PAC-Learning Unambiguous NTS Languages

Alexander Clark

Department of Computer Science, Royal Holloway University of London,
Egham, Surrey, TW20 0EX

Abstract. Non-terminally separated (NTS) languages are a subclass of deterministic context free languages where there is a stable relationship between the substrings of the language and the non-terminals of the grammar. We show that when the distribution of samples is generated by a PCFG, based on the same grammar as the target language, the class of unambiguous NTS languages is PAC-learnable from positive data alone, with polynomial bounds on data and computation.

1 Introduction

A long term research goal in grammatical inference is to find a class of languages which includes the natural languages, and is efficiently learnable from positive data. One of the earliest approaches in grammatical inference, though envisioned as a discovery procedure for linguists, rather than a model of first language acquisition is the distributional learning approach of [Har54]. This approach can form the basis for efficient algorithms for large scale context free grammatical inference, [Cla06], but the precise theoretical justification is still unclear. Given that it is possible to construct acyclic deterministic finite state automata that are hard to learn from positive examples, it is important to identify precisely what the language theoretic properties that allow learning to proceed are.

In this paper, we take a significant step towards this goal: we combine [CT04a] and [CE05] to prove a PAC-learnability result in a partially distribution-free setting of a class of context-free grammars from positive examples, without membership queries, structural information or any other side information except for some parameters we use to stratify the class. [CT04a] argues that for many problems, including natural language, there is a natural distribution or set of distributions, and that therefore the requirement in the standard PAC-framework for distribution-free learnability is too strict. They therefore argue that an appropriate modification is to consider a suitable set of distributions modelled by a related family of probabilistic automata (since they study finite automata).

[CE05], on the other hand, which shows the learnability of a class of "substitutable languages" is incomplete in a number of respects. Firstly, though it demonstrates polynomial identification in the limit, this is not enough to guarantee efficient learnability in practice, and secondly the class of substitutable languages is very small. For example, the language $\{a^n b^n | n > 0\}$ is not a substitutable language.

In this paper we consider context free grammars; given that distribution free learning is too difficult, we assume the data is generated by a PCFG, so that

Y. Sakakibara et al. (Eds.): ICGI 2006, LNAI 4201, pp. 59–71, 2006.

the distribution will be reasonable helpful, but without trivialising the results through possible collusion. Under this circumstance, we can then use statistical properties of the distribution to determine whether two substrings are congruent. The whole class of CFGs is too ambitious a goal to strive for; in this paper we use the class of non-terminally separated (NTS) languages [BS85, Sen85].

We attempt to stratify the learnability of this by adding a number of parameters that affect the complexity of learning. We use separate parameters wherever possible to get maximum discrimination over this class of languages; as a result some of the bounds may appear complicated, but they could be radically simplified by combining bounds.

The relevance of this approach to natural language needs some explanation. Natural languages are close to being NTS. The origin of the name is from non-terminally separated, and indeed if we take a natural language such as English, then given two non-terminals such as noun phrase and verb phrase, the sets of strings that can be generated by each of these are almost disjoint. Of course, lexical ambiguity is a problem, since a word like *share* can be both a noun and a verb, but working with a suitably disambiguated representation, to a large extent, natural languages are NTS. The underlying distributions we use are those of PCFGs. This seems well motivated since the current state of the art in language modelling uses just this sort of model [Cha01, JC00]. Thus the approach we take here, starting from a linguist's idea, is well motivated both in terms of learnability, (the constraints on the distribution) and in terms of the class of languages we consider. There are a number of limitations to the work presented here which we will discuss in the conclusion.

2 Notation and Definitions

An *alphabet* Σ is a finite nonempty set of symbols called *letters*. A *string* w over Σ is a finite sequence $w = a_1 a_2 \dots a_n$ of letters. Let $|w|$ denote the length of w. In the following, letters will be indicated by $a, b, c, \dots$, strings by $u, v, \dots, z$, and the empty string by λ. Let Σ^* be the set of all strings, the free monoid generated by Σ. By a language we mean any subset $L \subseteq \Sigma^*$. u is a substring of v, written $u \sqsubseteq v$ if there are $l, r \in \Sigma^*$ such that $lur = v$. The set of all substrings of a language L is denoted

$$Sub(L) = \{u \in \Sigma^+ : \exists w \in L \text{ such that } u \sqsubseteq w\} \tag{1}$$

(notice that the empty word does not belong to $Sub(L)$). We will define the number of contiguous occurrences of a substring u in w by $|w|_u = \sum_{l,r \in \Sigma^* : lur = w} 1$, so for example $|abab|_{ab} = 2$.

2.1 Grammars

A grammar is a quadruple $G = \langle V, \Sigma, P, I \rangle$ where Σ is a finite alphabet of *terminal symbols*, V is a finite alphabet of *non-terminals*, P is a finite set of *production rules*, and $I \subseteq V$ is a set of start (initial, or sentence) symbols. If

$P \subseteq V \times (\Sigma \cup V)^+$ then the grammar is said to be context-free (CF), and we will write the productions as $N \to w$. We will write $uNv \Rightarrow uwv$ when $N \to w \in P$. $\overset{*}{\Rightarrow}$ is the reflexive and transitive closure of $\Rightarrow$. The language defined by G is $L(G) = \{w \in \Sigma^* | \exists S \in I \text{ s.t. } S \overset{*}{\Rightarrow} w\}$.

Given a set $L \subseteq \Sigma^*$ we define the syntactic congruence of L to be the relation $u \equiv_L v$ iff $\forall l, r \in \Sigma^*, lur \in L$ iff $lvr \in L$. This is an equivalence relation and indeed a monoid congruence since $u \equiv_L v$ implies $lur \equiv_L lvr$ for all $l, r \in \Sigma^*$.

2.2 NTS Languages

In this paper we are interested in the class of NTS languages.

Definition 1. *A grammar $G = \langle \Sigma, V, P, A \rangle$ is non-terminally separated (NTS) iff whenever $N \in V$ such that $N \overset{*}{\Rightarrow} \alpha\beta\gamma$ and $M \overset{*}{\Rightarrow} \beta$ then $N \overset{*}{\Rightarrow} \alpha M \gamma$.*

A language L is NTS if it can be described by an NTS grammar. Space does not permit a full exposition of the properties of this class but we note first that the class of NTS languages properly includes all regular languages, and that there are efficient polynomial algorithms for deciding the NTS property.

NTS languages are deterministic and thus not inherently ambiguous. We shall restrict ourselves here to unambiguous grammars – i.e. those grammars such that every string in the language has only one (rightmost) derivation. Surprisingly, this restriction does reduce the class of languages significantly, i.e. there are NTS languages which cannot be described by an unambiguous NTS grammar. Consider the language $L = \{a^n | n > 0\}$. This is a NTS language, and it is easy to see that the grammar must contain at least the productions $S \to a$ and $S \to SS$ (since $S \overset{*}{\Rightarrow} aa$ and $S \overset{*}{\Rightarrow} a$). Therefore, this is ambiguous since aaa will have at least two rightmost derivations. $S \Rightarrow SS \Rightarrow Sa \Rightarrow SSa \Rightarrow Saa \Rightarrow aaa$ and $S \Rightarrow SS \Rightarrow SSS \Rightarrow SSa \Rightarrow Saa \Rightarrow aaa$.

We will also make some other assumptions about the form of the grammar, that do not affect the class of languages defined. In particular we assume that there are no redundant non-terminals in the grammar, i.e. that for all $N \in V \exists u \in \Sigma^* \, N \overset{*}{\Rightarrow} u$ and $\exists S \in I, l, r \in \Sigma^* \, S \overset{*}{\Rightarrow} lNr$. We will also assume that there are no duplicate non-terminals – i.e. no non-terminals that generate the same strings.

2.3 Distributions

A distribution D over Σ^* is a function $P_D : \Sigma^* \to [0, 1]$ such that $\sum_{u \in \Sigma^*} P_D(u) = 1$. We will write $supp(D) = \{w \in \Sigma^* | P_D(w) > 0\}$. For a language $L \subseteq \Sigma^*$ we will write $P_D(L) = \sum_{w \in L} P_D(w)$.

The L_∞ norm of a function F over a countable set X is defined as

$$L_\infty(F) = \max_{x \in X} |F(x)|$$

Note that this defines a metric ($L_\infty(F_1 - F_2) = 0$ implies $F_1 = F_2$), and it satisfies the triangle inequality.

We define $E_D[u] = \sum_{l,r \in \Sigma^*} P_D(lur)$, the expected number of times the substring will occur (not the probability since it can be greater than 1). We define the probability that we observe one or more us to be $O_D(u) = P_D(\{w \in \Sigma^* : u \sqsubseteq w\})$

A context distribution C is a function from $\Sigma^* \times \Sigma^* \to [0,1]$ such that $\sum_{l \in \Sigma^*} \sum_{r \in \Sigma^*} C(l,r) = 1$.

The context distribution of a string u, where $u \in Sub(L)$, is written as C_u^D and is defined as

$$C_u^D(l,r) = \frac{P_D(lur)}{E_D(u)} \tag{2}$$

We will normally suppress the distribution when it is unambiguous. Given a multiset M of elements of $\Sigma^* \times \Sigma^*$, we will write $\hat{M}$ for the empirical distribution.

We define a notion of probabilistic congruence analogous to that of syntactic congruence.

Definition 2. *Given two strings $u, v \in \Sigma^*$ and a distribution D over Σ^*, u and v are probabilistically congruent with respect to a distribution D, written $u \cong_D v$ if and only if $C_u^D = C_v^D$*

2.4 PCFGs

We will concern ourselves with distributions generated by probabilistic context free grammars. A PCFG is a CFG $G = \langle \Sigma, V, P, I \rangle$ together with two functions, an initial symbol probability function, $\iota : I \to (0,1]$ and a production probability function $\pi : P \to (0,1]$ that satisfy the following constraints, $\sum_{S \in I} \iota(S) = 1$ and for all $N \in V$ $\sum_{N \to \alpha \in P} \pi(N \to \alpha) = 1$. For any rightmost derivation we can attach a probability which is the product of the $\pi(N \to \alpha)$ of all productions used in the derivation, and the probability of a string is then the product of the $\iota(S)$ for the start symbol used, and the probability of the derivation: $P_D(u) = \iota(S)P(S \stackrel{*}{\Rightarrow}_G u)$ if $S \stackrel{*}{\Rightarrow}_G u$. (We assume here that it is unambiguous and NTS). If a PCFG is such that $\sum_{w \in \Sigma^*} P(w) = 1$ then this is a consistent PCFG, and it defines a distribution, whose support is $L(G)$.

3 Learnability and Parameters

Given an unambiguous NTS grammar G defining a language $L(G)$, and assuming that D a distribution is defined by a PCFG based on the same grammar, we can establish the following result.

Lemma 1. *if $N \stackrel{*}{\Rightarrow}_G u$ and $N \stackrel{*}{\Rightarrow}_G v$ then $u \cong_D v$.*

Proof Since G is NTS, $u \equiv_L v$. Consider any $l, r \in \Sigma^*$ such that $lur \in L$. Let $p_u = P(N \stackrel{*}{\Rightarrow}_G u)$ and $p_v = P(N \stackrel{*}{\Rightarrow}_G v)$ For any l, r such that $S \stackrel{*}{\Rightarrow}_G lNr$, let $p_{l,r} = \iota(S)P(S \stackrel{*}{\Rightarrow}_G lNr)$. By PCFG assumptions, $P(S \stackrel{*}{\Rightarrow} lur) = p_{l,r}p_u$ and $P(S \stackrel{*}{\Rightarrow} lvr) = p_{l,r}p_v$; therefore $u \cong_D v$.

Given the well-known results on learning acyclic PDFAs [KMR$^+$94], it is necessary to add some criterion for distinguishability of states. We modify the definition of [RST98] to handle PCFGs as follows:

Definition 3. *A PCFG is μ_1-distinguishable iff for every non-terminal N there is a string u such that $P(N \stackrel{*}{\Rightarrow} u) > \mu_1$.*

Note that since NTS grammars are non-terminally separated, this is sufficient for them to be distinguishable in the sense of [RST98].

We also need to add some restrictions not on the distribution of strings generated by the nonterminals but also on the context distributions. Since we will be using context distributions to identify the relation of syntactic congruence, we will require the context distributions to be reasonably far apart in the L_∞ norm. Clearly, given that there will be an infinite number of congruence classes, we cannot require an absolute lower bound on the distance between context distributions. Even for regular languages, the number of congruence classes can be exponentially large, though finite. So we will have two further requirements on our distribution.

Definition 4. *A PCFG is ν-separable for some $\nu > 0$ if for every pair of strings u, v in $Sub(L(G))$ such that $u \not\equiv v$, it is the case that $L_\infty(C_u - C_v) \geq \nu \min(L_\infty(C_u), L_\infty(C_v))$*

Note that according to this definition, if a language is substitutable [CE05], then it is ν-separable with $\nu = 1$.

Additionally we require a certain degree of concentration in the contexts. It is easy to construct examples where the L_∞ norms of the context distributions are exponentially small, in which case we will need exponentially large amounts of data to be able to reliable determine when two strings are congruent or not using the separability property. Accordingly we add the following definition.

Definition 5. *A PCFG is μ_2-reachable, if for every non-terminal $N \in V$ there is a string u such that $N \stackrel{*}{\Rightarrow}_G u$ and $L_\infty(C_u) > \mu_2$.*

Clearly every PCFG is μ_2-reachable for some μ_2. This implies that there are strings l, r such that $P(lur) > \mu_2 P(N \stackrel{*}{\Rightarrow}_G u)$. Formulating the bound in terms of the norm of the context distribution is slightly stronger, since there might be more than one occurrence of u in lur. Alternatively we could combine this with distinguishability: if a PCFG is both μ_1-distinguishable and μ_2-reachable, then we know that for every non terminal N there are strings l, u, r such that $\iota(S)P(S \stackrel{*}{\Rightarrow}_G lNr \stackrel{*}{\Rightarrow}_G lur) > \mu_1\mu_2$, and so we could simply have a single bound corresponding to $\mu_1\mu_2$. While this is more compact, it is conceptually cleaner to separate the two bounds and treat them independently – this gives a more accurate representation of the functional dependence of the sample complexity on these parameters.

Intuitively we require that we do not have any strings that are very frequent but such that all of the strings that they occur in have exponentially small probability.

4 Algorithm

We now define the algorithm **PACCFG**. Our primary concern here is not with the algorithmic aspects of this, so we will present this using naive, but polynomial,

procedures. When implementing this, we would obviously use more efficient data structures. We will start by defining it informally.

We are given a sequence of positive strings $S = w_1, \ldots w_N$. First of all we collect all the frequent substrings, those strings that occur in more than a certain threshold m_0 of these strings. For each of these frequent strings, we collect a multiset of contexts, as follows: for a string u, we consider each data point w_i such that $u \sqsubseteq w_i$, calculate $|w_i|_u$, and then collect all of the contexts from this string. Thus if we have $u = a$ and $w_i = axayaz$, then we add these three contexts $(\lambda, xayaz), (ax, yaz), (axay, z)$. This procedure will give at least m_0 samples from the context distribution, but these will not in general be independent; nonetheless we can be sure that we will have a good estimate with high probability, i.e. that $L_\infty(\hat{C_u}, C_u)$ is small. We then form the set of all those frequent strings u that have $L_\infty(\hat{C_u}) > \mu_2/2$. We then construct a graph, where each node corresponds to one of these frequent substrings, and there is an arc between two distinct nodes, if and only if the two context distributions are similar. The test we use returns true if

$$L_\infty(\hat{C}(u) - \hat{C(v)}) \leq 2\mu_3 \tag{3}$$

We design this similarity test so that we can be sure, knowing the separability of the distribution, that it will pass only if these two substrings are probabilistically congruent and thus syntactically congruent. Given this graph, we then identify the components (i.e. the maximal connected subgraphs). All of the substrings in a given component will be syntactically congruent. We will write below $[u]$ for the component that contains the string u. We then construct a grammar from this, using the procedure in [CE05].

For every component we have a corresponding non-terminal. The set of initial symbols, will be the set of components that contain one of the sentential strings $w_1, \ldots w_n$. The set of productions is defined as follows. For every letter $a \in \Sigma$ that is in U, we add a production $[a] \to a$. For every string of length greater than one, $u \in U$, we add every production of the form $[u] = [v][w]$ where $u = vw, |v| > 0, |w| > 0$; there will be $|u| - 1$ such productions for every string. Note that if $u \in U$ and v is a substring of u then v must occur at least as many times as u and thus $v \in U$ as well.

More formally we define the algorithm in Algorithm 1.

Proposition 1. PACCFG *runs in time polynomial in the total length of strings in the input data.*

Proof (sketch) The number of substrings is polynomial, all of the computations can be performed using standard algorithms that are polynomial in the number of substrings.

4.1 Bounds

We now define the various bounds that are used in the algorithm and its analysis. We start by defining the various parameters. One of the problems with CFGs is

Algorithm 1. PACCFG algorithm

Data: A sequence of strings $W = w_1, w_2 \ldots, w_n$, parameters m_0, ν, μ_2, alphabet Σ
Result: A context free grammar $\hat{G}$
Find all substrings that occur at least m_0 times
$U_{=}\{u \in \Sigma^+ : |\{w_i | u \sqsubseteq w_i\}| \geq m_0\}$;
foreach $u \in U$ **do**
  $C_u = \{\}$ empty list ;
  foreach $w_i \in W$ **do**
    if $u \sqsubseteq w_i$ **then**
      foreach l, r *such that* $lur = w_i$ **do**
        Append (l, r) to C_u ;
      end
    end
  end
end
$U_c = \{u \in U | L_\infty(\hat{C_u}) > \mu_2/2\}$;
$E = \{(u_i, u_j) \in U_c : L_\infty(\hat{C}(u_i) - \hat{C}(u_j)) < 2\mu_3\}.$;
Construct a graph $SG = (U, E)$;
Compute $\hat{V} = \{V_1, \ldots\}$ be the set of components of the graph SG;
Compute the set of productions
$\hat{P} = \{[a] \to a | a \in \Sigma\} \cup \{[w] \to [u][v] | w \in U, w = uv\}$;
Select the initial symbols : $\hat{I} = \{N \in \hat{V} | w_i \in \hat{N}\}.$;
output $\hat{G} = \langle \Sigma, \hat{V}, \hat{P}, \hat{I} \rangle$;

that the strings can be exponentially large, even if we observe a low expected length in the sample, the true expectation could still be exponentially large because there might be a very rare nonterminal that generates very long strings. We need to have a loose bound on the expected number of substrings, so that we can bound the number of possible context distributions we need to estimate: this only appears in a logarithmic bound so it is not very significant. Thus we require the following upper bound L where

$$\sum_{w \in \Sigma^*} \frac{1}{2} |w| \left(|w| + 1\right) P_D(w) \leq L \tag{4}$$

We have precision and confidence parameters, ϵ and δ, alphabet size $|\Sigma|$, an upper bound on the number of non-terminals of the grammar, n, and on the number of productions p. We will assume an upper bound on the length of the right hand sides of the productions of l. We also require that the distribution is μ_1-distinguishable, μ_2-reachable and ν-separable. Given these constraints we define the following quantities.

$$\epsilon_2 = \frac{\epsilon}{p + n} \tag{5}$$

$$\mu_3 = \frac{\nu \mu_2}{16} \tag{6}$$

$$M = \frac{2}{\mu_1^l \epsilon_2} \tag{7}$$

$$\delta_1 = \delta/4 \tag{8}$$

$$\delta_2 = \frac{\delta_1^2}{LM} \tag{9}$$

$$\tag{10}$$

The threshold for the counts of substrings is m_0:

$$m_0 = \max\left(\frac{1}{2\mu_3^2}\log\frac{8}{\mu_3\delta_2}, \frac{1}{\mu_3}\log\frac{128}{\delta_2\mu_3}, 4\log\frac{p}{\delta_1}\right) \tag{11}$$

The total number of strings we require, the sample complexity is N:

$$N = m_0 M \tag{12}$$

Given these quantities, we will now state and prove a series of propositions showing that with high probability, the samples we draw will have the right properties. We assume that the data is being generated by a PCFG with the properties discussed above. We draw N strings $w_1, \ldots, w_N$.

Proposition 2. *With probability greater than* $1 - \delta_1$,

$$\sum_{i=1}^{N} \frac{1}{2}|w_i|(|w_i|+1) \leq NL\delta_1^{-1} \tag{13}$$

Proof By the definition of L the expectation of the left hand side is less than NL. Using the Markov inequality establishes the result.

Therefore we can see that the total number of strings with counts above m_0 will be at most $NL\delta_1^{-1}m_0^{-1} = ML\delta_1^{-1}$. We will divide the productions into two sets, a set of frequent productions, that we expect to observe a significant number of, and a set of infrequent productions, that will constitute a source of errors. For every production $N \to \alpha$ in the set of productions P, we define the set of strings that use that production

$$W(N \to \alpha) = \{w \in \Sigma^* : \exists S \in I, \exists \beta, \gamma \in (V \cup \Sigma)^* \text{ s.t. } S \stackrel{*}{\Rightarrow}_G \beta N \gamma \Rightarrow \beta\alpha\gamma \stackrel{*}{\Rightarrow} w\}$$

A production is ϵ_2-frequent if $P_D(W(N \to \alpha)) > \epsilon_2$.

Proposition 3. *For every* ϵ_2*-frequent production* $N \to \alpha$ *in* P*, with probability at least* $1 - \delta_1$*, there will be a string* u *such that* $\alpha \stackrel{*}{\Rightarrow} u$*, and that* u *occurs in at least* m_0 *strings.*

Proof There must be a string u such that $\alpha \stackrel{*}{\Rightarrow} u$ and $P(\alpha \stackrel{*}{\Rightarrow} u) > \mu_1^l$, by the distinguishability, and the fact that $|\alpha| \leq l$. Therefore $O_D(u) \geq \epsilon_2\mu_1^l$. We have that $N > 2m_0\epsilon_2^{-1}\mu_1^{-l}$ therefore given N samples we would expect for any given production, using Chernoff bounds, the probability of seeing less than m_0

occurrences to be less than $e^{-N\epsilon_2\mu_1^l/8}$. Since there are at most p productions we will require

$$e^{-N\epsilon_2/8} < e^{-m_0/4} < \frac{\delta_1}{p} \tag{14}$$

which is satisfied by $m_0 > 4 \log \frac{p}{\delta_1}$.

For every initial symbol $S \in I$, we define $I(S) = \{w : S \stackrel{*}{\Rightarrow} w\}$. An initial symbol is ϵ_2-frequent iff $P_D(I(S)) > \epsilon_2$.

Proposition 4. *For every ϵ_2-frequent initial symbol S, with probability at least $1 - \delta_1$ there will be a string u such that u occurs at least m_0 times in the sample and $S \stackrel{*}{\Rightarrow} u$.*

Proof Since it is ϵ_2-frequent we know that $\iota(S) > \epsilon_2$. Since the PCFG is μ_1-distinguishable we know that there must be a string u such that $P(S \stackrel{*}{\Rightarrow}_G u) > \mu_1$, therefore there must be a string with $P_D(u) > \mu_1\epsilon_2$. Since $N > 2m_0\mu_1^{-1}\epsilon_2^{-1}$, using Chernoff bounds we expect for a given symbol S, the probability of not seeing this string to be less than $e^{-N\mu_1\epsilon_2/8}$. Since there are at most n initial symbols, we will require

$$e^{-N\epsilon_2\mu_1/8} < \frac{\delta_1}{n} \tag{15}$$

which is satisfied by $m_0 > 4 \log \frac{n}{\delta_1}$, which is a weaker bound.

Proposition 5. *Assuming that the bound above holds, with probability at least $1 - \delta_1$, for every substring u with count greater than m_0, $L_\infty(\hat{C}_u, C_u) < \mu_3$.*

Proof The following lemma can be proved, using techniques similar to those in [CT04b]. The important difference is that even though the strings may be drawn from the distribution independently, the draws from the context dstribution will not be independent, since the same substring may occur more than once in a single string, and thus there will be dependencies. Fortunately, the Bernouilli indicator variables associated with each context are negatively associated [DR98] and thus we can still apply Chernoff/Hoeffding bounds.

Lemma 2. *For any context distribution D, for any $\epsilon' > 0$ and any $\delta' > 0$, given N' samples drawn from D, which are negatively associated [DR98], where*

$$N' > \max\left(\frac{1}{2\epsilon'^2}\log\frac{8}{\epsilon'\delta'}, \frac{1}{\epsilon'}\log\frac{128}{\delta'\epsilon'}\right) \tag{16}$$

the empirical distribution $\hat{S}$ of the samples will satisfy $L_\infty(\hat{S} - D) < \epsilon'$, with probability at least $1 - \delta'$.

Using the assignments $\delta' = \delta_2$ and $\epsilon' = \mu_3$, establishes the result, given the bound on the number of substrings given above.

If all of these properties hold, then we say that the sample is m_0, μ_3-good.

We now come to the most important step; this lemma establishes that the comparison of the context distributions will give the right answer.

Proposition 6. *If the sample is m_0, μ_3-good, then whenever we have two strings u, v whose counts are at least m_0, and such that $L_\infty(\hat{C}_u) > \mu_2/2$ and $L_\infty(\hat{C}_v) > \mu_2/2$ then $L_\infty(\hat{C}_u, \hat{C}_v) < 2\mu_3$ if and only if $u \equiv_L v$.*

Proof Since $L_\infty(\hat{C}_v) > \mu_2/2$, and the sample is good, we know that $L_\infty(C_v) > \mu_2/4$. (and similarly for v). Suppose u and v are not congruent, then since the distribution is ν-separable, we know that $L_\infty(C_u, C_v) \geq \nu\mu_2/4 = 4\mu_3$. Since the sample is good, we know that $L_\infty(\hat{C}_u, C_u) < \mu_3$ and $L_\infty(\hat{C}_v, C_v) < \mu_3$. Therefore by the triangle inequality $L_\infty(\hat{C}_u, \hat{C}_v) > 2\mu_3$. Conversely if they are congruent, then $C_u = C_v$, and by the triangle inquality $L_\infty(\hat{C}_u, \hat{C}_v) < 2\mu_3$.

Proposition 7. *If the number of samples exceeds N then the sample is good with probability at least $1 - \delta$.*

Proof With probability at most δ_1 the strings are too long, with probability at most δ_1 some frequent production does not occur at least m_0 times. With probability at most δ_1 the sample is m_0, μ_3-good. Therefore the total probability this not being the case is less than δ. QED

5 Proof

Having established these proposition we can now prove the correctness of the algorithm. We now work under the assumption that the sample is good and show that in this situation the algorithm produces a hypothesis with small error.

First of all, we show that the hypothesized language will be a subset of the target language. Here the proof is very similar to [CE05]. For a string of terminal and non terminal symbols $\alpha \in (V \cup \Sigma)^+$, we can define $w(\alpha)$ to be the set of strings of Σ^* formed by replacing every element of $N \in V$ with one of the strings w in the component corresponding to N. So if $N_1 = \{u_1, u_2\}$ and $N_2 = \{v_1, v_2 v_3\}$ and $\alpha = aN_1bN_2$, then $w(\alpha) = \{au_1bv_1, au_1bv_2, au_1bv_3, au_2bv_1, au_2bv_2, au_2bv_2\}$. If $\alpha \in \Sigma^*$ then $w(\alpha) = \{\alpha\}$, and if $\alpha = N \in \hat{V}$ then $w(N)$ is precisely the set of substrings in that component of the substitution graph.

Lemma 3. *For every α, $u \in w(\alpha)$ and $v \in w(\alpha)$ implies that $u \equiv_L v$.*

Proof If u and v are in the same component, then they are congruent. Since syntactic congruence is a monoid congruence the result holds, by induction on the length of α.

Lemma 4. *For all $v \in \Sigma^*$, for all $\alpha \in (V \cup \Sigma)^*$, $\alpha \stackrel{*}{\Rightarrow}_{\hat{G}} \beta$ and $u \in w(\alpha), v \in w(\beta)$ implies $u \equiv_L v$*

Proof By induction on the length of the derivation $\alpha \stackrel{*}{\Rightarrow}_{\hat{G}} \beta$. Suppose we have a derivation of length 0, i.e. $\alpha = \beta$, then the previous lemma establishes the result. Otherwise suppose it is true for all derivations of length at most k. Suppose we have a derivation $\alpha \Rightarrow_{\hat{G}} \alpha' \stackrel{*}{\Rightarrow}_{\hat{G}} \beta$, and suppose $u \in w(\alpha)$ and $v \in w(\beta)$.

There are two possibilities. Suppose the production used in the derivational step $\alpha \Rightarrow_{\hat{G}} \alpha'$ is of the form $N \to QR$, where $N, Q, R \in \hat{V}$. Then $\alpha = \beta N \gamma$, and $\alpha' = \beta Q R \gamma$ for some β, γ. Since $u \in w(\alpha)$, we must have $u = u_\beta u_N u_\gamma$, $u_\beta \in w(\beta), u_\gamma \in w(\gamma)$. Pick an element of $u_Q \in w(Q)$, and $u_R \in w(R)$. Clearly $u' = u_\beta u_Q u_R u_\gamma \in w(\alpha')$, and therefore by the inductive hypothesis $u' \equiv_L v$. Since there is a production $N \to QR$ in the grammar $\hat{G}$, it must be the case that u_N was in the same component as $u_Q u_R$. Therefore $u_N \equiv_L u_Q u_R$, which implies $u \equiv_L u'$, which establishes that $u \equiv_L v$. Alternatively suppose the production used is of the form $N \to a$. As before we have $u' = u_\beta a u_\gamma$. By the inductive hypothesis $u' \equiv_L v$, and by the construction of the grammar we have $a \in w(N)$ therefore $u \equiv_L v$. QED

Lemma 5. $L(\hat{G}) \subseteq L(G)$

Since $w(u) = \{u\}$, it immediately follows from the previous lemma that if $\alpha \stackrel{*}{\Rightarrow}_{\hat{G}} u$ and $\alpha \stackrel{*}{\Rightarrow}_{\hat{G}} v$, then $u \equiv_L v$. Suppose we have some $u \in L(\hat{G})$, then $\exists S \in \hat{I}$ such that $S \stackrel{*}{\Rightarrow}_{\hat{G}} u$. Since $S \in \hat{I}$ there must have been a string $v \in L(G)$ which occurred frequently and is thus in $w(S)$. Therefore $v \equiv_L u$, which means that $u \in L(G)$. QED

Now we prove that the error will be small.

Lemma 6. $P_D(L(G) - L(\hat{G})) < \epsilon$

First of all we define the error set to be

$$L_{error} = \left(\bigcup_{N \to \alpha \in P : P_D(W(N \to \alpha)) < \epsilon_2} W(N \to \alpha) \right) \cup \bigcup_{S \in I : P_D(I(S)) < \epsilon_2} I(S) \quad (17)$$

We can define a partial mapping ϕ from the non-terminals in G to those in $\hat{G}$, (from $V \to \hat{V}$). If $N \in V$, then since it is μ_1-distinguishable we will have at least one frequent string u_N, since it is μ_2-reachable, the context distribution will have sufficently large norm, therefore there will be at least one non-terminal $[u_N]$ in $\hat{G}$. Every string $u \in U$ such that $N \stackrel{*}{\Rightarrow} u$, will be congruent to u_N and thus, since the sample is good, it will be in the same component. Therefore there is a unique non terminal in $\hat{V}$ corresponding to N, and thus ϕ is well defined. Define $\phi(a) = a$ for every letter $a \in \Sigma$ and then extend it to $(V_G \cup \Sigma)^+$. For every ϵ_2-frequent production $N \to \alpha$, we will also have a frequent string u such that $\alpha \stackrel{*}{\Rightarrow}_G u$. Write $\alpha = \alpha_1 \dots \alpha_k$ where $k = |\alpha|$,$\alpha_i \in \Sigma \cup V$ and let $u = u_1, \dots u_n$ where $\alpha_i \stackrel{*}{\Rightarrow} u_i$. By the construction of the set of productions we will have productions $[u] \to [u_1][u_2 \dots u_n]$, $[u_2 \dots u_n] \to [u_2][u_3 \dots u_n]$ up to $[u_{n-1} u_n] \to [u_{n-1}][u_n]$. Since for all of these productions $[u_i] = \phi(\alpha_i)$ we have that $\phi(N) \stackrel{*}{\Rightarrow}_{\hat{G}} \phi(\alpha)$. Similarly since the sample is good we will have that the non-terminal symbol S will be in the initial set, if it is frequent. Thus if $u \in L - L_{error}$, there will be a derivation $S \Rightarrow_G \alpha_1 \Rightarrow_G \cdots \Rightarrow_G \alpha_k \Rightarrow_G u$, for some $S \in I$; since u is not

in L_{error} all of the productions will be frequent and therefore there will be a derivation $\phi(S) \Rightarrow_{\hat{G}} \phi(\alpha_1) \Rightarrow_{\hat{G}} \ldots \phi(\alpha_k) \Rightarrow_{\hat{G}} u$, and thus $u \in L(\hat{G})$. Therefore $L(G) - L(\hat{G}) \subseteq L_{error}$, and since $P_D(L_{error}) < (p+n)\epsilon_2 = \epsilon$ we have the result. QED

Theorem 1. *If the sample is μ_3 good then* **PACCFG** *will generate a hypothesis grammar which is a subset of the target language and with error less than ϵ.*

6 Discussion

There are no directly comparable results for PAC-learning context free grammars. [Adr99] uses simple distributions in the Kolmogorov sense and membership queries to prove a learnability result for a class of rigid categorial grammars, but the proof is incomplete.

The results here are still incomplete – we are still in some sense hiding a language theoretic property in terms of a distributional property. The same problem occurs with the learning of DFAs in [RST98] – essentially we have some automata which will generally have exponentially small distinguishability. The best way of viewing this is that we have a parameter that measures the difficulty of learning some languages – the difficulty is affected both by the language and the distribution. This approach is worse, because while every PDFA is μ-distinguishable for some μ, it appears not to be necessarily the case that there will always be a polynomial that will bound the separability, though we have not yet managed to construct a counter-example.

Putting the bounds together and ignoring log factors we have, for a grammar $G = \langle \Sigma, V, P, I \rangle$ the sample complexity of $\mathcal{O}\left(\frac{|V|+|P|}{\epsilon \mu_1^l \mu_2^2 \nu^2}\right)$. The presence of the term μ_1^l is worrying. It is not always the case that one can convert an NTS grammar to a CNF grammar while preserving the NTS property. Thus l could in principle be large. However, a very slight strengthening of the NTS property, requiring the reduction system to be weakly confluent on $Sub(L)$ allows this reduction to take place, and thus allows l to be at most 2. Thus the sample complexity is not excessively high, and establish that polynomial learnability of interesting context free languages is an achievable goal.

This is a step towards a long term research goal of grammatical inference – finding a simple class of languages/distributions, defined in domain-general terms, that is provably learnable, in some practical sense, and includes observed natural languages. The class we present here is too limited in a number of respects; most importantly, the requirement that the grammars are unambiguous is too sharp, and we hope to remove this in future work. Additionally, just as our definition of separability allows a certain amount of overlap between the contexts of non congruent substrings, it might be possible to go beyond NTS languages, by allowing a limited amount of overlap between the strings generated by distinct non-terminals.

Acknowledgments

I would like to thank Remi Eyraud and Franck Thollard for collaborating on previous work that has led into this paper. This work was partly supported by the PASCAL network of excellence in machine learning.

References

[Adr99] Pieter Adriaans. Learning shallow context-free languages under simple distributions. Technical Report ILLC Report PP-1999-13, Institute for Logic, Language and Computation, Amsterdam, 1999.

[BS85] L Boasson and S Senizergues. NTS languages are deterministic and congruential. *J. Comput. Syst. Sci.*, 31(3):332–342, 1985.

[CE05] Alexander Clark and Remi Eyraud. Identification in the limit of substitutable context free languages. In Sanjay Jain, Hans Ulrich Simon, and Etsuji Tomita, editors, *Proceedings of The 16th International Conference on Algorithmic Learning Theory*, pages 283–296. Springer-Verlag, 2005.

[Cha01] Eugene Charniak. Immediate head parsing for language models. In *Proceedings of the 39th annual meeting of the ACL*, pages 116–123, Toulouse, France, 2001.

[Cla06] Alexander Clark. Learning deterministic context free grammars in the Omphalos competition. *Machine Learning*, 2006. to appear.

[CT04a] Alexander Clark and Franck Thollard. PAC-learnability of probabilistic deterministic finite state automata. *Journal of Machine Learning Research*, 5:473–497, May 2004.

[CT04b] Alexander Clark and Franck Thollard. Partially distribution-free learning of regular languages from positive samples. In *Proceedings of COLING*, Geneva, Switzerland, 2004.

[DR98] Devdatt P. Dubhashi and Desh Ranjan. Balls and bins: A study in negative dependence. *Random Structures and Algorithms*, 13(2):99–124, 1998.

[Har54] Zellig Harris. Distributional structure. *Word*, 10(2-3):146–62, 1954.

[JC00] F. Jelinek and C. Chelba. Structured language modeling for speech recognition. *Computer, Speech and Language*, 14(4), 283-332 2000.

[KMR+94] M.J. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R.E. Schapire, and L. Sellie. On the learnability of discrete distributions. In *Proc. of the 25th Annual ACM Symposium on Theory of Computing*, pages 273–282, 1994.

[RST98] D. Ron, Y. Singer, and N. Tishby. On the learnability and usage of acyclic probabilistic finite automata. *J. Comput. Syst. Sci.*, 56(2):133–152, 1998.

[Sen85] G Senizergues. The equivalence and inclusion problems for NTS languages. *J. Comput. Syst. Sci.*, 31(3):303–331, 1985.