Stochastic Finite Learning of Some Mildly Context-Sensitive Languages

John Case[1], Ryo Yoshinaka[2], and Thomas Zeugmann[3]

[1] Department of Computer and Information Sciences, University of Delaware
case@mail.eecis.udel.edu
[2] Graduate School of Informatics, Kyoto University
ry@i.kyoto-u.ac.jp
[3] Division of Computer Science, Hokkaido University
thomas@ist.hokudai.ac.jp

Abstract. In recent years different classes of mildly context-sensitive languages, which can handle complex structures beyond context-freeness, have been discovered that can be identified in the limit from positive examples efficiently in some sense. The goal of this paper is to convert those existing learning algorithms into ones that work under a more practical learning scheme, finite stochastic learning and to discuss the efficiency of the proposed algorithms.

1 Introduction

Identification in the limit (IIL) *from text* [8] is a classical paradigm that models language learning under an environment, where only positive examples are available. In fact in many real situations like when a child learns his/her first language, negative examples are hardly available. After Angluin [1] showed that pattern languages can be learned in this model, the literature has been paying much attention to IIL from text and obtained many fruitful results. However, this classical model is sometimes criticized for its inadequacy in practice. One cannot know when the hypothesis output by the learner has converged. That is, one cannot trust in the learner's hypothesis regardless of the quantity of information fed to the learner. Moreover, it is hard to define the efficiency of a learning algorithm due to the "in the limit" nature [4, 12]. We have no established way to analyze the efficiency of an IIL algorithm.

Reischuk and Zeugmann [13] and Rossmanith and Zeugmann [14] have proposed an alternative learning scheme, called *stochastic finite (*SF*) learning*, to overcome those difficulties of IIL. In their scheme, positive examples are sequentially given to a learner according to some probability distribution. Differently from the IIL model, an SF learner must eventually terminate and output a representation. One can evaluate a learner's efficiency by, say the expected number of examples that the learner needs to terminate. The literature has achieved positive results on the SF learning [13, 14, 3, 18], yet so far the learning target languages were limited to (variants of) pattern languages. This paper is concerned with the SF learning of the so-called *mildly context-sensitive languages.*

A long-term goal of grammatical inference is to find a reasonable class of formal languages that are powerful enough for expressing natural languages and are efficiently learnable under a reasonable scheme. Context-free languages are fairly expressive, yet natural languages are known to involve non-context-free structures like *multiple agreements*, *crossed agreements*, and *duplication structures* [9, 7, 15]. Joshi [10] proposed the notion of *mildly context-sensitive languages* in order to define a class of formal languages for modeling natural languages. They, on the one hand, should be rich enough to model the natural language phenomena mentioned above. On the other hand, its generative power should be still simple enough to allow for polynomial time parsing algorithms.

In recent years, several interesting IIL algorithms for different mildly context-sensitive languages have been proposed. Among these algorithms are the one discovered by Yoshinaka [16] that learns the class of *qD-substitutable multiple context-free languages*, and Becerra-Bonache *et al.*'s [2] algorithms targeting *simple external contextual languages*.

In the present paper we shall focus on these algorithms and translate those into SF learners. Section 3 shows how polynomial-time SF learning of qD-substitutable multiple context-free languages is possible under certain assumptions. We also prove that a subclass of simple external contextual languages is SF learnable in Section 4. In particular, it is shown that the class in concern is efficiently SF learnable under a certain condition — whether or not the condition holds true in general is left as an open problem.

2 Preliminaries

For a finite alphabet Σ, let Σ^* denote the set of strings over Σ. The empty string is denoted by λ and $\Sigma^+ = \Sigma^* \setminus \{\lambda\}$. For $u \in \Sigma^*$, let $|u|$ denotes the *length* of u. Any subset $L \subseteq \Sigma^*$ is called a *language*. For any set S, we use $|S|$ to denote its cardinality and $\wp(S)$ to denote its power set. The empty set is denoted by $\emptyset$. If L is a finite language over Σ, its size is defined as $\|L\| = |L| + \sum_{u \in L} |u|$.

We assume a countably infinite set Z of variables which is disjoint from Σ. We use $z_1, z_2, \ldots,$ to denote variables. A *pattern* π is an element of $(Z \cup \Sigma)^*$. A *substitution* is a homomorphism on $(Z \cup \Sigma)^*$ that maps every letter in Σ to itself. A homomorphism that maps z_i to u_i for $i = 1, \ldots, m$ is denoted as a suffix operator $[z_1 := u_1, \ldots, z_m := u_m]$. When $\boldsymbol{z}$ and $\boldsymbol{u}$ represent sequences of variables $z_1, \ldots, z_m$ and strings $u_1, \ldots, u_m$, respectively, the substitution is often denoted as $[\boldsymbol{z} := \boldsymbol{u}]$.

Definition 1. A *grammar system* on Σ is a pair $\langle \mathbb{G}, L \rangle$ where $\mathbb{G}$ is a class of representations called *grammars* and L is a function from $\mathbb{G}$ to $\wp(\Sigma^*)$ such that it is decidable whether $w \in L(G)$ for any $G \in \mathbb{G}$ and $w \in \Sigma^*$. We assume that every grammar $G \in \mathbb{G}$ is in some way assigned a positive integer called the *size* of G, denoted by $\|G\|$.

A *probabilistic grammar system* on Σ is a pair $\langle \mathbb{H}, Pr \rangle$ where $\mathbb{H}$ is a class of representations and Pr is a function from $\mathbb{H} \times (\Sigma^* \cup \{\#\})$ to $[0, 1]$ such that for every $H \in \mathbb{H}$, we have

$$\sum_{w \in \Sigma^* \cup \{\#\}} Pr(H, w) = 1\,,$$

where # is a special symbol not in Σ. We call each element of $\mathbb{H}$ a *probabilistic grammar*. The function Pr_H such that $Pr_H(w) = Pr(H, w)$ is called the *probabilistic language* of H. We define the *(non-probabilistic) language* of H by

$$\mathrm{L}_{Pr}(H) = \{\, w \in \Sigma^* \mid Pr_H(w) > 0 \,\}\,.$$

We often describe a probabilistic grammar system $\mathbb{H}$ based on a grammar system $\mathbb{G}$, where each H is defined on G such that $\mathrm{L}_{Pr}(H) = L(G)$. Such G is called a *underlying grammar* of H and the size of each H is defined to be that of G.

Definition 2. Let $\langle \mathbb{H}, Pr \rangle$ be a probabilistic grammar system and $\langle \mathbb{G}, L \rangle$ be its underlying grammar system. A *probabilistic text* of $H \in \mathbb{H}$ is an infinite sequence of elements of $\mathrm{L}_{Pr}(H) \cup \{\#\}$ independently drawn with respect to the distribution Pr_H. A *stochastic finite learner* (*learner*, in short) $\mathcal{A}$ is an algorithm that takes strings from a probabilistic text of H one by one. Each time it gets a string, it may either request a next string or terminate outputting a grammar $G \in \mathbb{G}$. We say that a learner *stochastically finitely (*SF*) learns* $\langle \mathbb{H}, Pr \rangle$ if for any $H \in \mathbb{H}$, it terminates in a finite number of steps and the output G satisfies that $L(G) = \mathrm{L}_{Pr}(H)$ with probability at least $1 - \delta$, where $\delta \in (0, 1)$ is the so-called *confidence parameter.*

Moreover, we say that a learner *polynomially stochastically finitely* (PSF) *learns* $\langle \mathbb{H}, Pr \rangle$ if the expected number of examples that it requests is polynomially bounded in $\|H\|\delta^{-1}$ where H is the learning target and the total running time is polynomially bounded in the total size of the given data.

We note that the above definition of SF learning differs from the original [13, 14] slightly. Our definition of SF learning targets a probabilistic grammar system, where we assume that a distribution of strings is in some way determined by part of a probabilistic grammar H. The original definition [13, 14] does not involve probabilistic grammars, but still they assume that a probabilistic text obeys an *admissible distribution* which may be defined in terms of underlying grammars.

When Pr is understood from the context, we let L denote L_{Pr} and simply say that $\mathcal{A}$ (P)SF-learns $\mathbb{H}$ by dropping Pr.

Example 1. A pattern can be seen as a grammar, whose language is defined to be the set of strings of letters that are obtained by substituting arbitrary nonempty strings for variables. That is, $L(\pi) = \{\, \pi\theta \mid \theta \text{ is a } \lambda\text{-free substitution} \,\}$. For example, $\pi = az_1bz_2z_1c$ is a pattern whose language is $L(\pi) = \{\, aubvuc \mid u, v \in \Sigma^+ \,\}$. Let D be a distribution of strings such that $D(w) > 0$ for all $w \in \Sigma^*$ and $D(\Sigma^*) = 1$, which represents the probability of the substitution of a string for each variable that occurs in π. We extend the domain of D to tuples of strings by $D(\langle u_1, \dots, u_k \rangle) = \prod_{1 \le i \le k} D(u_i)$. Then $H = \langle \pi, D \rangle$ is a probabilistic grammar such that $Pr(H, w) = \sum_{\boldsymbol{u} \in \mathrm{S}_\pi(w)} D(\boldsymbol{u})$, where $\mathrm{S}_\pi(w) = \{\, \langle u_1, \dots, u_k \rangle \mid \pi[z_1 := u_1, \dots, z_k := u_k] = w \,\}$ with the variables $z_1, \dots, z_k$ in π.

3 Learning multidimensionally substitutable MCFLs

3.1 Probabilistic multiple context-free grammars

A *ranked alphabet* is a pair of an alphabet V and a map $r\colon \Sigma \to \mathbb{N}$, where $\mathbb{N} = \{0, 1, 2, \ldots\}$ denotes the set of all natural numbers. Let x be any string and $m \in \mathbb{N}$, then $x^{\langle m\rangle}$ denotes the m-tuple containing x precisely m many times, while x^m denotes the usual concatenation of x, e.g., $x^{\langle 3\rangle} = \langle x, x, x\rangle$ and $x^3 = xxx$. Hence $(\Sigma^*)^{\langle m\rangle}$ is the set of m-tuples of strings over Σ, which are called *m-words.* Similarly we define $(\cdot)^{\langle *\rangle}$ and $(\cdot)^{\langle +\rangle}$, where, for instance, $(\Sigma^*)^{\langle +\rangle}$ denotes the set of all m-words for all $m \geq 1$. For an m-word $\boldsymbol{u} = \langle u_1, \ldots, u_m\rangle$, we write $|\boldsymbol{u}|$ to denote its arity, i.e., $|\boldsymbol{u}| = m$ and we use $\|\boldsymbol{u}\|$ to denote its *size* $m + \sum_{1\leq i\leq m} |u_i|$.

A *multiple context-free grammar* (MCFG) is a tuple $G = \langle \Sigma, N_{dim}, R, S\rangle$, where Σ is the alphabet of *terminal symbols*, $N_{dim} = (N, dim)$ is the ranked alphabet of *nonterminal symbols*, R is the finite set of *rules* and $S \in N$ is the *start symbol.* We call the value $dim(A)$ the *dimension* of $A \in N$. Each rule in R has the form

$$A_0(\alpha_1, \ldots, \alpha_{r_0}) \leftarrow A_1(z_{1,1}, \ldots, z_{1,r_1}), \ldots, A_n(z_{n,1}, \ldots, z_{n,r_n})$$

where $A_i \in N$ and $r_i = dim(A_i)$ for $i = 0, 1, \ldots, n$, $z_{i,j} \in Z$ for $i = 1, \ldots, n$ and $j = 1, \ldots, r_i$ and $\alpha_k \in (\Sigma \cup \{\, z_{i,j} \mid 1 \leq i \leq n,\, 1 \leq j \leq r_i \,\})^*$ for each $k = 1, \ldots, r_0$ for some $n \in \mathbb{N}$. Furthermore, in this paper we require each $z_{i,j}$ with $i = 1, \ldots, n$ and $j = 1, \ldots, r_i$ to occur exactly once through $\alpha_1, \ldots, \alpha_{r_0}$. A rule of the above form is an *A_0-rule* and R_{A_0} denotes the set of A_0-rules. The *size* of the rule is defined to be $|\alpha_1| + \cdots + |\alpha_{r_0}| + n + 1$. The size of a grammar is defined to be the sum of the sizes of its rules.

We represent ordered rooted labeled trees as terms. An *A-derivation tree* for $A \in N$ is a tree whose nodes are labeled by rules. We inductively define them as follows:

- the tree consisting of just one node labeled by a rule of the form $A(v_1, \ldots, v_r) \leftarrow$ is an A-derivation tree;
- if t_i are A_i-derivation trees for all $i = 1, \ldots, n$ and a rule ρ has the form $A(\boldsymbol{\alpha}) \leftarrow A_1(\boldsymbol{z}_1), \ldots, A_n(\boldsymbol{z}_n)$, then $\rho(t_1, \ldots, t_n)$ is an A-derivation tree;
- nothing else is an A-derivation tree.

A *derivation tree* means an S-derivation tree.

The *yield* $\mathrm{Y}_G(t)$ of an A-derivation tree t is a $dim(A)$-tuple of strings inductively defined by the following: If $t = \rho(t_1, \ldots, t_n)$ and ρ is a rule of the form $A(\boldsymbol{\alpha}) \leftarrow A_1(\boldsymbol{z}_1), \ldots, A_n(\boldsymbol{z}_n)$, then

$$\mathrm{Y}_G(t) = \boldsymbol{\alpha}[\boldsymbol{z}_1 := \mathrm{Y}_G(t_1), \ldots, \boldsymbol{z}_n := \mathrm{Y}_G(t_n)]\,.$$

We set $\mathrm{L}_G(A) = \{\, \mathrm{Y}_G(t) \mid t \text{ is an } A\text{-derivation tree} \,\}$, and refer to it as the *language* derived from a nonterminal A. The *language* of G is $\mathrm{L}(G) = \mathrm{L}_G(S)$.

A *probabilistic* MCFG is a tuple $H = \langle \Sigma, N, R, S, \pi \rangle$, where $G_H = \langle \Sigma, N, R, S \rangle$ is an MCFG and $\pi \colon R \to (0,1]$ gives a *probability* to each rule such that

$$\sum_{\rho \in R_A} \pi(\rho) = 1 \ , \text{ for all } A \in N \ .$$

The MCFG G_H is called the *underlying grammar* of H. The *probability* of an A-derivation tree t is defined to be the product of the probabilities assigned to rules that label nodes of t. That is, if t has the form $t = \rho(t_1, \dots, t_n)$, then the probability of t is defined by $\hat{\pi}(t) = \pi(\rho) \prod_{1 \le i \le n} \hat{\pi}(t_i)$. The probability of a string $u \in \Sigma^*$ assigned by a probabilistic grammar H is defined by

$$\mathrm{Pr}_H(u) = \sum_{t \in \mathrm{T}_S(u)} \hat{\pi}(t) \ ,$$

where $\mathrm{T}_A(u)$ is the set of A-derivation trees t such that $\mathrm{Y}_{G_H}(t) = u$. Moreover, for a language $K \subseteq \Sigma^*$, we let $\mathrm{Pr}_H(K) = \sum_{u \in K} \mathrm{Pr}_H(u)$. Obviously $\mathrm{Pr}_H(u) > 0$ if and only if $u \in \mathrm{L}(G_H)$ for any $u \in \Sigma^*$. Note that we have $\pi(\rho) > 0$ for all $\rho \in R$ by definition.

Finally, we define $\mathrm{Pr}_H(\#) = 1 - \mathrm{Pr}_H(\Sigma^*)$. Since H subsumes G_H, we often substitute H for G like Y_H or $\mathrm{L}(H)$ instead of Y_{G_H} or $\mathrm{L}(G_H)$. The size $\|H\|$ of a probabilistic MCFG is defined to be the size of its underlying grammar.

3.2 IIL of multidimensionally substitutable MCFLs

Yoshinaka [16] has defined a subclass of MCFLs that is interesting from its point of view of learnability. Let $\Box \notin \Sigma$ be a new symbol, which represents a *hole*. If $w \in (\Sigma \cup \{\Box\})^*$ contains m occurrences of $\Box$, then w is called an *m-context*. For an m-context $w = u_0 \Box u_1 \Box \dots \Box u_m$ such that $u_0, \dots, u_m \in \Sigma^*$ and an m-word $\boldsymbol{v} = \langle v_1, \dots, v_m \rangle \in (\Sigma^*)^{\langle * \rangle}$, we define an operation $\odot$ by setting $w \odot \boldsymbol{v} = u_0 v_1 u_1 \dots v_m u_m$. Note that $w \odot \boldsymbol{v}$ is defined only when w contains exactly $|\boldsymbol{v}|$ occurrences of $\Box$.

For a positive integer q, a language L is said to be *qD-substitutable* if and only if

$$w_1 \odot \boldsymbol{v}_1,\ w_1 \odot \boldsymbol{v}_2,\ w_2 \odot \boldsymbol{v}_1 \in L \text{ implies } w_2 \odot \boldsymbol{v}_2 \in L$$

for any $w_1, w_2 \in \Sigma^*(\Box\Sigma^+)^{m-1}\Box\Sigma^*$, $\boldsymbol{v}_1, \boldsymbol{v}_2 \in (\Sigma^+)^{\langle m \rangle}$ and $m \le q$. For notational convenience, we write $\Sigma_\Box^{[m]}$ for $\Sigma^*(\Box\Sigma^+)^{m-1}\Box\Sigma^*$.

We denote by $\mathbb{G}(q,r)$ the collection of MCFGs G whose nonterminals are assigned a dimension at most q and whose rules have at most r nonterminals on the right-hand side. Let $\mathbb{SL}(q,r)$ denote the set of languages that are qD-substitutable and are generated by a grammar from $\mathbb{G}(q,r)$. The class $\mathbb{SL}(2,1)$ contains non-context-free languages like $\{\, a^n b c^n d e^n \mid n > 0 \,\}$ and $\{\, a^m b c^n d e^m f g^n \mid m, n > 0 \,\}$, which can be seen as models of multiple agreements and crossed agreements.

Yoshinaka's [16] learning algorithm $\mathcal{A}(q,r)$ constructs a grammar G_K from a finite set K of positive examples of the target language $L \in \mathbb{SL}(q,r)$ in the following manner. The set of nonterminals is defined as

$$V_K = \{\, \boldsymbol{v} \in (\Sigma^+)^{\langle m \rangle} \mid w \odot \boldsymbol{v} \in K \text{ for some } w \in \Sigma_\Box^{[m]} \text{ and } 1 \le m \le q \,\} \cup \{S\},$$

where $\dim(\boldsymbol{v}) = |\boldsymbol{v}|$. We shall write $[\![\boldsymbol{v}]\!]$ instead of $\boldsymbol{v}$ for clarifying that it means a nonterminal symbol (indexed with $\boldsymbol{v}$). The set R_K consists of rules of the following three types:

I. $[\![\boldsymbol{v}]\!](\boldsymbol{\alpha}) \leftarrow [\![\boldsymbol{v}_1]\!](\boldsymbol{z}_1), \dots, [\![\boldsymbol{v}_n]\!](\boldsymbol{z}_n)$
if $n \leq r$ and $\boldsymbol{v} = \boldsymbol{\alpha}[\boldsymbol{z}_1 := \boldsymbol{v}_1, \dots, \boldsymbol{z}_n := \boldsymbol{v}_n]$, where $[\![\boldsymbol{v}]\!], [\![\boldsymbol{v}_1]\!], \dots, [\![\boldsymbol{v}_n]\!] \in V_K - \{S\}$;

II. $[\![\boldsymbol{v}]\!](\boldsymbol{z}) \leftarrow [\![\boldsymbol{v}']\!](\boldsymbol{z})$ if there is a $w \in \Sigma_{\square}^{[m]}$ such that $w \odot \boldsymbol{v}, w \odot \boldsymbol{v}' \in K$;

III. $S(z) \leftarrow [\![\langle v \rangle]\!](z)$ if $v \in K$.

Since the set V_K is finite, the set R_K is also finite. The conjecture made by the algorithm $\mathcal{A}(q,r)$ is then the MCFG $\mathcal{G}_{q,r}(K) = \langle \Sigma, V_K, R_K, S \rangle \in \mathbb{G}(q,r)$.

It is shown in [16] that for any qD-substitutable language L_* and any $K \subseteq L_*$, we have $\mathrm{L}(\mathcal{G}_{q,r}(K)) \subseteq L_*$. Furthermore, there exists also a set K_{G_*} such that $L_* = \mathrm{L}(\mathcal{G}_{q,r}(K))$ for any $K \supseteq K_{G_*}$, where G_* is a grammar generating L_*.

Proposition 1 (Yoshinaka [16]). *One can compute $\mathcal{G}_{q,r}(K)$ from a finite language $K \subset \Sigma^*$ in polynomial time. Assume that $K \supseteq K_{G_*}$ for an* MCFG *$G_* \in \mathbb{G}(q,r)$. Then we have $\mathrm{L}(G_*) \subseteq \mathcal{G}_{q,r}(K_{G_*})$. In particular, if $\mathrm{L}(G_*)$ is qD-substitutable, then $\mathrm{L}(\mathcal{G}_{q,r}(K)) = \mathrm{L}(G_*)$.*

3.3 Polynomial stochastic finite learning of substitutable MCFLs

Based on Yoshinaka's [16] IIL algorithm, we shall investigate (P)SF-learning of qD-substitutable MCFGs. Our learner outputs the grammar $\mathcal{G}_{q,r}(K)$ from a given positive data set K when a sufficient number of examples is received. Let $G = \langle \Sigma, N, R, S, \pi \rangle$ be a probabilistic MCFG of our learning target. This paper gives a definition of K_G in a way different from the original one, but still Proposition 1 holds with our definition. For each rule $\rho \in R$, among derivation trees that have a node labeled with ρ, let t_ρ be the one with the largest probability. We then define $K_G = \{\, v_\rho \mid \rho \in R \,\}$ where $v_\rho = \mathrm{Y}_G(t_\rho)$. We estimate $\mathrm{Pr}_G(v_\rho)$ for each $\rho \in R$ in order to determine how many examples our learner should request.

For each nonterminal $A \in N$, the most probable A-derivation tree has at most $(r^{|V|} - 1)/(r-1)$ nodes where r is the maximum number of nonterminals that appear on the right-hand side of a rule. Obviously every subtree of t_ρ is the most probable A-derivation tree except the ones rooted by nodes on the path from the root to a node labeled by ρ, whose length is at most $|V|$. Every node on the path has at most $r-1$ siblings and the node labeled with ρ has at most r children, all of which are a root node of the most probable A-derivation tree for some $A \in N$. Therefore, we have

$$|t_\rho| \leq |V| - 1 + ((r-1)|V| + 1)(r^{|V|} - 1)/(r-1) \leq (|V|+1)r^{|V|}$$

and hence

$$\mathrm{Pr}_G(v_\rho) \geq \hat{\pi}(t_\rho) \geq p^{(|V|+1)r^{|V|}} \ ,$$

where p is the least probability assigned to a rule in R. Let $s = p^{(|V|+1)r^{|V|}}$. Let Z_ρ^m denote the event that one does not observe v_ρ among randomly drawn m

examples. We have $\mathrm{P}(Z_\rho^m) \leq (1-s)^m \leq \mathrm{e}^{-ms}$ and thus the probability that we miss v_ρ for some $\rho \in R$ can be bounded as follows

$$\mathrm{P}\Big(\bigcup_{\rho \in R} Z_\rho^m\Big) \leq \sum_{\rho \in R} \mathrm{P}(Z_\rho^m) \leq |R|\mathrm{e}^{-ms} \ ,$$

where e is Napier's constant. Thus if we draw more than $s^{-1} \ln |R| \delta^{-1}$ examples, with probability at least $1-\delta$ we obtain v_ρ for all $\rho \in R$.

This number of required examples is not polynomially bounded in the description size of the target grammar because of $s = p^{(|V|+1)r^{|V|}}$. That is, even the most probable derivation tree of a grammar appear exponentially rarely. This is closely related to the well-known problem in the efficient learning of CFGs and richer formalisms: even the simplest derivation trees of a grammar may be exponentially large in the size of the grammar itself. Here we assume the *μ-observability* saying that we have a reasonably enough chance μ to observe the most probable string v_ρ for each ρ: $\mathrm{Pr}_G(v_\rho) \geq \mu$. This can be seen as an analogous property to the μ_1-distinguishability and μ_2-reachability that Clark [5] has assumed to achieve a PAC-type learnability of a subclass of CFGs. The μ_1-distinguishability and μ_2-reachability entails that every nonterminal A has a string v_A derived by using A whose probability is at least $\mu_1\mu_2$. We assume the similar property for each rule. If a rule is hardly used, we have little chance to learn it. Moreover, this assumption entails that $|R|$ is also bounded by μ^{-1}. Hence, if we draw more than

$$\frac{1}{\mu} \ln \frac{1}{\mu\delta}$$

examples, our grammar constructed on those examples generates the target language exactly with probability at least $1-\delta$.

Theorem 1. *Let $\mathbb{G}(q, r, \mu)$ be the class of probabilistic* MCFG*s in $\mathbb{G}(q, r)$ whose languages are q*D*-substitutable and μ-observable. The class $\mathbb{G}(q, r, \mu)$ is* PSF*-learnable.*

We remark that the above discussion on the PSF-learning of qD-substitutable MCFGs apply to other formalisms for which substitutability is accordingly defined and shown to be learnable: like CFGs [6], simple context-free tree grammars [11] and abstract categorial grammars [17].

4 Stochastic finite learning of simple external contextual grammars with one context

Becerra-Bonache *et al.* [2] have discussed the learnability of subclasses of *simple external contextual grammars (*SECG*s)*. A (q, s)-SECG is a pair $G = (\boldsymbol{v}, C)$ where $|\boldsymbol{v}| = q$, $|C| \leq s$ and $C \subseteq (\Sigma^*\Box\Sigma^*)^{\langle q \rangle}$. We call $\boldsymbol{v}$ and elements of C the *base* and *contexts* of G, respectively. The language of G, which we call a (q, s)-SECL, is given by

$$\mathrm{L}(G) = \{\, \Box^q \odot w_1 \odot \cdots \odot w_n \odot \boldsymbol{v} \mid w_1, \ldots, w_n \in C \,\}$$

(note that $\odot$ is associative). An SECG can be seen as a special case of an MCFG. Becerra-Bonache *et al.* [2] give an iterative learning algorithm for $(q, 1)$-SECGs. This section translates their algorithm into an SF learner. Following them, for ease of notation, when we say that a language equals $\{u_0, u_1, \dots\}$[4], we assume that the u_i are listed in length increasing order, that is, $|u_0| < |u_1| < |u_2| < \dots$. For a $(q, 1)$-SEC language $\{u_0, u_1, \dots\}$, it would be a natural idea to define a probabilistic distribution of strings by $Pr(u_i) = p^i(1 - p)$ where $0 < p < 1$.

The following theorem is a key ingredient that establishes the result obtained by Becerra-Bonache *et al.* [2].

Theorem 2 (Becerra-Bonache *et al.* [2], Theorem 22). *Let $V = \{v_0, v_1, v_2, \dots\}$ and $W = \{w_0, w_1, w_2, \dots\}$ be $(q, 1)$-SECLs. If $v_0 = w_0$, $v_1 = w_1$ and there are m, n with $v_n = w_m \wedge |v_n| > (|v_1| + 2)^3$ then $V = W$.*

It is easily seen that there is a polynomial-time algorithm (q is fixed) that for input strings u_0, u_1, u with $|u_0| < |u_1| < |u|$ decides whether there is an $(q, 1)$-SECL $\{u_0, u_1, \dots, u, \dots,\}$, and moreover if it is the case it outputs a $(q, 1)$-SECG $\mathcal{H}_q(u_0, u_1, u)$ generating the language. If there is no such a grammar then $\mathcal{H}_q(u_0, u_1, u)$ is undefined.

Therefore what is important is to identify the two shortest strings in the learning target. Our SF learner assumes that the two shortest strings u_0, u_1 amongst examples are indeed the two shortest in the target language when both strings have been drawn at least n times where n is determined by the confidence parameter δ. When in addition it observes a string u such that $|u| > (|u_1| + 2)^3$, it outputs $\mathcal{H}_q(u_0, u_1, u)$ as its conjecture.

Suppose that the target language is $L_* = \{v_0, v_1, \dots,\}$. We bound the probability that the output of our learner is wrong, which happen only when u_0 is not the shortest or u_1 is not the second shortest in L_*. Let Z_j denote the event that v_j is drawn n times before none of v_i with $i < j$ is drawn. Note that $Pr(v_{j+1}) \leq Pr(v_j)$ for all $j \geq 0$. Hence $\Pr(Z_j) < (j + 1)^{-n}$. Thus, the probability that u_0 is not the shortest is bounded from above by

$$\bigcup_{j \geq 1} \Pr(Z_j) < \sum_{j \geq 1} \frac{1}{(j + 1)^n} < \frac{1}{2^n} + \int_2^\infty \frac{1}{x^n} dx < \frac{3}{2^n}$$

for $n \geq 2$. The probability that u_1 is not the second shortest is similarly bounded from above by $\frac{3}{2^n}$. Therefore, the failure probability is at most $\frac{3}{2^{n-1}}$, which should be bounded by $1 - \delta$. In other words, for $n \geq 1 + \log_2 \frac{3}{\delta}$, our learner's conjecture is exactly correct with probability at least $1 - \delta$.

Theorem 3. *The class of $(q, 1)$-SECGs is SF learnable.*

Note that this theorem holds true as long as $\Pr(v_i) \geq \Pr(v_j)$ whenever $i \leq j$.

Next we estimate the expected number of examples that the learner requests. As in the previous section, we assume that the probabilities (p and $1-p$) assigned

[4] We assume that every $(q, 1)$-SECG in this paper generates an infinite language; otherwise, its language is only a singleton.

to rules are bounded from below by a constant. The probability that we draw v_j for some $j \geq k$ is p^k. Taking into account that $|v_k| = |v_0| + k(|v_1| - |v_0|)$, for $k > ((|v_1| + 2)^3 - |v_0|)/(|v_1| - |v_0|)$, we have $|v_k| > (|v_1| + 2)^3$. Thus the number of expected times that we draw positive examples until we find a long enough one is at most p^{-k} where $k = \lceil ((|v_1| + 2)^3 - |v_0|)/(|v_1| - |v_0|) \rceil + 1$, which is not polynomially bounded by the size of the target grammar if we define the size of an SECG $G = \langle \boldsymbol{v}, C \rangle$ to be $\|C\| + \|\boldsymbol{v}\|$, which is equivalent to $|w_1|$ modulo a constant term. On the other hand, it is open whether one can strengthen Theorem 2 so that w_0, w_1, w_n with $n \geq 2$ suffices for ensuring the uniqueness. If one can answer the open problem in the affirmative, it is not hard to see that the class of $(q, 1)$-SECGs becomes PSF-learnable.

At least it is the case when $q = 1$, where $(q, 1)$-SECGs generate only linear context-free languages.

Lemma 1. *Let P_k be the equation $u^k v w^k = (u')^k v (w')^k$. $P_1 \wedge P_2$ implies $\forall k.P_k$.*

Corollary 1. *The class of $(1, 1)$-SECGs is PSF-learnable under the assumption that the probability of drawing each of the shortest three positive example is bounded from below.*

5 Discussion

While language classes targeted by finite stochastic learning have so far been limited to pattern languages in the literature, this paper has discussed how existing learning algorithms for mildly context-sensitive languages working under the identification in the limit paradigm can be translated into finite stochastic learners. The positive result on qD-substitutable substitutable MCFLs can easily be applied to the "substitutable" subclasses of other formalisms including context-free string/tree languages [6, 11] and abstract categorial grammars [17] as well.

On the other hand, while we have established a positive result on the SF learning of $(q, 1)$-SECGs, it remains open whether $(q, 1)$-SECLs are PSF learnable for $q \geq 2$. At last, we remark another open problem related to the learning algorithm for (q, r)-SECLs from Becerra-Bonache *et al.*'s paper [2, Theorem 8]. Translating their algorithm into one for SF learning does not seem easy, since their learnability proof does not constructively describe a set of positive examples on which their learner converges to a correct grammar for the learning target.

Acknowledgement

The authors would like to thank Chihiro Shibata for a useful suggestion on part of our technical discussions.

References

[1] D. Angluin. Finding patterns common to a set of strings. *J. of Comput. Syst. Sci.*, 21(1):46–62, 1980.

[2] L. Becerra-Bonache, J. Case, S. Jain, and F. Stephan. Iterative learning of simple external contextual languages. *Theoret. Comput. Sci.*, 411:2741–2756, 2010.

[3] J. Case, S. Jain, R. Reischuk, F. Stephan, and T. Zeugmann. Learning a subclass of regular patterns in polynomial time. *Theoret. Comput. Sci.*, 364:115–131, 2006. Special Issue for *ALT'03*.

[4] J. Case and T. Kötzing. Difficulties in forcing fairness of polynomial time inductive inference. In R. Gavaldà, G. Lugosi, T. Zeugmann, and S. Zilles, editors, *Algorithmic Learning Theory, 20th International Conference, ALT 2009, Porto, Portugal, October 2009, Proceedings*, volume 5809 of *Lecture Notes in Artificial Intelligence*, pages 263–277, Berlin/Heidelberg, 2009. Springer.

[5] A. Clark. PAC-learning unambiguous NTS languages. In *Grammatical Inference: Algorithms and Applications, 8th International Colloquium, ICGI 2006, Tokyo, Japan, September 20-22, 2006, Proceedings*, volume 4201 of *Lecture Notes in Artificial Intelligence*, pages 59–71, Berlin, 2006. Springer.

[6] A. Clark and R. Eyraud. Polynomial identification in the limit of substitutable context-free languages. *Journal of Machine Learning Research*, 8:1725–1745, 2007.

[7] C. Culy. The complexity of the vocabulary of Bambara. *Linguistics and Philosophy*, 8:345–351, 1985.

[8] E. M. Gold. Language identification in the limit. *Inform. Control*, 10(5):447–474, 1967.

[9] S. P. Joan Bresnan, Ronald M. Kaplan and A. Zaenen. Cross-serial dependencies in dutch. *Linguistic Inquiry*, 13:613–635, 1982.

[10] A. K. Joshi. Tree adjoining grammars: how much context-sensitivity is required to provide reasonable structural descriptions? In D. R. Dowty, L. Karttunen, and A. Zwicky, editors, *Natural Language Parsing*, pages 206–250. Cambridge University Press, Cambridge, MA, 1985.

[11] A. Kasprzik and R. Yoshinaka. Distributional learning of simple context-free tree grammars. In J. Kivinen, C. Szepesvári, E. Ukkonen, and T. Zeugmann, editors, *ALT*, volume 6925 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 2011.

[12] L. Pitt. Inductive inference, DFAs, and computational complexity. In *Analogical and Inductive Inference, Proceedings of the Second International Workshop (AII'89)*, volume 397 of *Lecture Notes in Artificial Intelligence*, pages 18–44. Springer-Verlag, Berlin, 1989.

[13] R. Reischuk and T. Zeugmann. An average-case optimal one-variable pattern language learner. *J. Comput. Syst. Sci.*, 60(2):302–335, 2000.

[14] P. Rossmanith and T. Zeugmann. Stochastic finite learning of the pattern languages. *Machine Learning*, 44(1/2):67–91, 2001.

[15] S. Shieber. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343, 1985.

[16] R. Yoshinaka. Learning multiple context-free languages with multidimensional substitutability from positive data. *Inform. Comput.*, 412:1821–1831, 2011.

[17] R. Yoshinaka and M. Kanazawa. Distributional learning of abstract categorial grammars. In S. Pogodalla and J.-P. Prost, editors, *LACL*, volume 6736 of *Lecture Notes in Computer Science*, pages 251–266. Springer, 2011.

[18] T. Zeugmann. From learning in the limit to stochastic finite learning. *Theoret. Comput. Sci.*, 364(1):77–97, 2006.