Consistent Unsupervised Estimators for Anchored PCFGs

Alexander Clark
Department of Philosophy
King's College London
alexsclark@gmail.com

Nathanaël Fijalkow
CNRS, LaBRI, Bordeaux, and The
Alan Turing Institute of Data
Science, London
nathanael.fijalkow@labri.fr

Abstract

Learning probabilistic context-free grammars (PCFGs) from strings is a classic problem in computational linguistics since Horning (1969). Here we present an algorithm based on distributional learning that is a consistent estimator for a large class of PCFGs that satisfy certain natural conditions including being anchored (Stratos et al., 2016). We proceed via a reparameterization of (top–down) PCFGs that we call a bottom–up weighted context-free grammar. We show that if the grammar is anchored and satisfies additional restrictions on its ambiguity, then the parameters can be directly related to distributional properties of the anchoring strings; we show the asymptotic correctness of a naive estimator and present some simulations using synthetic data that show that algorithms based on this approach have good finite sample behavior.

1 Introduction

This paper presents an approach for strongly learning a linguistically interesting subclass of probabilistic context-free grammars (PCFGs) from strings in the realizable case. Unpacking this, we assume that we have some PCFG that we are interested in learning and that we have access only to a sample of strings generated by the PCFG (i.e., sampled from the distribution defined by the context-free grammar). Crucially, we do not observe the derivation trees—the hierarchical latent structure. Strong learning means that we want the learned grammar to define the same distribution over labeled trees as the original grammar and not just the same distribution over strings.

Clearly, there can be many structurally different PCFGs that define the same distribution over strings. Consider for example the distribution that generates a single string of length 3 with probability one and the various PCFGs that give rise to that same distribution; for these obvious reasons, that we discuss in more detail later, we cannot have an algorithm that does this for all PCFGs. Accordingly, we define some sufficient conditions on PCFGs for this algorithm to perform correctly. More precisely, we define some simple structural conditions on the underlying CFGs (in Section 3), and we will show that the resulting class of PCFGs is *identifiable from strings*, in the sense that any two PCFGs that define the same distribution over strings will be isomorphic.

We then provide a computationally trivial learning algorithm in Section 4, together with a proof that it will strongly learn every grammar in this class. The algorithm is not intended to be a realistic algorithm, but merely to illustrate the fundamental correctness of this general approach. We then show that general PCFGs in Chomsky normal form (CNF) that approximate the observable properties of natural language syntax are efficiently learnable using some simulations with synthetic data in Section 5.

Our primary scientific motivation is to understand the process of first-language acquisition, in particular the early phases of the acquisition of syntactic structure. Importantly, the grammar is not just a decision procedure that classifies strings as being grammatical or ungrammatical, but additionally assigns a tree structure to the grammatical sentences, a structure the primary role of which is to support semantic interpretation. The standard view is that children learn the syntactic structure of their languages not by purely syntactic means, but rather by using information about the range of available interpretations, derived from the situational context of the sentences they hear and inferences about the intentions and goals of the speaker (e.g., Abend et al., 2017). Indeed there is ample direct evidence from the developmental psycholinguistics literature that this does in fact happen at certain stages of language acquisition:

Transactions of the Association for Computational Linguistics, vol. 8, pp. 409–422, 2020. https://doi.org/10.1162/tacl_a_00323
Action Editor: Mark-Jan Nederhof. Submission batch: 11/2019; Revision batch: 4/2020; Published 7/2020.

For example, Gropen et al. (1991) showed that the acquisition of argument structure of verbs exploits semantic information about the verb and the arguments. However, the children in these experiments—the youngest cohort being nearly 4 years old—have already acquired a great deal of knowledge about English syntax.

Here, we are exploring an alternative or perhaps complementary hypothesis: namely, that the acquisition of the syntactic categories and rules of the language can to a certain extent be learned using only information derived from the surface strings without any appeal to external information about the hierarchical structure of the language that is being learned. In other words, the initial phases of language acquisition are based on purely syntactic information rather than the semantic bootstrapping discussed above.

The contributions of this paper are as follows. First, we provide a reparameterization of PCFGs within the space of weighted context-free grammars (WCFGs) that we call Bottom–up WCFGs. Next, we define three structural conditions on CFGs and show that they imply the identifiability of the class of all PCFGs based on those grammars. We then present a naive computationally trivial estimator and prove its asymptotic consistency for that class of PCFGs. We present some experiments on synthetic grammars that show that a variant of this algorithm has good finite sample behavior. Finally, we examine the extent to which these conditions are plausible, using a corpus of child-directed speech.

2 Definitions

We assume we have a finite set of atomic symbols Σ. The set of finite strings over this set is written Σ^*, nonempty finite strings are denoted by Σ^+, and the empty string is λ. We will typically write $a, b, c, \ldots$ for elements of Σ and $u, v, w, \ldots$ for elements of Σ^*. A (formal) language L is a subset of Σ^*. A context is an ordered pair of strings, that is, an element of $\Sigma^* \times \Sigma^*$ that we write as l, r. If U, V are languages, then their concatenation is UV defined in the normal way, and we will also write uV where u is a string instead of $\{u\}V$ and so on. Given a fixed language L, we define for a set of strings U a set of contexts $U^{\triangleright}$ as

$$U^{\triangleright} = \{l, r \mid lUr \subseteq L\}$$

If $U = \{u\}$ we will write $u^{\triangleright}$ for the distribution of u—the set of contexts in which it can occur.

A stochastic language is a function $\mathbb{P}$ from $\Sigma^* \to [0, 1]$, such that $\sum_{w \in \Sigma^*} \mathbb{P}(w) = 1$. Note that the support of this distribution is a formal language as defined above. We assume for the rest of the paper that the expected length of strings drawn from this distribution is finite.

We can define for some $u \in \Sigma^+$, the expected number of times that u will occur as a substring in a string distributed according to $\mathbb{P}$.[1]

$$\mathbb{E}(u) = \sum_{l,r \in \Sigma^* \times \Sigma^*} \mathbb{P}(lur)$$

We can also define, for a string u, its context distribution, which is a probability distribution over its contexts written $\mathcal{D}(u)$, whose support will be $u^{\triangleright}$, given for $l, r \in \Sigma^* \times \Sigma^*$ by

$$\mathcal{D}(u)[l, r] = \frac{\mathbb{P}(lur)}{\mathbb{E}(u)}.$$

Context-Free Grammars

We consider context-free grammars (CFGs) in Chomsky normal form $\langle \Sigma, V, S, P \rangle$ where Σ is a nonempty finite set of terminal symbols; V is a nonempty finite set, disjoint from Σ of nonterminal symbols, S is a distinguished element of V, the start symbol and P is a finite nonempty set of productions each of which is either of the form $A \to a$ where $A \in V$ and $a \in \Sigma$ or $A \to BC$ where $A \in V$ and $B, C \in V \setminus \{S\}$.[2]

We write $A, B, C, \ldots$ for elements of V and α for strings over $V \cup \Sigma$. A *derivation tree* τ is a singly rooted ordered tree where every node is labeled with an element of $V \cup \Sigma$ and each local tree is in P. The yield of a derivation is the string of symbols of leaves of the tree taken left to right; we write this as $y(\tau)$. The set of all derivations licensed by G and rooted by a nonterminal A, and with a yield in a set Γ is written as $\Omega(G, A, \Gamma)$; here we follow the notation of Smith and Johnson (2007) among others. We will omit G when it is clear.

[1] This is the expectation because if u occurs n times in a string w, there will be n distinct contexts l, r such that $lur = w$.

[2] We follow the classical definition of Chomsky normal form in not allowing S to occur on the right-hand side of any rules. This simplifies various parts of the analysis, and makes the learning problem slightly harder, but it is not hard to remove this restriction if it is desired. Note that we do not allow an empty right-hand side of a production.

We want to be able to combine trees using tree substitution; thus, if we have a tree τ_1 whose yield is lBr, where l and r are strings over Σ, and a tree τ_2 whose root is B and whose yield is α, we can combine them to get a tree $\tau_1 \otimes \tau_2$ whose yield is $l\alpha r$.

We define the string language defined by a nonterminal A to be

$$\mathcal{L}(G, A) = \left\{ y(\tau) : \tau \in \Omega(G, A, \Sigma^+) \right\}.$$

The string language defined by a CFG G is $\mathcal{L}(G) = \mathcal{L}(G, S)$.

For a tree τ and a production $A \to \alpha$ we write $f(A \to \alpha; \tau)$ for the number of times the production occurs in t. We write $|\tau|$ for the number of nonterminal symbols in a tree, and $|w|$ for the length of a string.

2.1 WCFGs

We will now consider the probabilistic case where we have a (discrete) probability distribution over trees, that is, over $\Omega(G, S, \Sigma^+)$, which will then define a stochastic language, whose support will be a context-free language. We will only consider those distributions which satisfy some simple conditional independence assumptions and can be represented by weighted CFGs.

A weighted CFG (WCFG) is a CFG together with a parameter function $\theta : P \to \mathbb{R}$ that maps productions to nonnegative real values; we will write this as $G; \theta$. The weight or score of a tree τ is the product of the weights of each production. Formally $s : \Omega(G) \to \mathbb{R}$ is defined as

$$s(\tau; \theta) = \prod_{A \to \alpha \in P} \theta(A \to \alpha)^{f(A \to \alpha; \tau)}$$

Note that $s(\tau_1 \otimes \tau_2) = s(\tau_1) s(\tau_2)$. In general we will define the score of a set of trees Ω to be the sum of the scores of the trees in that set: $s(\Omega) = \sum_{\tau \in \Omega} s(\tau)$. The weight of a string w is the sum of the weights of each derivation tree which yields w; $s(w) = s(\Omega(G, S, w))$.

Definition 2.1. The inside value of a nonterminal A, written $I(A)$ is

$$I(A) = s(\Omega(G, A, \Sigma^+))$$

Note that this quantity is sometimes called the *partition function*, written $Z(A)$. The outside value, $O(A)$, is defined likewise as

$$O(A) = s(\Omega(G, S, \Sigma^* A \Sigma^*))$$

Note that $O(S) = 1$ by definition, since $\Omega(G, S, \Sigma^* S \Sigma^*)$ is a single element set consisting of the trivial tree with one node S, which has score 1.

A WCFG is globally normalized if $I(S) = 1$. In this case it defines a probability distribution over trees, we can identify the probability of a tree with its score: $\mathbb{P}(\tau) = s(\tau)$, and via that a stochastic language.

2.2 Expectations

We define expectations of nonterminals, terminals, and productions, with respect to the distribution over trees defined by a globally normalized WCFGs.

Given a globally normalized WCFG, the quantity $\mathbb{E}(A \to \alpha)$ is the expected number of times the production $A \to \alpha$ occurs in a tree generated by the distribution induced by the grammar:

$$\mathbb{E}(A \to \alpha) = \sum_{\tau \in \Omega(G, S, \Sigma^+)} s(\tau) f(A \to \alpha; \tau)$$

Using this we define the expectation of a nonterminal:

$$\mathbb{E}(A) = \sum_{\alpha : A \to \alpha \in P} \mathbb{E}(A \to \alpha)$$

Note that $\mathbb{E}(S) = 1$ (because it can only occur at the root of every tree).

For nonterminals A, B, C and terminals a, the following identities relate the expectations and the inside and outside values, which can be established using the methods of, for example, Chi (1999).

$$\begin{aligned} \mathbb{E}(A) &= I(A) O(A) \\ \mathbb{E}(A \to a) &= O(A) \theta(A \to a) \\ \mathbb{E}(A \to BC) &= O(A) \theta(A \to BC) I(B) I(C) \end{aligned} \tag{1}$$

Note that for any nonterminal A that is not S, and any $\beta > 0$, we can scale all parameters for productions with A on the left-hand side by β, and every production with A on the right-hand side by β^{-1} (or β^{-2} if A occurs twice on the right-hand side), and the score of every tree will remain the same. There are two natural ways of resolving this arbitrariness: one is to stipulate that for all nonterminals $I(A) = 1$, which gives us the familiar PCFG. The parameters of a tight PCFG satisfy

$$\theta(A \to \alpha) = \frac{\mathbb{E}(A \to \alpha)}{\mathbb{E}(A)}. \tag{2}$$

The learning approach we take here is based on modeling the context distribution, and it is therefore more mathematically convenient to use the second normalization method where we stipulate that $O(A) = 1$ for all nonterminals. We now define this alternative parameterization, which we call a *bottom–up* WCFG, in contrast to the top–down generative process associated with a PCFG.

Definition 2.2 (bottom–up WCFG)**.** We say that a WCFG is in bottom–up form if $I(S) = 1$, and for all nonterminals A, $O(A) = 1$.

If a WCFG is in bottom–up form then the parameters satisfy:

$$\theta(A \to BC) = \frac{\mathbb{E}(A \to BC)}{\mathbb{E}(B)\mathbb{E}(C)} \qquad (3)$$
$$\theta(A \to a) = \mathbb{E}(A \to a).$$

Note that in this form, we condition the parameters on the right-hand side of the production not on the left-hand side as is done with a PCFG.

There is a unique bijection between the class of tight PCFGs and bottom–up WCFGs; we can easily convert from one form to the other. We can efficiently compute the inside and outside values of a convergent WCFG using standard techniques (Hutchins, 1972; Nederhof and Satta, 2008; Etessami et al., 2012); these involve solving a system of quadratic equations (since the grammar is in Chomsky normal form) in the case of the inside values, which can be done using the Newton method or a fixed point iteration, and a linear system in the case of the outside values. The expectations of each production can then be computed using Equation 1 and then converted into a PCFG or bottom up WCFG as desired using Equations 2 and 3, respectively.

3 Identifiability

We assume that we have a sequence of strings generated independently and identically distributed (i.i.d.) from some distribution generated by an unknown PCFG or WCFG, which we call the target grammar.

We are interested in the problem of producing a PCFG from this input data that is close to the target PCFG; namely, the underlying CFG is isomorphic to the underlying CFG of the target grammar and additionally the parameters are within ϵ of the corresponding parameters of the target grammar: we call this being ϵ-close. Two CFGs are isomorphic if they are identical apart from the labels of the nonterminals; the isomorphism is just a bijection between the nonterminals and productions in the natural way.

Definition 3.1. Two WCFGs, $G; \theta$ and $G'; \theta'$, are ϵ-close if there is an CFG-isomorphism ϕ from G to G' such that for all $A \to \alpha$ in the grammars,

$$|\theta(A \to \alpha) - \theta'(\phi(A \to \alpha))| < \epsilon$$

More precisely, we say that a learning algorithm A is a consistent estimator for a class of globally normalized WCFGs, $\mathcal{G}$, if for every WCFG, G_*, θ_* in the class, for every $\epsilon, \delta > 0$, there is an N such that if the algorithm receives a sample of m strings, sampled i.i.d. where $m \geq N$ then it outputs a WCFG $\hat{G}, \hat{\theta}$ such that with probability at least $1 - \delta$ we have that $\hat{G}, \hat{\theta}$ is ϵ-close to G_*, θ_*.

3.1 Structural Conditions on Grammars

We now define three structural conditions on PCFGs that will be sufficient to guarantee identifiability of the class from strings.

Condition 3.1. A grammar G is *anchored* if for every nonterminal A, there exists a terminal a such that $A \to a \in P$ and, if $B \to a \in P$ then $B = A$. In other words a occurs on the right-hand side of exactly one production.

We will call such a terminal a characterizing terminal of A, and if a characterizes A we will sometimes write $[[a]]$ for A.

This condition is very close to a number of conditions that have been proposed in the literature both for topic modeling and for grammatical inference: We use here the terminology of Stratos et al. (2016), but similar ideas occur in, for example, Adriaans's (1999) approach to learning CFGs and Denis et al.'s (2004) approach to learning regular languages. This is also very closely related to what is called the 1-Finite Kernel Property in distributional learning of CFGs (Clark and Yoshinaka, 2016).

The key idea behind the learning algorithm is this: If every nonterminal has a characterizing terminal then we can infer the probabilities of the productions of the grammar from distributional properties of the strings of corresponding terminals. Thus if A, B, and C are nonterminals characterized by a, b, and c, respectively, then we can infer something about the parameter of the production $A \to BC$ by looking at the distributional properties of a and bc. And if A is a nonterminal

characterized by a and b is any terminal, then we can infer something about the parameter of the production $A \to b$ by looking at the distributional properties of a and b.

3.2 Divergences

We start by defining some quantities that depend only on a distribution over strings. Recall that the Rényi α-divergence (Rényi, 1961) between two discrete distributions P and Q is defined for $\alpha = \infty$

$$R_\infty(P\|Q) = \log \sup_x \frac{P(x)}{Q(x)} \tag{4}$$

Given two strings u, v we will be concerned with $\rho(u \to v)$,

$$\rho(u \to v) = R_\infty(\mathcal{D}(u)\|\mathcal{D}(v)) \tag{5}$$

This is an asymmetric nonnegative measure of ''distance'' between the context distributions of u and v, which takes the value 0 only when they are identical. Note that, because $u^\triangleright$ is the support of $\mathcal{D}(u)$,

$$e^{-\rho(u\to v)} = \frac{\mathbb{E}(u)}{\mathbb{E}(v)} \inf_{l,r\in u^\triangleright} \frac{\mathbb{P}(lvr)}{\mathbb{P}(lur)}$$

We can now state a foundational result, which relates the parameters of a production to these divergences. We will start by proving an inequality, that we will later strengthen to an equality under additional conditions.

Theorem 3.1. Suppose $G;\theta$ is a bottom–up WCFG, and G is anchored. Let D be the distribution it defines, and P the set of productions. Suppose that a, b, c are characterizing terminals for nonterminals A, B, C respectively. Then for any terminal d if $A \to d \in P$

$$\theta(A \to d) \le \mathbb{E}(d) e^{-\rho(a\to d)}$$

and if $A \to BC \in P$

$$\theta(A \to BC) \le \frac{\mathbb{E}(bc)}{\mathbb{E}(b)\mathbb{E}(c)} e^{-\rho(a\to bc)}$$

Proof. Suppose A is a nonterminal in G that is characterized by a. Then, for every context l, r, since the only way that we can derive an a is via A, $\mathbb{P}(lar) = s(\Omega(S, lAr))\theta(A \to a)$. Summing both sides with respect to l, r we obtain

$$\mathbb{E}(a) = O(A)\theta(A \to a)$$

Since $O(A) = 1$ in a bottom-up WCFG we have that

$$\theta(A \to a) = \mathbb{E}(a) \tag{6}$$

and therefore

$$s(\Omega(S, lAr)) = \frac{\mathbb{P}(lar)}{\mathbb{E}(a)} \tag{7}$$

Now consider lexical rules. Consider some production $A \to d$ in the grammar, where a characterizes A. Consider some $l, r \in a^\triangleright$. Since a is an anchor of A, we know that $s(\Omega(S, lAr)) > 0$, and therefore $\mathbb{P}(ldr) > 0$. Clearly

$$\mathbb{P}(ldr) \ge s(\Omega(S, lAr))\theta(A \to d) \tag{8}$$

since the probability on the left-hand side is a sum over the scores of many possible derivations, and the right-hand side is a sum over a subset of those derivations.

Therefore:

$$\theta(A \to d) \le \frac{\mathbb{P}(ldr)}{s(\Omega(S, lAr))}$$

Now using Equation 7, we obtain

$$\theta(A \to d) \le \mathbb{E}(a)\frac{\mathbb{P}(ldr)}{\mathbb{P}(lar)}$$

Because this is true for all $l, r \in a^\triangleright$ we have

$$\frac{\theta(A \to d)}{\mathbb{E}(d)} \le \frac{\mathbb{E}(a)}{\mathbb{E}(d)} \inf_{l,r\in a^\triangleright} \frac{\mathbb{P}(ldr)}{\mathbb{P}(lar)} = e^{-\rho(a\to d)}$$

The same argument goes through for the binary rules. Suppose we have A, B, C nonterminals characterized by a, b, c, respectively, and a production $A \to BC$ with parameter $\theta(A \to BC)$. Let l, r be some context in $a^\triangleright$, then $\mathbb{P}(lar) > 0$ and $\mathbb{P}(lbcr) > 0$. Clearly

$$\mathbb{P}(lbcr) \ge s(\Omega(S, lAr))\theta(A \to BC)\theta(B \to b)\theta(C \to c) \tag{9}$$

Therefore $\theta(A \to BC)$ is smaller than or equal to

$$\frac{\mathbb{P}(lbcr)}{s(\Omega(S, lAr))\theta(B \to b)\theta(C \to c)}.$$

Using Equation 6 twice, and Equation 7 we get

$$\theta(A \to BC) \le \frac{\mathbb{E}(bc)}{\mathbb{E}(b)\mathbb{E}(c)} \frac{\mathbb{E}(a)}{\mathbb{E}(bc)} \frac{\mathbb{P}(lbcr)}{\mathbb{P}(lar)}$$

Again, because this is true for all $l, r \in a^\triangleright$ we have

$$\theta(A \to BC) \le \frac{\mathbb{E}(bc)}{\mathbb{E}(b)\mathbb{E}(c)} e^{-\rho(a\to bc)}$$

□

This shows us that we have an *upper bound* on the parameters from a distributional property. But looking at Equations 8 and 9, we can consider the circumstances under which this inequality will be tight, in which case we can recover the parameters directly.

In particular, if the grammar is unambiguous (i.e., if every string has at most one derivation tree) then if the left-hand side of the inequality is nonzero we can immediately see that the inequality will become an equality. As it happens, there will also be equality under some much weaker conditions that we now define.

3.3 Ambiguity

We now define two closely related conditions that are both related to the degree of ambiguity of the grammar.

Condition 3.2. Suppose a CFG G contains a production $A \to \alpha$. We say that G has an unambiguous context for that production if there is a string w and strings l, u, r such that $w = lur$, $\Omega(G, S, w)$ is nonempty and

$$\Omega(G, S, w) = \Omega(G, S, lAr) \otimes \Omega(G, A, u)$$

and all elements of $\Omega(G, A, u)$ have an occurrence of $A \to \alpha$ at the root. A CFG is *locally unambiguous* if it has an unambiguous context for every production in its set of productions.

Informally this condition says that for every production there is some string which, although it can be ambiguous, always uses that production at the same point. Note that if G is locally unambiguous and is anchored, then for every binary production, $[[a]] \to [[b]][[c]]$ there will be a context l, r such that $lbcr$ satisfies the condition; and for every production $[[a]] \to b$ there will be a context l, r such that lbr satisfies the condition.

If a grammar is unambiguous, then every context is an unambiguous context for every derivation that uses it, but this condition is much weaker than that; indeed, we don't need there to be any unambiguous strings, since $\Omega(G, S, lAr)$ can have more than one element.

Lemma 3.1. *If $G; \theta$ is a bottom–up WCFG and G is anchored and is locally unambiguous, then if* $[[a]] \to b \in P$

$$\theta([[a]] \to b) = \mathbb{E}(b) e^{-\rho(a \to b)}$$

and if $[[a]] \to [[b]][[c]] \in P$

$$\theta([[a]] \to [[b]][[c]]) = \frac{\mathbb{E}(bc)}{\mathbb{E}(b)\mathbb{E}(c)} e^{-\rho(a \to b, c)}$$

Proof. If we have a production $[[a]] \to [[b]][[c]]$ in the grammar, we know there is a context such that $\Omega(S, lwr) = \Omega(S, l[[a]]r) \otimes \Omega([[a]], w)$ where all the elements of $\Omega(A, w)$ have an occurrence of $[[a]] \to [[b]][[c]]$ at the root. Because we know that $\Omega([[a]], bc)$ consists of a single tree using $[[a]] \to [[b]][[c]]$; and $\Omega(S, l[[b]][[c]]r) = \Omega(S, l[[a]]r) \otimes \Omega([[a]], [[b]][[c]])$, therefore $\Omega(S, lbcr) = \Omega(S, l[[a]]r) \otimes \Omega(A, bc)$. Now we apply the same manipulations to get that for this l, r

$$\theta([[a]] \to [[b]][[c]]) = \frac{\mathbb{E}(a)}{\mathbb{E}(b)\mathbb{E}(c)} \frac{\mathbb{P}(lbcr)}{\mathbb{P}(lar)}$$

and therefore

$$\theta([[a]] \to [[b]][[c]]) = \frac{\mathbb{E}(bc)}{\mathbb{E}(b)\mathbb{E}(c)} e^{-\rho(a \to bc)}.$$

The argument for lexical rules is analogous. □

We can understand this better by taking the log.

$$\begin{aligned} & \log \theta([[a]] \to [[b]][[c]]) \\ = \; & \log \tfrac{\mathbb{E}(bc)}{\mathbb{E}(b)\mathbb{E}(c)} - \rho(a \to bc) \end{aligned} \tag{10}$$

The natural parameter is then the sum of two terms: The first is just the pointwise mutual information (Church and Hanks, 1990) between b and c.[3] The second term penalizes cases where the right-hand side is distributionally dissimilar from the left-hand side. For the lexical productions, similarly we have two terms:

$$\log \theta([[a]] \to b) = \log \mathbb{E}(b) - \rho(a \to b) \tag{11}$$

3.4 Upward Monotonicity

We need one more condition, however. There may be many different grammars that define the same distribution over strings that satisfy these two conditions because we may have multiple nonterminals that could be merged together.

Condition 3.3. A grammar $G = \langle \Sigma, V, S, P \rangle$ is *strictly upward monotonic* if for all $Q \supset P$, $\mathcal{L}(\langle \Sigma, V, S, Q \rangle) \supset \mathcal{L}(G)$. (Where Q is restricted to CNF productions of $V \times (\Sigma \cup V^2)$.)

[3] With an adjustment of $\log \mathbb{E}(|w|)$ because they are expectations and not probabilities.

Informally, if we add a new production to the grammar, then the language defined increases. Note that of course all grammars have the property that if $Q \supseteq P$, then $\mathcal{L}(\langle \Sigma, V, S, Q \rangle \supseteq \mathcal{L}(G)$. Here we require this monotonicity to be strict.

We define the set of derivation contexts of a nonterminal A to be

$$\mathcal{C}(G, A) = \{l, r : \Omega(G, S, lAr) \neq \emptyset\}.$$

Lemma 3.2. *Suppose G is anchored and upward monotonic: If A, B are nonterminals and $\mathcal{C}(G, A) = \mathcal{C}(G, B)$ then $A = B$.*

Proof. Let a be an anchor for A; we can clearly add the production $B \to a$ without increasing the language generated. Therefore, $B \to a$ is in the grammar, and so $A = B$ as a is an anchor. □

Lemma 3.3. *Suppose G is anchored and upward monotonic: Then*

$$[[a]] \to b \in P \text{ iff } a^{\triangleright} \subseteq b^{\triangleright}$$

and

$$[[a]] \to [[b]][[c]] \in P \text{ iff } a^{\triangleright} \subseteq (bc)^{\triangleright}$$

Using the same condition we can show that productions not in the grammar will have parameters zero, because of an infinite divergence term.

Lemma 3.4. *Suppose G is anchored, and upward monotonic, then*

- *If $[[a]] \to b$ is not in the grammar, then $\rho(a \to b) = \infty$.*
- *If $[[a]] \to [[b]][[c]]$ is not in the grammar, then $\rho(a \to bc) = \infty$.*

Proof. If $A \to b$ is not in the grammar, then by Lemma 3.3, there is some l, r such that lar is in the language but lbr is not in the language and so $\rho(a \to b) = \infty$. Similarly for binary rules. □

3.5 Selecting Nonterminals

The preceding discussion shows that if we have a set of terminals that are anchors for the true nonterminals in the original grammar, then the productions and the (bottom–up) parameters of the associated productions will be fixed correctly, but it says nothing about parameters that might be associated to productions that use other nonterminals. However, it is easy to show that under these assumptions there can be no other nonterminals.

Lemma 3.5. *Suppose G_1 and G_2 are anchored and strictly monotonic, and are weakly equivalent. Then they are isomorphic, and there is a unique isomorphism between them.*

Proof. Let A be a nonterminal in G_1, and let a be an anchor for A. Suppose $B \to a$ be some production in G_2. Let b be an anchor for B. Therefore $a^{\triangleright} \supseteq b^{\triangleright}$. By a similar argument there must be a nonterminal C in G_1 and a terminal c that anchors C such that $b^{\triangleright} \supseteq c^{\triangleright}$. But because $a^{\triangleright} \supseteq c^{\triangleright}$, we must have a production $C \to a$ in G_1. Since a is an anchor $C = A$, and therefore $a^{\triangleright} = b^{\triangleright} = c^{\triangleright}$. Therefore $\mathcal{C}(G_1, A) = \mathcal{C}(G_2, B)$.

Let ϕ then be the CFG-morphism from $G_1 \to G_2$, defined by $\phi(A) = A'$ iff $\mathcal{C}(G_1, A) = \mathcal{C}(G_2, A')$. This is well defined by Lemma 3.2, and is clearly a bijection. Given this bijection, by Lemma 3.3, they will have the same set of productions, and thus be isomorphic. □

3.6 Identifiability

We can now define the classes of grammars that we are interested in. Let $\mathfrak{G}_{\mathrm{A}}$ be the set of all trim CFGs that are in Chomsky normal form, anchored (Condition 3.1), are locally unambiguous (Condition 3.2), and are strictly upward monotonic (Condition 3.3).

Let $\mathfrak{P}_{\mathrm{A}}$ be the set of all tight PCFGs with finite expectations, with CFGs in $\mathfrak{G}_{\mathrm{A}}$, and let $\mathfrak{W}_{\mathrm{A}}$ be the set of all WCFGs in bottom–up form with CFGs in $\mathfrak{G}_{\mathrm{A}}$.

Theorem 3.2. *Suppose $G_1; \theta_1$ and $G_2; \theta_2$ are in $\mathfrak{W}_{\mathrm{A}}$ and are stochastically equivalent: In other words, for all $w \in \Sigma^+$, $\mathbb{P}(w; G_1) = \mathbb{P}(w; G_2)$, then G_1 is isomorphic to G_2, and if ϕ is the unique such morphism, for all $A \to \alpha$, $\theta_1(A \to \alpha) = \theta_2(\phi(A \to \alpha))$.*

Proof. Because they are stochastically equivalent, the support of their distributions is equal, and thus G_1 and G_2 are weakly equivalent. Therefore by Lemma 3.5 there is a unique isomorphism between them, ϕ. By Lemma 3.1 the parameters of corresponding productions must also be equal. □

Because there is a bijection between $\mathfrak{W}_{\mathrm{A}}$ and $\mathfrak{P}_{\mathrm{A}}$, $\mathfrak{P}_{\mathrm{A}}$ is also identifiable from strings.

4 Naive Estimators

We now analyze the properties of a particular estimator that we call the *naive plugin estimator*,

which we will show can learn all grammars in $\mathfrak{W}_{\mathrm{A}}$ and $\mathfrak{P}_{\mathrm{A}}$. This approach uses a trivial manner of estimating the ρ values, and from this we derive a consistent estimator for the class. This approach has poor sample complexity but is algorithmically trivial.

We will need to estimate the ρ divergences from a sample of strings drawn i.i.d. from the distribution defined by the grammar. Given a sample of strings, the most naive approach is to estimate $\mathbb{P}(w)$ and $\mathbb{E}(a)$ by the empirical distribution, to estimate the ratio as the ratio of these estimates, and to take the supremum over the frequent contexts of a rather than over the infinite set $a^{\triangleright}$.

We are interested in convergence in probability, which we will write as $\hat{X}_N \xrightarrow{N\to\infty} X$; in other words, for any $\epsilon, \delta > 0$, there is an n such that for all $N > n$, with probability greater than $1-\delta$ we have $|\hat{X}_N - X| < \epsilon$.

Let $w_1, \ldots, w_N$ be the sample of N strings drawn i.i.d from a target PCFG, and let $n(w)$ be the number of times that w occurs in the sample (as a whole string), and let $m(w)$ be the number of times substring occurs as a substring; clearly, $\sum_{l,r} n(lwr) = m(w)$. Define $\hat{\mathbb{P}}(w) = n(w)/N$ to be the empirical probability of w and $\hat{\mathbb{E}}(u) = m(w)/N$ to be the empirical expectation of u. Clearly, for any string w we have $\hat{\mathbb{P}}(w) \xrightarrow{N\to\infty} \mathbb{P}(w)$ and $\hat{\mathbb{E}}(w) \xrightarrow{N\to\infty} \mathbb{E}(w)$.

The naive plugin estimator is given by:

Definition 4.1. For $a, b, c \in \Sigma$ we define

$$\hat{\rho}_N(a \to bc) = \log \frac{\hat{\mathbb{E}}(bc)}{\hat{\mathbb{E}}(a)} \max_{l,r:n(lar)>\sqrt{N}} \frac{n(lar)}{n(lbcr)} \tag{12}$$

And for $a, b \in \Sigma$ we define

$$\hat{\rho}_N(a \to b) = \log \frac{\hat{\mathbb{E}}(b)}{\hat{\mathbb{E}}(a)} \max_{l,r:n(lar)>\sqrt{N}} \frac{n(lar)}{n(lbr)} \tag{13}$$

Note that $\hat{\rho}_N(a \to bc) = \infty$ if there is some context l, r such that $n(lar) > \sqrt{N}$, and $n(lbcr) = 0$.

We can show the convergence of the estimators when one side is anchored, starting with the case when the divergence is infinite.

Lemma 4.1. *For some $G; \theta \in \mathfrak{W}_{\mathrm{A}}$ suppose that a is an anchor for a nonterminal A and suppose that for some $b \in \Sigma$, $\rho(a \to b) = \infty$. Then for every $\delta > 0$, there is an N such that with probability at least $1-\delta$, $\hat{\rho}_N(a \to b) = \infty$. Similarly, if there is a c if $\rho(c \to a) = \infty$, there is an N such that with probability at least $1-\delta$, $\hat{\rho}_N(c \to a) = \infty$.*

Lemma 4.2. *For some $G; \theta \in \mathfrak{W}_{\mathrm{A}}$ suppose that a is an anchor for a nonterminal A, b for B, and c for C. If $\rho(a \to bc) = \infty$, then for every $\delta > 0$, there is an N such that with probability at least $1-\delta$ $\hat{\rho}_N(a \to bc) = \infty$.*

Proof. If $A \to BC$ were in P then $\rho(a \to bc)$ would be finite. So $A \to BC$ is not in P. By Condition 3.3, there must be some context l_*, r_* in $a^{\triangleright}$ but not in $(bc)^{\triangleright}$, and so for sufficiently large N, $l_* a r_*$ will occur more than $\sqrt{N}$ times. □

Lemma 4.3. *For some $G; \theta \in \mathfrak{W}_{\mathrm{A}}$ suppose that a is an anchor for a nonterminal A. Suppose $\rho(a \to b)$ is finite; then $\hat{\rho}_N(a \to b) \xrightarrow{N\to\infty} \rho(a \to b)$.*

Lemma 4.4. *For some $G; \theta \in \mathfrak{W}_{\mathrm{A}}$ suppose that a is an anchor for a nonterminal A, b for B, and c for C; if $\rho(a \to bc)$ is finite, then $\hat{\rho}_N(a \to bc) \xrightarrow{N\to\infty} \rho(a \to bc)$.*

When ρ is finite the convergence is straightforward since $|\{l, r : n(lar) > \sqrt{N}\}| \leq \sqrt{N}$ and so we can use Chernoff bounds in a standard way.

4.1 Definition of the Algorithm

We can now define the algorithm, taking as input a sequence of strings $\langle w_1, \ldots, w_N \rangle$ and using the trivial plugin estimators $\hat{\rho}_N$. The pseudocode is presented in Algorithm A. The algorithm starts by identifying the set of terminals that are anchors, which is illustrated in Figure 1. If a terminal d is not an anchor then there will be some terminal a which is an anchor such that $\rho(a \to d) < \infty$ and $\rho(d \to a) = \infty$; in other words, such that $a^{\triangleright} \subset d^{\triangleright}$. If the $\hat{\rho}_N$ estimates are infinite iff ρ is infinite, then we can see that Γ will be the set of possible anchors; that is, those terminals that occur on the right-hand side of exactly one production. Clearly, if a and b are anchors for the same nonterminal then $\rho(a \to b) = \rho(b \to a) = 0$, and if they are anchors for different nonterminals then $\rho(a \to b) = \rho(b \to a) = \infty$, so we can just group them into equivalence classes and pick the most frequent one from each class as the anchor. The start symbol will be anchored by the symbol that occurs most frequently as a whole sentence.

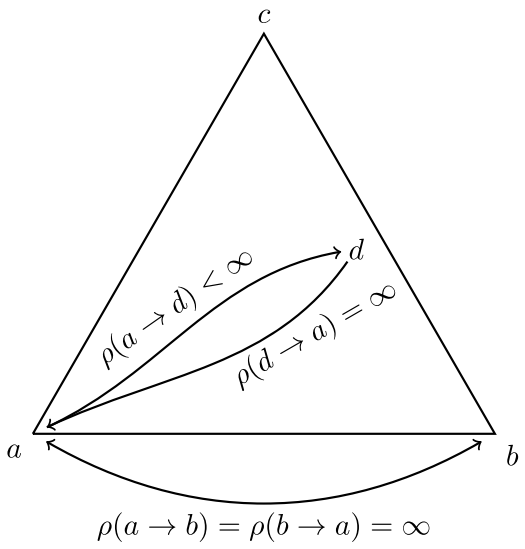

Figure 1: Diagram showing the terminal selection algorithm for a grammar with three nonterminals with anchors a, b, c. This diagram represents the space of context distributions: All terminals have a context distibution in the convex hull of the anchors. $d \notin \Gamma$ because $\rho(a \to d) < \infty$ but $\rho(d \to a) = \infty$, and it is therefore in the interior of the convex hull.

We can now prove that this algorithm is a consistent estimator for the class of WCFGs that we consider, $\mathfrak{W}_{\mathrm{A}}$.

Theorem 4.1. *For every grammar $G_*, \theta_* \in \mathfrak{W}_{\mathrm{A}}$, for every $\epsilon, \delta > 0$, there is an n such that when Algorithm A is run on a sample of N strings, $N > n$, generated i.i.d. from $G_*; \theta_*$ it produces a WCFG $G; \theta$ such that with probability at least $1 - \delta$*

- *G_* is CFG-isomorphic to G, and if ϕ is an isomorphism from G_* to G*
- $|\theta_*(A \to \alpha) - \theta(\phi(A \to \alpha))| < \epsilon$

Proof. (Sketch) Assume first that N is sufficiently large that $\hat{\rho}_N(a \to b)$ is close to $\rho(a \to b)$ for all a, b such that either a or b is an anchor; we can then show that Γ in Line 2 is just the set of possible anchors; and $a \sim b$ will be true iff a, b are anchors for the same nonterminal. We define a bijection between the nonterminals of the hypothesis and the target. Line 5 picks the start symbol to be the unique anchor that can occur in a length 1 string. The grammar will have the right productions via Lemma 3.3, and the parameters will converge via Lemmas 4.3 and 4.4. □

The output of this is a WCFG that may be divergent: We therefore define Algorithm B that uses the inside outside (IO) algorithm (Eisner, 2016) to normalize the WCFG produced by Algorithm A; we take the output WCFG and run one iteration of the IO algorithm on the same data to estimate the expectations of all the rules that are then normalized to produce a PCFG. Proving the convergence of this estimator requires a little bit of care. Chi (1999) shows that the result of this procedure will always be a tight PCFG; the finite expectation of $|\tau|$ allows us to apply a variant of the dominated convergence theorem combined with the law of large numbers to show that this is a consistent estimator for the class of grammars $\mathfrak{P}_{\mathrm{A}}$.

Input: A sequence of strings
$D = w_1, w_2, \ldots, w_N$
Output: A WCFG $G; \theta$

1 Compute $\hat{\rho}_N(a \to b)$ for all $a, b \in \Sigma$;
2 $\Gamma \leftarrow \{a \in \Sigma \mid \forall b \in \Sigma,\ \hat{\rho}_N(a \to b) < \infty \vee \hat{\rho}_N(b \to a) = \infty\}$;
3 Define the equivalence relation on Γ given by $a \sim b$ iff $\hat{\rho}_N(a \to b) < \infty$ and $\hat{\rho}_N(b \to a) < \infty$. Let Δ be the set formed by picking the terminal a with maximal $m(a)$ from each equivalence class in $\Gamma / \sim$;
4 $V \leftarrow \{[[a]] \mid a \in \Delta\}$;
5 $s \leftarrow \arg\max\{n(a) \mid a \in \Delta\}$;
6 $P_L \leftarrow \{[[a]] \to b \mid a \in \Delta, b \in \Sigma, \hat{\rho}_N(a \to b) < \infty\}$;
7 Compute $\hat{\rho}_N(a \to bc)$ for all $a, b, c \in \Delta$;
8 $P_B \leftarrow \{[[a]] \to [[b]][[c]] \mid a, b, c \in \Delta, \hat{\rho}_N(a \to bc) < \infty\}$;
9 $G \leftarrow \langle \Sigma, V, [[s]], P_L \cup P_B \rangle$;
10 $\theta([[a]] \to b) \leftarrow e^{-\hat{\rho}_N(a \to b)} \hat{\mathbb{E}}(b)$;
11 $\theta([[a]] \to [[b]][[c]]) \leftarrow e^{-\hat{\rho}_N(a \to bc)} \hat{\mathbb{E}}(bc) / \hat{\mathbb{E}}(b) \hat{\mathbb{E}}(c)$;
12 **return** $G; \theta$

Algorithm A: WCFG learner.

5 Experiments

The contributions of this paper are primarily theoretical but the reader may have legitimate concerns about the practicality of this approach given the naive estimator, the assumptions that are required, and the asymptotic nature of the correctness result. Here we present some computational simulations that address these issues, using synthetic PCFGs that mimic to a certain

extent the observable properties of child-directed speech (Pearl and Sprouse, 2012). We generate CFGs that have 10 nonterminals, 1,000 terminal symbols, and all possible rules in CNF; none of these grammars are in $\mathfrak{G}_A$. To obtain a PCFG, we sample the parameters for the binary productions and an extra parameter for the lexical rules from a symmetric Dirichlet distribution with parameter α, which we vary to control the degree of ambiguity of the grammar. We then train these parameters using the IO algorithm to get a distribution of lengths close to a zero-truncated Poisson with parameter 5. We then sample the conditional lexical parameters from a multivariate log normal distribution with $\sigma = 5$.[4]

To obtain a practical algorithm we follow Stratos et al. (2016). We consider only the local context—the immediate preceding and following word including a distinguished sentence boundary marker—and use Ney-Essen clustering (Ney et al., 1994) with 20 clusters to get a low-dimensional feature space. We give the learning algorithm the true number of nonterminals as a hyperparameter (in contrast to Algorithm A, which learns the number of nonterminals) and run the NMF algorithm of Stratos et al. (2016) to find the anchors, considering only those that occur at least 1,000 times. We set the lexical parameters using the Frank-Wolfe algorithm, and the binary parameters using the Renyi divergence with $\alpha = 5$. To alleviate data sparsity with estimating the distribution of the anchor bigrams when computing the binary rule parameters, we use all bigrams consisting of words that have probability at least 0.9 of being derived from the respective nonterminal. This produces a WCFG (A) which may be divergent. We then run one iteration of the IO algorithm[5] to obtain a PCFG (B), and then a further 10 iterations to get another PCFG (C); this is guaranteed to increase the likelihood of the model; if the PCFG B is sufficiently close to the target then this will converge towards the global optimum, the ML estimate; if not it will only converge to a local optimum.

For efficiency reasons we only run the IO algorithm on sentences of length at most 10; and we evaluate on lengths up to 20. The performance continues to improve with further iterations.

[4]This gives a Zipfian long-tailed distribution. We experimented also with a truncation of a Pitman Yor process with similar results.

[5]We are grateful to Mark Johnson for his efficient C implementation.

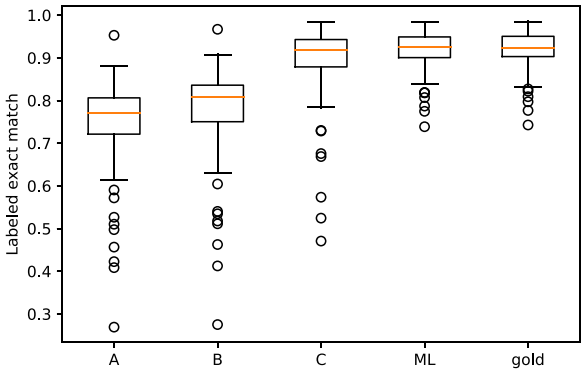

Figure 2: Box and whisker plot showing labeled exact match for 100 grammars sampled with $\alpha = 0.01$. We compare algorithms A, B, and C against gold (the target PCFG) and ML (the maximum likelihood PCFG learned by supervised learning from the training data).

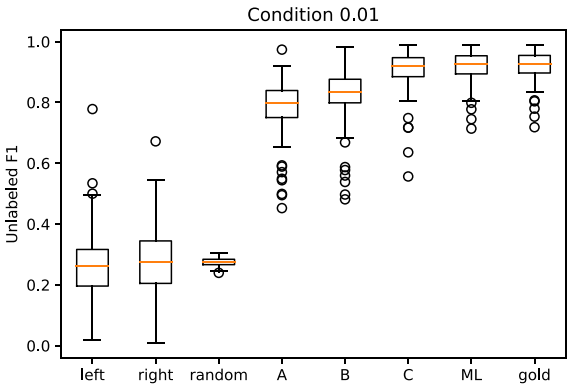

Figure 3: Box and whisker plot showing unlabeled accuracy. We add trivial baselines of left and right branching and random trees. 100 grammars sampled with $\alpha = 0.01$.

5.1 Results

After fixing the hyperparameters, we generate 100 different PCFGs for each condition, and sample 10^6 sentences from each. We evaluate the results according to how well they recover the true tree structures. We sample 1,000 trees from the target PCFG and evaluate the Viterbi parse of the yield of the tree using labeled exact match in Figure 2 and micro-averaged unlabeled precision/recall in Figure 3.[6] In all cases we exclude all forced choices so it is possible to score zero. The performance of the original grammar is a measure of the ambiguity of the grammar.

To see the effect of varying the degree of ambiguity, Figure 4 plots unlabeled exact match against the supervised baseline for values of $\alpha \in \{0.01, 0.1, 1.0\}$. For $\alpha = 1$ both are close to

[6]Because both trees are binary, precision is equal to recall.

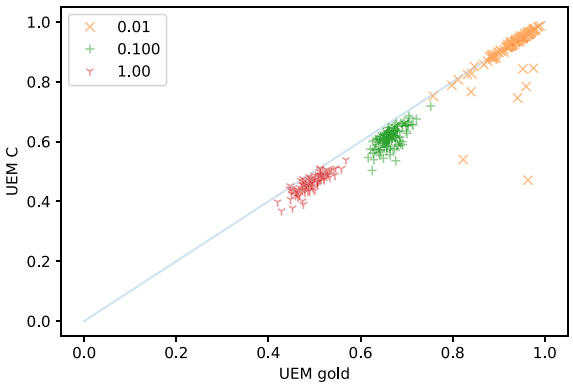

Figure 4: Scatter plot showing unlabeled exact match with the x-axis showing the ML model and the y-axis showing the algorithm C for three different values of the Dirichlet hyperparameter for the binary rules, $\alpha = 0.01, 0.1$, and 1.0. The diagonal line is the theoretical upper bound.

the random baseline; apart from that extreme case we find the performance degrading smoothly as predicted by theory. The labeled exact match (not shown here) in contrast shows a more pronounced decrease.

These grammars are about an order of magnitude smaller than plausible natural language grammars for child-directed speech as derived from the treebank in Pearl and Sprouse (2012), but this is largely for resource limitations because whereas Algorithm A is very fast, the IO algorithm is computationally expensive, and running these experiments on hundreds of synthetic grammars/languages at a time would be prohibitively expensive. It is certainly computationally feasible to run these experiments on single grammars with up to 100 nonterminals and 20,000 terminals. In small-scale experiments the results appear comparable with those we report here. The major failure mode is when there are nonterminals A where $\sum_a \mathbb{E}(A \rightarrow a)$ is very small. In those cases, though the grammar may be technically anchored, the anchors will be below the frequency threshold being considered.[7]

6 Applicability to Natural Language Corpora

An important question is whether this approach is directly applicable to natural language corpora either of transcribed child-directed speech or of text; a number of the assumptions we make are clearly false. First, even looking at English, we can see that the anchoring assumption is too strong. For example, the expletive pronouns in English, *there* and *it*, are both ambiguous, since *there* is also an adverb and *it* is also a personal pronoun, and so if there is a nonterminal representing such pronouns, then it will not be anchored.

When we consider phrasal categories, the question of whether such nonterminals are anchored requires asking two questions: first, whether such nonterminals generate single words at all, and secondly whether among those words we can find anchors. The existence of pro-forms, such as pronouns in the case of noun phrases, guarantees this for at least some categories. Clearly, this is genre-dependent, because it is sensitive to sentence length. Here we look at the Adam corpus of child-directed speech in English as syntactically annotated in the Penn treebank style by Pearl and Sprouse (2012). Table 1 shows the results. We can see that nonclausal categories are mostly anchored at this crude level of analysis, but that clausal categories are not. This implies that simple sentences without embedded clauses can be learned using this approach, but that learning complex clausal structures will require this approach to be extended at least to anchors of length more than one.

Most fundamentally, simple PCFGs of the type that we consider here are very poor models of natural language syntax. In order to obtain reasonable results, such grammars need to be lexicalized because otherwise the independence assumptions of the PCFG are violated because of semantic relations, for example, between a verb and its subject. Thus the realizability assumption the approach relies on is dramatically false.

7 Discussion and Conclusion

There are two ways of thinking about PCFGs: one is as a nontrivial CFG with parameters attached, where the support of the distribution is the language generated by the CFG, and the other is where the CFG is trivial, containing all possible productions, and where the support is the set of all strings; we can call these *sparse* and *dense* PCFGs, respectively. Hsu et al. (2013) show that in the dense case the class of PCFGs is not identifiable without additional constraints, even when one can exclude a set of grammars of measure

[7] Full code for reproducing these experiments is available at `https://github.com/alexc17/locallearner`.

t	$P(l=1)$	$w_{\max}$	$P(\mathrm{t}\|w_{\max})$
ADJP	0.67	careful	0.85
ADVP	0.84	already	1.0
FRAG	0.3	seal	0.2
INTJ	0.87	hmm	1.0
NP	0.7	he	1.0
PP	0.078	for	0.13
PRT	0.99	off	0.72
S	0.017	-	-
SBAR	0.0046	if	0.0024
SBARQ	0.0	-	-
SQ	0.021	-	-
VP	0.11	crying	0.82
WHADVP	0.98	when	1.0
WHNP	0.8	who	0.95

Table 1: Phrasal categories from the corpus of child-directed speech in Pearl and Sprouse (2012) showing that the proportion of length 1 yields the best anchor with frequency at least 10 and the proportion of tokens of that word that occurs as a yield of that tag.

zero.[8] The class of sparse PCFGs we consider, $\mathfrak{P}_{\mathrm{A}}$, has measure zero in their framework, and thus there is no incompatibility between their result and Theorem 3.2. However, there is some incompatibility between the empirical results in Section 5 and Hsu et al. (2013)'s result. With the protocol used in Section 5 we are indeed trying to learn a nonidentifiable class because the PCFGs are dense. However, the grammars are approximately anchored in the sense that for each nonterminal A there is a terminal a such that $\mathbb{E}(A \to a)$ is very close to $\mathbb{E}(a)$. In these cases, even though there are different parameter settings that give rise to the same distribution over strings, they will all be quite close to each other.

There have been many different attempts to solve this problem over the decades since the learning problem was initially introduced by Horning (1969); a useful survey of older work on learning CFGs is contained in Lee (1996). One strand of research looks at using the IO algorithm to train some heuristically initialized grammar (Baker, 1979; Lari and Young, 1990; Pereira and Schabes, 1992; de Marcken, 1999). However, this approach is only guaranteed to converge to a local maximum of the likelihood, and does not work well in practice. A related problem that we do not discuss in this paper is learning when the labeled tree structures are observed—essentially that of estimating a PCFG from a treebank, a problem which is algorithmically trivial and statistically well behaved, as Cohen and Smith (2012) show. The approach we take is most closely related to the work by Stratos et al. (2016) and work on weakly learning CFGs from samples generated by PCFGs developed by Shibata and Yoshinaka (2016). However, there are very few approaches to learning PCFGs with any nontrivial theoretical guarantees.

The approach here is essentially an exemplar-based model: The syntactic categories are based on single strings of length 1. This can be naturally extended, mutatis mutandis, to sets of exemplars, and to exemplars with length greater than 1. The extension beyond CFGs to mildly context sensitive grammars such as MCFGs (Seki et al., 1991) seems to present some problems that do not occur in the nonprobabilistic case (Clark and Yoshinaka, 2016); although the same bounds on the bottom up parameters can be derived, identifying the set of anchors seems to be challenging.

The variant of Algorithm A discussed in Section 5 is also interesting because it only uses local information in the initial phase: Indeed, it only uses the bigram and trigram counts, and it is only in the use of the IO algorithm that a pass through the data using the full sentence is used; this is compatible with psycholinguistic evidence about infants' abilities to track transitional probabilities (e.g., work following Saffran et al., 1996). Of course the original version in Section 4 uses complete sentences and not just the low-order counts.

Note that Equation 10 provides some theoretical justification for the long literature (Harris, 1955; McCauley and Christiansen, 2019) on using mutual information as a heuristic for unsupervised chunking. Although it is intuitively reasonable that chunks should correspond to subsequences that have high pointwise mutual information, it is gratifying to finally have some mathematical basis for these intuitions.

Acknowledgments

This work was partially carried out while the first author was a visiting researcher at The Alan Turing

[8] For technical reasons they consider only grammars where all probability mass is evenly distributed over all possible binary trees of a given length, and which are as a result highly ambiguous.

Institute. The second author was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1 and the DeLTA project (ANR-16-CE40-0007). We would like to thank the reviewers for helpful comments that have improved the paper.

References

Omri Abend, Tom Kwiatkowski, Nathaniel J. Smith, Sharon Goldwater, and Mark Steedman. 2017. Bootstrapping language acquisition. *Cognition*, 164:116–143.

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

James K. Baker. 1979. Trainable grammars for speech recognition. In *Speech Communication Papers for the 97th Meeting of the Acoustic Society of America*, pages 547–550.

Zhiyi Chi. 1999. Statistical properties of probabilistic context-free grammars. *Computational Linguistics*, 25(1):131–160.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Alexander Clark and Ryo Yoshinaka. 2016. Distributional learning of context-free and multiple context-free grammars. In Jeffrey Heinz and M. José Sempere, editors, *Topics in Grammatical Inference*, pages 143–172, Springer Berlin Heidelberg, Berlin, Heidelberg.

Shay B. Cohen and Noah A. Smith. 2012. Empirical risk minimization for probabilistic grammars: Sample complexity and hardness of learning. *Computational Linguistics*, 38(3): 479–526.

François Denis, Aurélien Lemay, and Alain Terlutte. 2004. Learning regular languages using RFSAs. *Theoretical Computer Science*, 313(2):267–294.

Jason Eisner. 2016. Inside-outside and forward-backward algorithms are just backprop (tutorial paper). In *Proceedings of the Workshop on Structured Prediction for NLP*, pages 1–17.

Kousha Etessami, Alistair Stewart, and Mihalis Yannakakis. 2012. Polynomial time algorithms for multi-type branching processes and stochastic context-free grammars. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, pages 579–588. ACM.

Jess Gropen, Steven Pinker, Michelle Hollander, and Richard Goldberg. 1991. Affectedness and direct objects: The role of lexical semantics in the acquisition of verb argument structure. *Cognition*, 41(1):153–195.

Zellig Harris. 1955. From phonemes to morphemes. *Language*, 31:190–222.

James Jay Horning. 1969. *A Study of Grammatical Inference*. Ph.D. thesis, Computer Science Department, Stanford University.

Daniel Hsu, Sham M. Kakade, and Percy Liang. 2013. Identifiability and unmixing of latent parse trees. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1520–1528.

Sandra E. Hutchins. 1972. Moments of string and derivation lengths of stochastic context-free grammars. *Information Sciences*, 4(2):179–191.

Karim Lari and Stephen J. Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–56.

Lillian Lee. 1996. Learning of context-free languages: A survey of the literature. Technical Report TR-12-96, Center for Research in Computing Technology, Harvard University.

Carl G. de Marcken. 1999. On the unsupervised induction of phrase-structure grammars. In *Natural Language Processing Using Very Large Corpora*, pages 191–208. Kluwer.

Stewart M. McCauley and Morten H. Christiansen. 2019. Language learning as language use: A cross-linguistic model of child language development. *Psychological Review*, 126(1):1.

Mark-Jan Nederhof and Giorgio Satta. 2008. Computing partition functions of PCFGs. *Research on Language and Computation*, 6(2):139–162.

Hermann Ney, Ute Essen, and Reinhard Kneser. 1994. On structuring probabilistic dependencies in stochastic language modelling. *Computer Speech and Language*, 8:1–38.

Lisa Pearl and Jon Sprouse. 2012. Computational models of acquisition for islands. In J. Sprouse and N. Hornstein, editors, *Experimental Syntax and Island Effects*. Cambridge University Press, Cambridge, UK.

Fernando Pereira and Yves Schabes. 1992. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 128–135.

Alfréd Rényi. 1961. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California.

Jenny R. Saffran, Richard N. Aslin, and Elissa L. Newport. 1996. Statistical learning by eight month old infants. *Science*, 274:1926–1928.

Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. 1991. On multiple context-free grammars. *Theoretical Computer Science*, 88(2):229.

Chihiro Shibata and Ryo Yoshinaka. 2016. Probabilistic learnability of context-free grammars with basic distributional properties from positive examples. *Theoretical Computer Science*, 620:46–72.

Noah A. Smith and Mark Johnson. 2007. Weighted and probabilistic context-free grammars are equally expressive. *Computational Linguistics*, 33(4):477–491.

Karl Stratos, Michael Collins, and Daniel Hsu. 2016. Unsupervised part-of-speech tagging with anchor hidden Markov models. *Transactions of the Association for Computational Linguistics*, 4:245–257.