社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2005−MPS−55（11）
2005／6／28

単純決定性言語の学習におけるサンプル分布のある制限と学習可能性

但馬 康宏，小谷 善行

東京農工大学 大学院共生科学技術研究院 システム情報科学部門

あらまし　単純決定性言語は，それを生成する文法のサイズと生成規則の出現確率の最小値が与えられれば，ランダムサンプルと所属性質問から多項式時間学習可能であることが知られている．ここで，サンプルの分布は任意の分布である．しかしながら，生成規則の出現確率の最小値を与えることは，サンプル分布の学習者からの独立性を損なっている．本研究において，学習者にその最小値を与えることなく学習可能となるような，サンプル分布に対するある条件を示し，多項式時間学習アルゴリズムを示す．

A restricted sample distribution of simple deterministic languages and its learnability

Yasuhiro TAJIMA and Yoshiyuki KOTANI

Institute of Symbiotic Science and Technology,
Tokyo University of Agriculture and Technology

Abstract　In our previous work[5], it has been shown that simple deterministic languages are polynomial time learnable from random examples and membership queries, if the size of the target grammar and the minimum occurring probability of rules are given. Here, random examples are drawn along an arbitrary distribution. However, giving the minimum occurring probability inhibits independence of the distribution from the learner. In this paper, we consider a condition of the distribution and show the learnability without the minimum occurring probability.

1 Introduction

In our previous work[5], it has been shown that simple deterministic languages are polynomial time learnable from random examples and membership queries, if the size of the target grammar and the minimum occurring probability of rules are given. Here, a hypothesis is in a simple deterministic grammar and random examples are drawn along an arbitrary distribution. However, giving the minimum occurring probability inhibits independence of the distribution from the learner and the target language. In this paper, we consider a condition of the distribution and show the learnability without the minimum occurring probability. The condition is that if an occurring probability of a rule is not the minimum among the target grammar and denoted by d, then there exists a rule whose occurring probability is bigger than $d/2$ and less than d. With this condition, we can obtain the number of examples for the polynomial time learning of the target language in polynomial time via membership queries.

2 Preliminaries

A *context-free grammar* (CFG) is a 4-tuple $G = (N, \Sigma, P, S)$ where N is a finite set of *nonterminals*, Σ is a finite set of *terminals*, P is a finite set of *rewriting rules* (rules for short) and $S \in N$ is the *start symbol*. Let σ be the word whose length is 0, and $\emptyset$ be the empty set. If $G = (N, \Sigma, P, S)$ is σ-free and any rule in P is of the form $A \to a\beta$ then G is said to be in *Greibach normal form*, where $A \in N, a \in \Sigma$, $\beta \in N^*$ and $|\beta| \leq 2$.

Let $A \to a\beta$ be in P where $A \in N$, $a \in \Sigma$ and $\beta \in N^*$. Let γ and $\gamma' \in N^*$. Then

$\gamma A\gamma' \underset{G}{\Rightarrow} \gamma a\beta\gamma'$ denotes the *derivation* and $\overset{*}{\underset{G}{\Rightarrow}}$ denotes the reflexive and transitive closure of $\underset{G}{\Rightarrow}$. The *language* generated from γ by G is denoted by $L_G(\gamma) = \{w \in \Sigma^* \mid \gamma \overset{*}{\underset{G}{\Rightarrow}} w\}$. The language generated from the start symbol S by G is called the language generated by G, and it is denoted by $L(G) = L_G(S)$. A nonterminal $A \in N$ is said to be *reachable* if $S \overset{*}{\underset{G}{\Rightarrow}} wA\beta$ for some $w \in \Sigma^*$, $\beta \in N^*$, and a nonterminal $D \in N$ is said to be *live* if $L_G(D) \neq \emptyset$.

A CFG G is a *simple deterministic grammar* (SDG) iff there exists at most one rule which is of the form $A \to a\beta$ for every pair of $A \in N$ and $a \in \Sigma$ where $\beta \in \Sigma \cup N$ and $|\beta| \leq 2$, i. e. if $A \to a\beta$ is in P then $A \to a\gamma$ is not in P for any $\gamma \in N^*$ such that $\gamma \neq \beta$[3]. We note that there exists exactly one derivation for each $w \in L(G)$ in an SDG G. The language generated by an SDG is called a *simple deterministic language* (SDL for short). In addition, such a set P of rules is called *simple deterministic*. The set of symmetric differences between $L(G_1)$ and $L(G_2)$ is denoted by $L(G_1)\Delta L(G_2)$.

Throughout this paper, we denote a hypothesis by L_h and the target language by L_t. Let D be a probability distribution over Σ^* and let $Pr(w)$ be the probability for $w \in \Sigma^*$. The learning from randomly drawn examples is called a *PAC*[6] learning if a hypothesis L_h satisfies

$$Pr[P(L_h \Delta L_t) \leq \varepsilon] \geq 1 - \delta \qquad (1)$$

for an error parameter $0 < \varepsilon \leq 1$ and a confidence parameter $0 < \delta \leq 1$, where $P(L_h \Delta L_t)$ is the probability of difference between L_h and L_t, i.e. the total of the probability for every $w \in L_h \Delta L_t$ on the distribution D. Even though the learner can use either some queries or additional information, we call L_h a PAC hypothesis if L_h satisfies (1). An *example* consists an *example word* $w \in \Sigma^*$ and the teaching signal $\{0, 1\}$ according to $w \in L_t$ or not. For any other definitions about PAC learning, the reader refers to [4].

For an SDG $G = (N, \Sigma, P, S)$ and the distribution D, we can define the probability for every rule $A \to a\beta$ in P as follows:

$$Pr(A \to a\beta) = \sum_{w \in Z(A \to a\beta)} Pr(w)$$

where

$$Z(A \to a\beta) = \{w \in \Sigma^* \mid S_t \overset{*}{\Rightarrow} \alpha_1 A \alpha_2 \Rightarrow \alpha_1 a\beta\alpha_2 \overset{*}{\Rightarrow} w \ for\ some\ \alpha_1, \alpha_2 \in (N \cup \Sigma)^*\}.$$

That is to say, $Pr(A \to a\beta)$ is an occurring probability of $A \to a\beta$ when a sample word is given.

We call a class of languages is exact learnable via some additional settings (such as queries or a special set of examples) if there exists a learning algorithm which uses the additional settings and whose hypothesis G_h is equivalent to the target language L_t, i. e. $L(G_h) = L_t$.

A membership query replies with 1 or 0 according to $w \in L_t$ or $w \notin L_t$, respectively. Here, $w \in \Sigma^*$ is the input word asked by the learner.

3 The SDL learning algorithm

In this section, we introduce outline of our previous work. In [5], the following theorems are proved by showing the learning algorithm.

Theorem 1 (Tajima et al.[5] Theorem 6) SDLs are polynomial time exact learnable via membership queries and a set of representative sample. □

Here, a set of representative sample is defined as follows.

Definition 2 Let $G = (N, \Sigma, P, S)$ be an SDG such that every $A \in N$ is reachable and live. Let Q be a finite subset of $L(G)$. Then Q is a *representative sample* (RS) of G iff the following holds.

- For any $A \to a\beta$ in P, there exists a word $w \in Q$ such that $S \overset{*}{\Rightarrow} xA\gamma \Rightarrow xa\beta\gamma \overset{*}{\Rightarrow} w$ for some $x \in \Sigma^*$ and $\gamma \in N^*$. □

Definition 3 For an SDL L, a finite set $Q \subseteq L$ is an RS iff there exists an SDG $G = (N, \Sigma, P, S)$ such that $L(G) = L$ and Q is an RS of G. □

Our result in this paper is a reduction of conditions in the following theorem.

Theorem 4 (Tajima et al.[5] Theorem 20) There exists a polynomial time learning algorithm of SDLs such that

- the hypothesis is PAC,
- there exists an SDG $G_t = (N_t, \Sigma, P_t, S_t)$ such that $L(G_t) = L_t$ and every rule $A \to \beta$ in P_t has the occurring probability which is bigger than or equal to d, i.e. $Pr(A \to \beta) \geq d$,
- the learner knows the size of G_t and d, and

- the learner can ask membership queries and can obtain m random examples where

$$m > \frac{1}{d}\log(\frac{|P_t|}{\delta}).$$

□

The outline of the learning algorithm of Theorem 4 is as follows[5].

1. Take m examples (let Q be the set of sample words). Here, $m > \frac{1}{d}\log(\frac{|P_t|}{\delta})$.

2. Construct the CFG $G_C = (N_C, \Sigma, P_C, S_C)$ as follows.

   - The set of rules P_C is made from all possible skeletons by which all positive example word in m examples can be generated.
   - Then, all rules which lead conflicts on checking words W are deleted from P_C.

   In other words, G_C can generate all words whose derivations on G_t only consist of rules used in that of positive example words. Thus, $L(G_C) \supseteq L_t$ holds if the set of m example words contain an RS. If the learner constructs a set W of correct checking words then the hypothesis becomes correct. This CFG has the same characteristics as the hypothesis of Ishizaka's algorithm[2].

3. Construct an SDG for every rule in P_C, and let $\mathbf{G}$ be the set of such SDGs. We call $\mathbf{G}$ *base grammars*.

4. Find $L_{G_1}(A)\Delta L_{G_2}(A)$ for every $A \in N_C$ and every pair of $G_1 \in \mathbf{G}$ and $G_2 \in \mathbf{G}$.

5. If there exists a witness word $w \in L_{G_1}(A)\Delta L_{G_2}(A)$ then add all sub-words of w to W and go back to 2.

6. If there is no witness word and $\mathbf{G} \neq \emptyset$ then output any $G \in \mathbf{G}$ else the learning fails.

We call this learning algorithm Algorithm1. Here, the CFG G_C satisfies the following conditions.

- For every nonterminal $A \in N_t$ occurs in derivations of m example words, there exists $A_C \in N_C$ such that if $S_t \overset{*}{\underset{G_t}{\Rightarrow}} \alpha A \beta \overset{*}{\underset{G_t}{\Rightarrow}} \alpha w' \beta \overset{*}{\underset{G_t}{\Rightarrow}} w$ for $w \in Q$ and $w' \in \Sigma^+$ then $A_C \overset{*}{\underset{G_C}{\Rightarrow}} w'$.

We call that A_C corresponds to A.

From this property, for every derivation $S_t \overset{*}{\underset{G_t}{\Rightarrow}} a_1 A_1 \beta_1 \overset{*}{\underset{G_t}{\Rightarrow}} a_1 a_2 A_2 \beta_2 \overset{*}{\underset{G_t}{\Rightarrow}} \cdots \overset{*}{\underset{G_t}{\Rightarrow}} a_1 \cdots a_n = w$ where $w \in Q$, $a_i \in \Sigma$, $A_i \in N_t$ and $\beta_i \in N_t^*$ $(i = 1, \cdots, n)$, there exist $A_{Ci} \in N_C$ which corresponds to A_i $(i = 1, \cdots, n)$ such that $S_C \overset{*}{\underset{G_C}{\Rightarrow}} a_1 A_{C1} \beta_{C1} \overset{*}{\underset{G_C}{\Rightarrow}} a_1 a_2 A_{C2} \beta_{C2} \overset{*}{\underset{G_C}{\Rightarrow}} \cdots \overset{*}{\underset{G_C}{\Rightarrow}} a_1 \cdots a_n = w$ where $\beta_{Ci} \in N_C^*$.

In the theorem 4, d is partial information of the distribution D for the learner. Thus, a learning setting without knowing d is more desirable setting for the learning.

4 A setting without the minimum occurring probability

Suppose an SDG $G_t = (N_t, \Sigma, P_t, S_t)$ such that $L(G_t) = L_t$. We consider the following restrictions for the occurring probability of G_t. Here, let $d = \min\{Pr(A \to \beta) \mid A \to \beta \text{ in } P_t\}$.

- For a rule $A \to \beta$ in P_t, if the occurring probability $Pr(A \to \beta) > d$ then there exists at least one rule $B \to \gamma$ in P_t such that $Pr(A \to \beta) > Pr(B \to \gamma) > Pr(A \to \beta)/2$.

We call this restriction *continuous occurrence* of G_t, and such a distribution is called continuous occurrence distribution. Because of this restriction, the distribution D is not independent of G_t.

In Fig. 1, we show the SDL learning algorithm under a continuous occurrence distribution. Angluin[1] has shown that the sample complexity n_i is enough to check the hypothesis is PAC or not. That is

$$n_i \geq \frac{1}{\varepsilon}\left(\log(\frac{1}{\delta}) + (\log 2)(i+1)\right).$$

Now, we show the correctness of this algorithm.

Theorem 5 SDLs are polynomial time learnable under the continuous occurrence distribution via

- membership queries,
- random examples,
- ε, δ and $|P_t|$.

Here, the hypothesis is PAC.
Proof : Let d_0 be the minimum occurring probability in G_t. If d in the algorithm shown in Fig. 1 becomes less than d_0, the learning successes with the probability $\sqrt[|P_t|]{1-\delta} > 1-\delta$. Thus, we show

Algorithm2
INPUT : ε, δ, $|P_t|$;
OUTPUT : a hypothesis SDG G_h;
begin
    $d := 1$;
    $s := 0$;
    repeat
        $d := d/2$;
        run Algorithm1 with $\varepsilon, 1 - \sqrt[|P_t|]{1-\delta}, d, |P_t|$;
        (let the CFG constructed in the algorithm be $G_C = (N_C, \Sigma, P_C, S_C)$)
        (let the hypothesis SDG be $G_h = (N_h, \Sigma, P_h, S_h)$)
        $s_0 := s$;
        $s := |P_C|$;
        take n_i examples;
        (Here, $n_i \geq \frac{1}{\varepsilon}(\log(\frac{1}{\delta}) + (\log 2)(i+1))$)
        if (n_i examples are not conflict with $L(G_h)$) then
            output G_h and terminate;
    until ($s_0 \geq s$ and $s \neq 0$)
    output $G = (\emptyset, \Sigma, \emptyset, S)$;
    (the learning fails)
end.

Figure 1: The SDL learning algorithm under the continuous occurrence distribution

that d becomes less than d_0 with the probability at least $(\sqrt[|P_t|]{1-\delta})^{|P_t|-1}$.

Assume that $d > d_0$. The learner obtains a set of examples such that derivations of example words use all rules in G_t whose occurring probabilities are bigger than d with the probability $\sqrt[|P_t|]{1-\delta}$. If Algorithm1 fails with this set of examples, the learner repeats the loop with $d/2$. On the other hand, there exists at least one rule whose occurring probability d' satisfies that $d > d' > d/2$ from the assumption of continuous occurring distribution. Thus, at most $|P_t|-1$ times repetition is enough to make $d < d_0$, and such repetitions occurs with the probability $(\sqrt[|P_t|]{1-\delta})^{|P_t|-1}$.

Thus, this theorem holds. □

5 Conclusions

In this paper, we define a special distribution called a continuous occurring distribution. SDLs are polynomial time learnable via membership queries and random examples if the sample distribution is continuous occurring. In this setting, the learner does not have to obtain the minimum probability, but variety of distributions is restricted.

References

[1] D. Angluin. Learning regular languages from queries and counterexamples. *Inf. & Comp.*, 75:87–106, 1987.

[2] H. Ishizaka. Polynomial time learnability of simple deterministic languages. *Machine Learning* 5:151–164, 1990.

[3] A. J. Korenjak, J. E. Hopcroft. Simple deterministic languages. *Proc. IEEE 7th Annu. Symp. on Switching and Automata Theory* 36–46, 1966.

[4] B. K. Natarajan. *Machine Learning : A Theoretical Approach.* Morgan, Kaufmann Publishers, San Mateo, CA, 1991.

[5] Y. Tajima, E. Tomita, M. Wakatsuki and M. Terada. Polynomial time learning of simple deterministic languages via queries and a representative sample. *Theoretical Computer Science*, 329:203–221, 2004.

[6] L. G. Valiant. A theory of the learnable. *Comm. of the ACM* 27:1134–1142, 1984.