alpha). \tag{4}$$

After P^+_{in} have been calculated, a new derivation mesh is sampled in a top-down manner, i.e., for each $s(i,j)$, we determine its children $s(i,h)$ and $s(h+1,j)$ according to the probabilities in the right-hand side of Eq. 4, except α is determined by tracing the ancestors of $s(i,j)$, as described in the previous section.

Whether $s(i,j)$ is constituent or not is determined naturally as follows: 1) $s(1,m)$ is constituent; and 2) if some substring is constituent, its children are constituent. Thus, for constituent substrings, tracing the ancestors is well-defined since each of them has one constituent parent. However, this is not the case for distuent substrings. To trace the ancestors, even for distuent substrings, it is necessary to determine a map from the distuent to parent substrings. We sample a parent $s(i,h)$ (or $s(h,j)$) for each distuent $s(i,j)$, according to the following probability for each h:

$$\begin{aligned}&\Pr(S \Rightarrow^{A(\cdot\cdot)} s(1,i-1)A(i,h)\alpha(i,h) \Rightarrow s(1,i-1)A(i,j)A(j+1,h)\alpha(i,h) \Rightarrow^{A(\cdot\cdot)} s(1,m))\\ &=P^+_{\text{out}}(i,h,\alpha(i,h))P(A(i,h) \to A(i,j)A(j+1,h)|(s(1,i-1),\alpha(i,h)))\\ &\quad P^+_{\text{in}}(i,j,A(j+1,h)\alpha(i,h))P^+_{\text{in}}(j+1,h,\alpha(i,h)),\end{aligned}$$

where $\alpha(i,h)$ is the right-hand context of $A(i,h)$, which is identified uniquely since part of the derivation tree that comprises the upper nodes greater than $s(i,j)$ has already been sampled. After sampling the parent of $s(i,j)$, $P^+_{\text{out}}(i,j,\alpha(i,j))$ is calculated as follows:

$$\begin{aligned}P^+_{\text{out}}(i,j,\alpha(i,j)) = \sum_h &P^+_{\text{out}}(i,h,\alpha(i,h))P(A(i,h) \to A(i,j)A(j+1,h)|(s(1,i-1),\alpha(i,h)))\\ &P^+_{\text{in}}(i,j,A(j+1,h)\alpha(i,h)).\end{aligned}$$

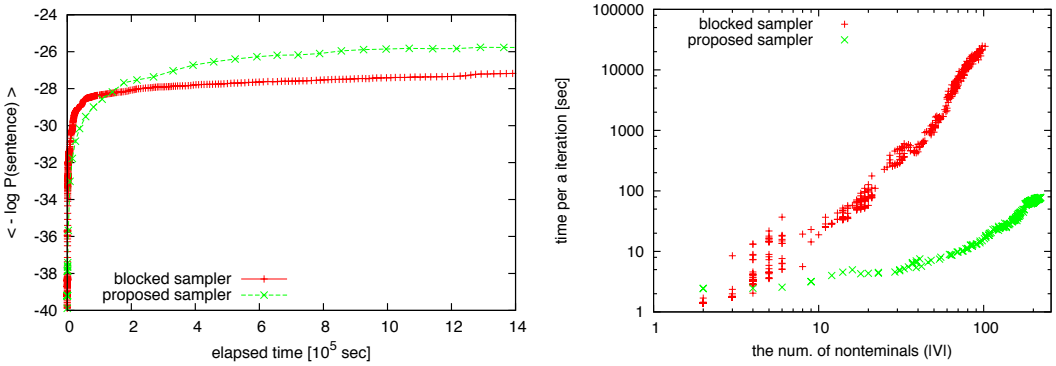
6. Experiments

Using experiments, we assess the following two points: a) the mixing speed and the practical computational cost of the proposed sampling method; and b) the effect of (k,l)-context-sensitive probabilities. We used sentences from the Brown Corpus (Francis and Kucera, 1982) with lengths of less than 16 as the input data for the experiments. Part-of-speech tags annotated in the corpus were assumed to be terminals. We let $(k,l) = (0,0)$ and $(1,0)$, i.e., implemented the algorithms for usual PCFGs and $(1,0)$-context-sensitive PCFGs.

As shown in Fig. 2(a), compared to the blocked sampler, the proposed sampler finally gives the better score than the blocked sampler, though it slowly mixes in the early stage. The slow mixing is considered to be the effect of the pointwise sampling for nonterminals in the proposed sampling method.

Fig. 2(b) shows the practical computational cost of both methods as a function of the number of nonterminals. The cost of the blocked sampling is extremely high for the large number of nonterminals, while that of the proposed method is small and the slope is relatively flat. Since the number of nonterminals increases as the time is elapsed, the blocked sampler becomes inefficient except for the early stage and thus does not give better results than the proposed sampler finally.

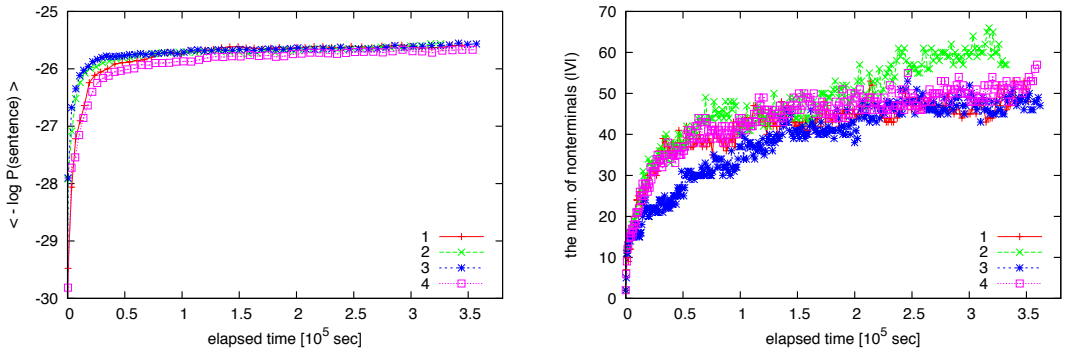
Fig. 3(a) shows the result of the proposed sampler for learning $(1,0)$-context-sensitive PCFGs. The mixing speed appears to be faster and the final score is better than it of PCFGs. The numbers of nonterminals are around 50 though they still appear to grow in the last stage (Fig. 3(b)).



(a) The average of $-\log P(\text{sentence})$ in the test data as a function of elapsed time.

(b) Time per a iteration as a function of the number of nonterminals $|V|$.

Figure 2: Comparing the proposed sampring with the blocked sampling. The number of sentences is 23988(train:21589 test:2399) and the number of terminals $|\Sigma|$ is 301. The CPU of the machine where the experiments are done is Core i7-3930K.



(a) The averages of $-\log P(\text{sentence})$ of 4 trials as functions of the elapsed time.

(b) The numbers of nonterminals for 4 trails as functions of the elapsed time.

Figure 3: Experimental results for $(1, 0)$-context-sensitive PCFGs with the proposed sampler.

Modified Kneser-Ney (Kneser and Ney, 1995) (MKN) is known to be a language model which give high perplexities. As Tab. 1 shows, the score of PCFGs learned by the proposed sampler is slightly worse than the best one of MKNs (4-gram). The score of learned $(1, 0)$-context-sensitive PCFGs is slightly better than those of MKNs.[3] In addition, the numbers of nonterminals for $(1, 0)$-context-sensitive PCFGs are smaller than those for PCFGs. This is a natural result because the existence of contexts allows multiple probabilities to be assigned to a nonterminal in $(1, 0)$-context-sensitive PCFGs. Since the smaller number of nonterminals gives the smaller computational cost for sampling, learning CFGs with $(1, 0)$-context-sensitive probabilities has the effect of reducing the actual computational cost compared to learning PCFGs.

3. As Fig. 2(a) and Fig. 3(a) shows, the scores of learned PCFGs and $(1, 0)$-context-sensitive PCFGs may grow more if we take sampling iterations more.

Table 1: Comparing proposed methods with a baseline algorithm (modified Kneser-Ney). "Score" represents the average of log P(sentence) in the test data. The average of 4 trials is taken in the row of "$(1,0)$-context". Type-0 and type-2 nonterminals represent those which have a terminal and two nonterminals in the left-hand side of rules, respectively.

Method name	Score	The num. of type-0 nonterminals	The num. of type-2 nonterminals
$(0,0)$-context(blocked sampler)	27.043	60	37
$(0,0)$-context(proposed sampler)	25.775	122	94
$(1,0)$-context(proposed sampler)	**25.596**	46.0	6.75
modified Kneser-Ney(unigram)	39.407	-	-
modified Kneser-Ney(bigram)	27.067	-	-
modified Kneser-Ney(trigram)	25.802	-	-
modified Kneser-Ney(4-gram)	**25.675**	-	-
modified Kneser-Ney(5-gram)	25.823	-	-
modified Kneser-Ney(6-gram)	25.902	-	-

7. Discussion

We show that taking a value of k that is not zero gives good results in terms of both prediction accuracy and computational cost in the experiments. On the other hand, as shown in Sec. 5, the computational cost of the blocked sampler and the proposed sampler are proportional to $|V|^{l+3}$ and $|V|^{l+1}$, respectively. Taking a value of l that is not zero increases the computational cost largely. Since the computational cost become critical, the proposed sampler may be of advantage when $l \neq 0$.

Another concern is that, although the order of constructing the hierarchies of base measures is straightforward and it may be appropriate in general cases, it is likely that these hierarchies will fail to capture the appropriate patterns of contexts in many cases. Pickhardt et al. (2014) proposed the use of combinations of skipped n-grams for smoothing and showed that these combination actually yielded better result. For (k,l)-context-sensitive PCFGs, we should examine whether it might be possible to mix the orders of constructing the hierarchies of base measures.

Acknowledgments

I am grateful to Ryo Yoshinaka for supporting my understanding of the theories and algorithms of distributional learning. This work was supported by JSPS KAKENHI Grant Number 26780123.

References

Phil Blunsom and Trevor Cohn. A hierarchical Pitman-Yor process HMM for unsupervised part of speech induction. *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 865–874, 2011.

Alexander Clark. Combining distributional and morphological information for part of speech induction. *In Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, 1:59–66, 2003.

Alexander Clark. PAC-learning unambiguous NTS languages. *In Proceedings of the 8th International Colloquium on Grammatical Inference (LNAI)*, 4201:59–71, 2006.

Trevor Cohn, Phil Blunsom, and Sharon Goldwater. Inducing tree-substitution grammars. *The Journal of Machine Learning Research*, 11:3053–3096, 2010.

Winthrop Nelson Francis and Henry Kucera. *Frequency analysis of English usage*. Houghton Mifflin Company, 1982.

Hemant Ishwaran and Lancelot F. James. Generalized weighted Chinese restaurant processes for species sampling mixture models. *Statistica Sinica*, 13:1211–1235, 2003.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. *Advances in neural information processing systems*, 19:641–648, 2007a.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. Bayesian inference for PCFGs via Markov chain Monte Carlo. *In HLT-NAACL*, pages 139–146, 2007b.

Dan Klein and Cristopher D. Manning. A generative constituent-context model for improved grammar induction. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 128–135, 2002.

Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. *In proceeding of International Conference on Acoustics, Speech, and Signal Processing, IEEE*, 1:181–184, 1995.

Karim Lari and Steve J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer speech and language*, 4(1):35–56, 1990.

Percy Liang, Slav Petrov, Michael I. Jordan, and Dan Klein. The infinite PCFG using hierarchical Dirichlet processes. *In EMNLP-CoNLL*, pages 688–697, 2007.

Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. *In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 1:100–108, 2009.

Rene Pickhardt, Thomas Gottron, Martin Körner, Paul Georg Wagner, Till Speicher, and Steffen Staab. A generalized language model as the combination of skipped n-grams and modified Kneser-Ney smoothing. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014.

Jim Pitman and Marc Yor. The two-parameter Poisson-Dirchlet distribution derived from a stable subordinator. *The Annals of Probablity*, 25:855–900, 1997.

Chihiro Shibata and Ryo Yoshinaka. PAC learning of some subclasses of context-free grammars with basic distributional properties from positive data. *In proceedings of the 24th International Conference on Algorithmic Learning Theory, LNAI*, 8139:143–157, 2013a.

Chihiro Shibata and Ryo Yoshinaka. A comparison of collapsed Bayesian methods for probabilistic finite automata. *Machine Learning (DOI:10.1007/s10994-013-5410-3)*, 2013b.

Hiroyuki Shindo, Yusuke Miyao, Akinori Fujino, and Masaaki Nagata. Bayesian symbol refined tree substitution grammars for syntactic parsing. *In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 440–448, 2012.

Hiromi Shirakawa and Takashi Yokomori. Polynomial-time MAT learning of c-deterministic context-free grammars. *IPSJ Journal*, 34(3):380–390, 1993.

Yee Whye Teh. A Bayesian interpretation of interpolated kneser-ney. *NUS School of Computing Technical Report TRA 2/06*, 2006.

Menno van Zaanen. ABL:alignment-based learning. *In Proceedings of the 18th International Conference on Computational Linguistics*, 2:961–967, 2000.

Ryo Yoshinaka. Towards dual approaches for learning context-free grammars based on syntactic concept lattices. *In proceedings of the 15th International Conference on Developments in Language Theory, LNCS*, 6795:429–440, 2011.

Ryo Yoshinaka. Integration of the dual approaches in the distributional learning of context-free grammars. *In Proceedings of the 6th International Conference on Language and Automata Theory and Applications, LNCS*, 7183:538–550, 2012.