Empirical Risk Minimization for Probabilistic Grammars: Sample Complexity and Hardness of Learning

Shay B. Cohen*
Columbia University

Noah A. Smith**
Carnegie Mellon University

Probabilistic grammars are generative statistical models that are useful for compositional and sequential structures. They are used ubiquitously in computational linguistics. We present a framework, reminiscent of structural risk minimization, for empirical risk minimization of probabilistic grammars using the log-loss. We derive sample complexity bounds in this framework that apply both to the supervised setting and the unsupervised setting. By making assumptions about the underlying distribution that are appropriate for natural language scenarios, we are able to derive distribution-dependent sample complexity bounds for probabilistic grammars. We also give simple algorithms for carrying out empirical risk minimization using this framework in both the supervised and unsupervised settings. In the unsupervised case, we show that the problem of minimizing empirical risk is NP-hard. We therefore suggest an approximate algorithm, similar to expectation-maximization, to minimize the empirical risk.

1. Introduction

Learning from data is central to contemporary computational linguistics. It is in common in such learning to estimate a model in a parametric family using the maximum likelihood principle. This principle applies in the supervised case (i.e., using annotated data) as well as semisupervised and unsupervised settings (i.e., using unannotated data). *Probabilistic grammars* constitute a range of such parametric families we can estimate (e.g., hidden Markov models, probabilistic context-free grammars). These parametric families are used in diverse NLP problems ranging from syntactic and morphological processing to applications like information extraction, question answering, and machine translation.

Estimation of probabilistic grammars, in many cases, indeed starts with the principle of maximum likelihood estimation (MLE). In the supervised case, and with traditional parametrizations based on multinomial distributions, MLE amounts to

* Department of Computer Science, Columbia University, New York, NY 10027, United States. E-mail: scohen@cs.columbia.edu. This research was completed while the first author was at Carnegie Mellon University.
** School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, United States. E-mail: nasmith@cs.cmu.edu.

Submission received: 1 November 2010; revised submission received: 21 June 2011; accepted for publication: 3 August 2011.

© 2012 Association for Computational Linguistics

normalization of rule frequencies as they are observed in data. In the unsupervised case, on the other hand, algorithms such as expectation-maximization are available. MLE is attractive because it offers statistical consistency if some conditions are met (i.e., if the data are distributed according to a distribution in the family, then we will discover the correct parameters if sufficient data is available). In addition, under some conditions it is also an unbiased estimator.

An issue that has been far less explored in the computational linguistics literature is the *sample complexity* of MLE. Here, we are interested in quantifying the number of samples required to accurately learn a probabilistic grammar either in a supervised or in an unsupervised way. If bounds on the requisite number of samples (known as "sample complexity bounds") are sufficiently tight, then they may offer guidance to learner performance, given various amounts of data and a wide range of parametric families. Being able to reason analytically about the amount of data to annotate, and the relative gains in moving to a more restricted parametric family, could offer practical advantages to language engineers.

We note that grammar learning has been studied in formal settings as a problem of *grammatical inference*—learning the structure of a grammar or an automaton (Angluin 1987; Clark and Thollard 2004; de la Higuera 2005; Clark, Eyraud, and Habrard 2008, among others). Our setting in this article is different. We assume that we have a fixed grammar, and our goal is to estimate its parameters. This approach has shown great empirical success, both in the supervised (Collins 2003; Charniak and Johnson 2005) and the unsupervised (Carroll and Charniak 1992; Pereira and Schabes 1992; Klein and Manning 2004; Cohen and Smith 2010a) settings. There has also been some discussion of sample complexity bounds for statistical parsing models, in a distribution-free setting (Collins 2004). The distribution-free setting, however, is not ideal for analysis of natural language, as it has to account for pathological cases of distributions that generate data.

We develop a framework for deriving sample complexity bounds using the maximum likelihood principle for probabilistic grammars in a distribution-dependent setting. Distribution dependency is introduced here by making empirically justified assumptions about the distributions that generate the data. Our framework uses and significantly extends ideas that have been introduced for deriving sample complexity bounds for probabilistic graphical models (Dasgupta 1997). Maximum likelihood estimation is put in the empirical risk minimization framework (Vapnik 1998) with the loss function being the log-loss. Following that, we develop a set of learning theoretic tools to explore rates of estimation convergence for probabilistic grammars. We also develop algorithms for performing empirical risk minimization.

Much research has been devoted to the problem of learning finite state automata (which can be thought of as a class of grammars) in the Probably Approximately Correct setting, leading to the conclusion that it is a very hard problem (Kearns and Valiant 1989; Pitt 1989; Terwijn 2002). Typically, the setting in these cases is different from our setting: Error is measured as the probability mass of strings that are not identified correctly by the learned finite state automaton, instead of measuring KL divergence between the automaton and the true distribution. In addition, in many cases, there is also a focus on the distribution-free setting. To the best of our knowledge, it is still an open problem whether finite state automata are learnable in the distribution-dependent setting when measuring the error as the fraction of misidentified strings. Other work (Ron 1995; Ron, Singer, and Tishby 1998; Clark and Thollard 2004; Palmer and Goldberg 2007) also gives treatment to probabilistic automata with an error measure which is more suitable for the probabilistic setting, such as Kullback-Lielder (KL) divergence or variation distance.

These also focus on learning the structure of finite state machines. As mentioned earlier, in our setting we assume that the grammar is fixed, and that our goal is to estimate its parameters.

We note an important connection to an earlier study about the learnability of probabilistic automata and hidden Markov models by Abe and Warmuth (1992). In that study, the authors provided positive results for the sample complexity for learning probabilistic automata—they showed that a polynomial sample is sufficient for MLE. We demonstrate positive results for the more general class of probabilistic grammars which goes beyond probabilistic automata. Abe and Warmuth also showed that the problem of finding or even approximating the maximum likelihood solution for a two-state probabilistic automaton with an alphabet of an arbitrary size is hard. Even though these results extend to probabilistic grammars to some extent, we provide a novel proof that illustrates the NP-hardness of identifying the maximum likelihood solution for probabilistic grammars in the specific framework of "proper approximations" that we define in this article. Whereas Abe and Warmuth show that the problem of maximum likelihood maximization for two-state HMMs is not approximable within a certain factor in time polynomial in the alphabet and the length of the observed sequence, we show that there is no polynomial algorithm (in the length of the observed strings) that identifies the maximum likelihood estimator in our framework. In our reduction, from 3-SAT to the problem of maximum likelihood estimation, the alphabet used is binary and the grammar size is proportional to the length of the formula. In Abe and Warmuth, the alphabet size varies, and the number of states is two.

This article proceeds as follows. In Section 2 we review the background necessary from Vapnik's (1988) empirical risk minimization framework. This framework is reduced to maximum likelihood estimation when a specific loss function is used: the log-loss.[1] There are some shortcomings in using the empirical risk minimization framework in its simplest form. In its simplest form, the ERM framework is *distribution-free*, which means that we make no assumptions about the distribution that generated the data. Naively attempting to apply the ERM framework to probabilistic grammars in the distribution-free setting does not lead to the desired sample complexity bounds. The reason for this is that the log-loss diverges whenever small probabilities are allocated in the learned hypothesis to structures or strings that have a rather large probability in the probability distribution that generates the data. With a distribution-free assumption, therefore, we would have to give treatment to distributions that are unlikely to be true for natural language data (e.g., where some extremely long sentences are very probable).

To correct for this, we move to an analysis in a distribution-dependent setting, by presenting a set of assumptions about the distribution that generates the data. In Section 3 we discuss probabilistic grammars in a general way and introduce assumptions about the true distribution that are reasonable when our data come from natural language examples. It is important to note that this distribution need not be a probabilistic grammar.

The next step we take, in Section 4, is *approximating* the set of probabilistic grammars over which we maximize likelihood. This is again required in order to overcome the divergence of the log-loss for probabilities that are very small. Our approximations are

1 It is important to remember that minimizing the log-loss does not equate to minimizing the error of a linguistic analyzer or natural language processing application. In this article we focus on the log-loss case because we believe that probabilistic models of language phenomena have inherent usefulness as explanatory tools in computational linguistics, aside from their use in systems.

based on *bounded approximations* that have been used for deriving sample complexity bounds for graphical models in a distribution-free setting (Dasgupta 1997).

Our approximations have two important properties: They are, by themselves, probabilistic grammars from the family we are interested in estimating, and they become a tighter approximation around the family of probabilistic grammars we are interested in estimating as more samples are available.

Moving to the distribution-dependent setting and defining proper approximations enables us to derive sample complexity bounds. In Section 5 we present the sample complexity results for both the supervised and unsupervised cases. A question that lingers at this point is whether it is computationally feasible to maximize likelihood in our framework even when given enough samples.

In Section 6, we describe algorithms we use to estimate probabilistic grammars in our framework, when given access to the required number of samples. We show that in the supervised case, we can indeed maximize likelihood in our approximation framework using a simple algorithm. For the unsupervised case, however, we show that maximizing likelihood is NP-hard. This fact is related to a notion known in the learning theory literature as **inherent unpredictability** (Kearns and Vazirani 1994): Accurate learning is computationally hard even with enough samples. To overcome this difficulty, we adapt the expectation-maximization algorithm (Dempster, Laird, and Rubin 1977) to approximately maximize likelihood (or minimize log-loss) in the unsupervised case with proper approximations.

In Section 7 we discuss some related ideas. These include the failure of an alternative kind of distributional assumption and connections to regularization by maximum a posteriori estimation with Dirichlet priors. Longer proofs are included in the appendices. A table of notation that is used throughout is included as Table D.1 in Appendix D.

This article builds on two earlier papers. In Cohen and Smith (2010b) we presented the main sample complexity results described here; the present article includes significant extensions, a deeper analysis of our distributional assumptions, and a discussion of variants of these assumptions, as well as related work, such as that about the Tsybakov noise condition. In Cohen and Smith (2010c) we proved NP-hardness for unsupervised parameter estimation of probalistic context-free grammars (PCFGs) (without approximate families). The present article uses a similar type of proof to achieve results adapted to empirical risk minimization in our approximation framework.

2. Empirical Risk Minimization and Maximum Likelihood Estimation

We begin by introducing some notation. We seek to construct a predictive model that maps inputs from space $\mathcal{X}$ to outputs from space $\mathcal{Z}$. In this work, $\mathcal{X}$ is a set of strings using some alphabet Σ ($\mathcal{X} \subseteq \Sigma^*$), and $\mathcal{Z}$ is a set of derivations allowed by a grammar (e.g., a context-free grammar). We assume the existence of an unknown joint probability distribution $p(x, z)$ over $\mathcal{X} \times \mathcal{Z}$. (For the most part, we will be discussing discrete input and output spaces. This means that p will denote a probability mass function.) We are interested in estimating the distribution p from examples, either in a supervised setting, where we are provided with examples of the form $(x, z) \in \mathcal{X} \times \mathcal{Z}$, or in the unsupervised setting, where we are provided only with examples of the form $x \in \mathcal{X}$. We first consider the supervised setting and return to the unsupervised setting in Section 5. We will use q to denote the estimated distribution.

In order to estimate p as accurately as possible using $q(x, z)$, we are interested in minimizing the *log-loss*, that is, in finding q_{opt}, from a fixed family of distributions $\mathcal{Q}$ (also called "the concept space"), such that

$$q_{\text{opt}} = \operatorname*{argmin}_{q \in \mathcal{Q}} \mathbb{E}_p \left[-\log q \right] = \operatorname*{argmin}_{q \in \mathcal{Q}} - \sum_{x,z} p(x, z) \log q(x, z) \tag{1}$$

Note that if $p \in \mathcal{Q}$, then this quantity achieves the minimum when $q_{\text{opt}} = p$, in which case the value of the log-loss is the entropy of p. Indeed, more generally, this optimization is equivalent to finding q such that it minimizes the KL divergence from p to q.

Because p is unknown, we cannot hope to minimize the log-loss directly. Given a set of examples $(x_1, z_1), \ldots, (x_n, z_n)$, however, there is a natural candidate, the empirical distribution $\tilde{p}_n$, for use in Equation (1) instead of p, defined as:

$$\tilde{p}_n(x, z) = n^{-1} \sum_{i=1}^{n} \mathbb{I} \{(x, z) = (x_i, z_i)\}$$

where $\mathbb{I} \{(x, z) = (x_i, z_i)\}$ is 1 if $(x, z) = (x_i, z_i)$ and 0 otherwise.[2] We then set up the problem as the problem of **empirical risk minimization** (ERM), that is, trying to find q such that

$$q^* = \operatorname*{argmin}_{q \in \mathcal{Q}} \mathbb{E}_{\tilde{p}_n} \left[-\log q \right] \tag{2}$$

$$= \operatorname*{argmin}_{q \in \mathcal{Q}} -n^{-1} \sum_{i=1}^{n} \log q(x_i, z_i)$$

$$= \operatorname*{argmax}_{q \in \mathcal{Q}} n^{-1} \sum_{i=1}^{n} \log q(x_i, z_i) \tag{3}$$

Equation (3) immediately shows that minimizing empirical risk using the log-loss is equivalent to the maximizing likelihood, which is a common statistical principle used for estimating a probabilistic grammar in computational linguistics (Charniak 1993; Manning and Schütze 1999).[3]

As mentioned earlier, our goal is to estimate the probability distribution p while quantifying how accurate our estimate is. One way to quantify the estimation accuracy is by bounding the **excess risk**, which is defined as

$$\mathcal{E}_p(q; \mathcal{Q}) = \mathcal{E}_p(q) \triangleq \mathbb{E}_p \left[-\log q \right] - \min_{q' \in \mathcal{Q}} \mathbb{E}_p \left[-\log q' \right] \tag{4}$$

We are interested in bounding the excess risk for q^*, $\mathcal{E}_p(q^*)$. The excess risk is reduced to KL divergence between p and q if $p \in \mathcal{Q}$, because in this case the quantity $\min_{q' \in \mathcal{Q}} \mathbb{E} \left[-\log q' \right]$ is minimized with $q' = p$, and equals the entropy of p. In a typical

2 We note that $\tilde{p}_n$ itself is a random variable, because it depends on the sample drawn from p.

3 We note that being able to attain the minimum through an hypothesis q^* is not necessarily possible in the general case. In our instantiations of ERM for probabilistic grammars, however, the minimum can be attained. In fact, in the unsupervised case the minimum can be attained by more than a single hypothesis. In these cases, q^* is arbitrarily chosen to be one of these minimizers.

case, where we do not necessarily have $p \in \mathcal{Q}$, then the excess risk of q is bounded from above by the KL divergence between p and q.

We can bound the excess risk by showing the double-sided convergence of the empirical process $R_n(\mathcal{Q})$, defined as follows:

$$R_n(\mathcal{Q}) \triangleq \sup_{q \in \mathcal{Q}} \left| \mathbb{E}_{\tilde{p}_n} \left[-\log q \right] - \mathbb{E}_p \left[-\log q \right] \right| \to 0 \tag{5}$$

as $n \to \infty$. For any $\epsilon > 0$, if, for large enough n it holds that

$$\sup_{q \in \mathcal{Q}} \left| \mathbb{E}_{\tilde{p}_n} \left[-\log q \right] - \mathbb{E}_p \left[-\log q \right] \right| < \epsilon \tag{6}$$

(with high probability), then we can "sandwich" the following quantities:

$$\begin{aligned} \mathbb{E}_p \left[-\log q_{\text{opt}} \right] &\leq \mathbb{E}_p \left[-\log q^* \right] && (7) \\ &\leq \mathbb{E}_{\tilde{p}_n} \left[-\log q^* \right] + \epsilon \\ &\leq \mathbb{E}_{\tilde{p}_n} \left[-\log q_{\text{opt}} \right] + \epsilon \\ &\leq \mathbb{E}_p \left[-\log q_{\text{opt}} \right] + 2\epsilon && (8) \end{aligned}$$

where the inequalities come from the fact that q_{opt} minimizes the expected risk $\mathbb{E}_p \left[-\log q \right]$ for $q \in \mathcal{Q}$, and q^* minimizes the empirical risk $\mathbb{E}_{\tilde{p}_n} \left[-\log q \right]$ for $q \in \mathcal{Q}$. The consequence of Equations (7) and (8) is that the expected risk of q^* is at most 2ϵ away from the expected risk of q_{opt}, and as a result, we find the excess risk $\mathcal{E}_p(q^*)$, for large enough n, is smaller than 2ϵ. Intuitively, this means that, under a large sample, q^* does not give much worse results than q_{opt} under the criterion of the log-loss.

Unfortunately, the regularity conditions which are required for the convergence of $R_n(\mathcal{Q})$ do not hold because the log-loss can be unbounded. This means that a modification is required for the empirical process in a way that will actually guarantee some kind of convergence. We give a treatment to this in the next section.

We note that all discussion of convergence in this section has been about convergence *in probability*. For example, we want Equation (6) to hold with high probability—for most samples of size n. We will make this notion more rigorous in Section 2.2.

2.1 Empirical Risk Minimization and Structural Risk Minimization Methods

It has been noted in the literature (Vapnik 1998; Koltchinskii 2006) that often the class $\mathcal{Q}$ is too complex for empirical risk minimization using a fixed number of data points. It is therefore desirable in these cases to create a family of subclasses $\{\mathcal{Q}_\alpha \mid \alpha \in \mathcal{A}\}$ that have increasing complexity. The more data we have, the more complex our $\mathcal{Q}_\alpha$ can be for empirical risk minimization. Structural risk minimization (Vapnik 1998) and the method of sieves (Grenander 1981) are examples of methods that adopt such an approach. Structural risk minimization, for example, can be represented in many cases as a penalization of the empirical risk method, using a regularization term.

In our case, the level of "complexity" is related to allocation of small probabilities to derivations in the grammar by a distribution $q \in \mathcal{Q}$. The basic problem is this: Whenever we have a derivation with a small probability, the log-loss becomes very large (in absolute value), and this makes it hard to show the convergence of the empirical process

$R_n(\mathcal{Q})$. Because grammars can define probability distributions over infinitely many discrete outcomes, probabilities can be arbitrarily small and log-loss can be arbitrarily large.

To solve this issue with the complexity of $\mathcal{Q}$, we define in Section 4 a series of approximations $\{\mathcal{Q}_n \mid n \in \mathbb{N}\}$ for probabilistic grammars such that $\bigcup_n \mathcal{Q}_n = \mathcal{Q}$. Our framework for empirical risk minimization is then set up to minimize the empirical risk with respect to $\mathcal{Q}_n$, where n is the number of samples we draw for the learner:

$$q_n^* = \operatorname*{argmin}_{q \in \mathcal{Q}_n} \mathbb{E}_{\tilde{p}_n} \left[-\log q \right] \tag{9}$$

We are then interested in the convergence of the empirical process

$$R_n(\mathcal{Q}_n) = \sup_{q \in \mathcal{Q}_n} \left| \mathbb{E}_{\tilde{p}_n} \left[-\log q \right] - \mathbb{E}_p \left[-\log q \right] \right| \tag{10}$$

In Section 4 we show that the minimizer q_n^* is an *asymptotic* empirical risk minimizer (in our specific framework), which means that $\mathbb{E}_p \left[-\log q_n^* \right] \to \mathbb{E}_p \left[-\log q^* \right]$. Because we have $\bigcup_n \mathcal{Q}_n = \mathcal{Q}$, the implication of having asymptotic empirical risk minimization is that we have $\mathcal{E}_p(q_n^*; \mathcal{Q}_n) \to \mathcal{E}_p(q^*; \mathcal{Q})$.

2.2 Sample Complexity Bounds

Knowing that we are interested in the convergence of $R_n(\mathcal{Q}_n) = \sup_{q \in \mathcal{Q}_n} |\mathbb{E}_{\tilde{p}_n} \left[-\log q \right] - \mathbb{E}_p \left[-\log q \right]|$, a natural question to ask is: "At what *rate* does this empirical process converge?"

Because the quantity $R_n(\mathcal{Q}_n)$ is a random variable, we need to give a probabilistic treatment to its convergence. More specifically, we ask the question that is typically asked when learnability is considered (Vapnik 1998): "How many samples n are required so that with probability $1 - \delta$ we have $R_n(\mathcal{Q}_n) < \epsilon$?" Bounds on this number of samples are also called "sample complexity bounds," and in a distribution-free setting they are described as a function $N(\epsilon, \delta, \mathcal{Q})$, independent of the distribution p that generates the data.

A complete distribution-free setting is not appropriate for analyzing natural language. This setting poses technical difficulties with the convergence of $R_n(\mathcal{Q}_n)$ and needs to take into account pathological cases that can be ruled out in natural language data. Instead, we will make assumptions about p, parametrize these assumptions in several ways, and then calculate sample complexity bounds of the form $N(\epsilon, \delta, \mathcal{Q}, p)$, where the dependence on the distribution is expressed as dependence on the parameters in the assumptions about p.

The learning setting, then, can be described as follows. The user decides on a level of accuracy (ϵ) which the learning algorithm has to reach with confidence ($1 - \delta$). Then, $N(\epsilon, \delta, \mathcal{Q}, p)$ samples are drawn from p and presented to the learning algorithm. The learning algorithm then returns an hypothesis according to Equation (9).

3. Probabilistic Grammars

We begin this section by discussing the family of probabilistic grammars. A probabilistic grammar defines a probability distribution over a certain kind of structured object (a derivation of the underlying symbolic grammar) explained step-by-step as a stochastic

process. Hidden Markov models (HMMs), for example, can be understood as a random walk through a probabilistic finite-state network, with an output symbol sampled at each state. PCFGs generate phrase-structure trees by recursively rewriting nonterminal symbols as sequences of "child" symbols (each itself either a nonterminal symbol or a terminal symbol analogous to the emissions of an HMM).

Each step or emission of an HMM and each rewriting operation of a PCFG is conditionally independent of the others given a single structural element (one HMM or PCFG state); this Markov property permits efficient inference over derivations given a string.

In general, a probabilistic grammar $\langle G, \theta \rangle$ defines the joint probability of a string x and a grammatical derivation z:

$$q(x, z \mid \theta, G) = \prod_{k=1}^{K} \prod_{i=1}^{N_k} \theta_{k,i}^{\psi_{k,i}(x,z)} = \exp \sum_{k=1}^{K} \sum_{i=1}^{N_k} \psi_{k,i}(x, z) \log \theta_{k,i} \tag{11}$$

where $\psi_{k,i}$ is a function that "counts" the number of times the kth distribution's ith event occurs in the derivation. The parameters θ are a collection of K multinomials $\langle \theta_1, \ldots, \theta_K \rangle$, the kth of which includes N_k competing events. If we let $\theta_k = \langle \theta_{k,1}, \ldots, \theta_{k,N_k} \rangle$, each $\theta_{k,i}$ is a probability, such that

$$\forall k, \forall i, \qquad \theta_{k,i} \geq 0$$

$$\forall k, \qquad \sum_{i=1}^{N_k} \theta_{k,i} = 1$$

We denote by Θ_G this parameter space for θ. The grammar G dictates the support of q in Equation (11). As is often the case in probabilistic modeling, there are different ways to carve up the random variables. We can think of x and z as correlated structure variables (often x is known if z is known), or the derivation event counts $\psi(x, z) = \langle \psi_{k,i}(x, z) \rangle_{1 \leq k \leq K, 1 \leq i \leq N_k}$ as an integer-vector random variable. In this article, we assume that x is always a deterministic function of z, so we use the distribution $p(z)$ interchangeably with $p(x, z)$.

Note that there may be many derivations z for a given string x—perhaps even infinitely many in some kinds of grammars. For HMMs, there are three kinds of multinomials: a starting state multinomial, a transition multinomial per state and an emission multinomial per state. In that case $K = 2s + 1$, where s is the number of states. The value of N_k depends on whether the kth multinomial is the starting state multinomial (in which case $N_k = s$), transition multinomial ($N_k = s$), or emission multinomial ($N_k = t$, with t being the number of symbols in the HMM). For PCFGs, each multinomial among the K multinomials corresponds to a set of N_k context-free rules headed by the same nonterminal. The parameter $\theta_{k,i}$ is then the probability of the ith rule for the kth nonterminal.

We assume that G denotes a fixed grammar, such as a context-free or regular grammar. We let $N = \sum_{k=1}^{K} N_k$ denote the total number of derivation event types. We use $D(G)$ to denote the set of all possible derivations of G. We define $D_x(G) = \{z \in D(G) \mid \text{yield}(z) = x\}$. We use $\deg(G)$ to denote the "degree" of G, i.e., $\deg(G) = \max_k N_k$. We let $|x|$ denote the length of the string x, and $|z| = \sum_{k=1}^{K} \sum_{i=1}^{N_k} \psi_{k,i}(z)$ denote the "length" (number of event tokens) of the derivation z.

Going back to the notation in Section 2, $\mathcal{Q}$ would be a collection of probabilistic grammars, parametrized by θ, and q would be a specific probabilistic grammar with a specific θ. We therefore treat the problem of ERM with probabilistic grammars as the problem of parameter estimation—identifying θ from complete data or incomplete data (strings x are visible but the derivations z are not). We can also view parameter estimation as the identification of a *hypothesis* from the concept space $\mathcal{Q} = \mathcal{H}(G) = \{h_\theta(z) \mid \theta \in \Theta_G\}$ (where h_θ is a distribution of the form of Equation [11]) or, equivalently, from negated log-concept space $\mathcal{F}(G) = \{-\log h_\theta(z) \mid \theta \in \Theta_G\}$. For simplicity of notation, we assume that there is a fixed grammar G and use $\mathcal{H}$ to refer to $\mathcal{H}(G)$ and $\mathcal{F}$ to refer to $\mathcal{F}(G)$.

3.1 Distributional Assumptions about Language

In this section, we describe a parametrization of assumptions we make about the distribution $p(x, z)$, the distribution that generates derivations from $D(G)$ (note that p does not have to be a probabilistic grammar). We first describe empirical evidence about the decay of the frequency of long strings x.

Figure 1 shows the frequency of sentence length for treebanks in various languages.[4] The trend in the plots clearly shows that in the extended tail of the curve, all languages have an exponential decay of probabilities as a function of sentence length. To test this, we performed a simple regression of frequencies using an exponential curve. We estimated each curve for each language using a curve of the form $f(l; c, \alpha) = cl^\alpha$. This estimation was done by minimizing squared error between the frequency versus sentence length curve and the approximate version of this curve. The data points used for the approximation are (l_i, p_i), where l_i denotes sentence length and p_i denotes frequency, selected from the extended tail of the distribution. Extended tail here refers to all points with length longer than l_1, where l_1 is the length with the highest frequency in the treebank. The goal of focusing on the tail is to avoid approximating the head of the curve, which is actually a monotonically increasing function. We plotted the approximate curve together with a length versus frequency curve for new syntactic data. It can be seen (Figure 1) that the approximation is rather accurate in these corpora.

As a consequence of this observation, we make a few assumptions about G and $p(x, z)$:

- Derivation length proportional to sentence length: There is an $\alpha \geq 1$ such that, for all z, $|z| \leq \alpha|\text{yield}(z)|$. Further, $|z| \geq |x|$. (This prohibits unary cycles.)
- Exponential decay of derivations: There is a constant $r < 1$ and a constant $L \geq 0$ such that $p(z) \leq Lr^{|z|}$. Note that the assumption here is about the frequency of length of separate derivations, and not the aggregated frequency of all sentences of a certain length (cf. the discussion above referring to Figure 1).

4 Treebanks offer samples of cleanly segmented sentences. It is important to note that the distributions estimated may not generalize well to samples from other domains in these languages. Our argument is that the family of the estimated curve is reasonable, not that we can correctly estimate the curve's parameters.

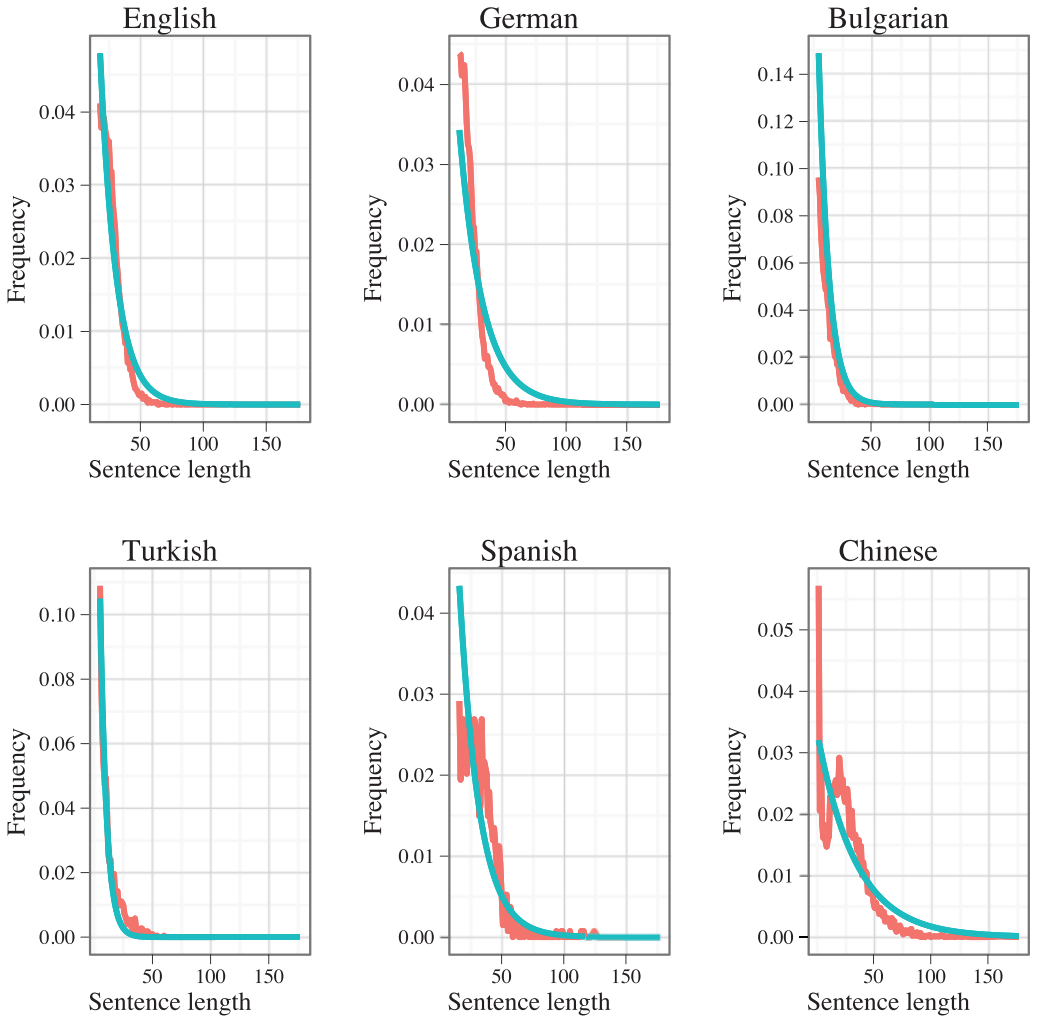


Figure 1
A plot of the tail of frequency vs. sentence length in treebanks for English, German, Bulgarian, Turkish, Spanish, and Chinese. Red lines denote data from the treebank, blue lines denote an approximation which uses an exponential function of the form $f(l; c, \alpha) = cl^{\alpha}$ (the blue line uses data which is different from the data used to estimate the curve parameters, c and α). The parameters (c, α) are (0.19, 0.92) for English, (0.06, 0.94) for German, (0.26, 0.89) for Bulgarian, (0.26, 0.83) for Turkish, (0.11, 0.93) for Spanish, and (0.03, 0.97) for Chinese. Squared errors are 0.0005, 0.0003, 0.0007, 0.0003, 0.001, and 0.002 for English, German, Bulgarian, Turkish, Spanish, and Chinese, respectively.

- Exponential decay of strings: Let $\Lambda(k) = |\{z \in D(G) \mid |z| = k\}|$ be the number derivations of length k in G. We assume that $\Lambda(k)$ is an increasing function, and complete it such that it is defined over positive numbers by taking $\Lambda(t) \triangleq \Lambda(\lceil t \rceil)$. Taking r as before, we assume there exists a constant $q < 1$, such that $\Lambda^2(k)r^k \leq q^k$ (and as a consequence, $\Lambda(k)r^k \leq q^k$). This implies that the number of derivations of length k may be exponentially large (e.g., as with many PCFGs), but is bounded by $(q/r)^k$.

- Bounded expectations of rules: There is a $B < \infty$ such that $\mathbb{E}_p \left[\psi_{k,i}(z)\right] \leq B$ for all k and i.

These assumptions must hold for any p whose support consists of a finite set. These assumptions also hold in many cases when p itself is a probabilistic grammar. Also, we note that the last requirement of bounded expectations is optional, and it can be inferred from the rest of the requirements: $B = L/(1-q)^2$. We make this requirement explicit for simplicity of notation later. We denote the family of distributions that satisfy all of these requirements by $\mathcal{P}(\alpha, L, r, q, B, G)$.

There are other cases in the literature of language learning where additional assumptions are made on the learned family of models in order to obtain positive learnability results. For example, Clark and Thollard (2004) put a bound on the expected length of strings generated from any state of probabilistic finite state automata, which resembles the exponential decay of strings we have for p in this article.

An immediate consequence of these assumptions is that the entropy of p is finite and bounded by a quantity that depends on L, r and q.[5] Bounding entropy of labels (derivations) given inputs (sentences) is a common way to quantify the *noise* in a distribution. Here, both the *sentential* entropy ($H_s(p) = -\sum_x p(x) \log p(x)$) is bounded as well as the *derivational* entropy ($H_d(p) = -\sum_{x,z} p(x,z) \log p(x,z)$). This is stated in the following result.

Proposition 1
Let $p \in \mathcal{P}(\alpha, L, r, q, B, G)$ be a distribution. Then, we have

$$H_s(p) \leq H_d(p) \leq -\log L + \frac{L \log r}{(1-q)^2} \log \frac{1}{r} + \frac{\lceil (1 + \log L)/\log \frac{1}{r} \rceil}{e} \Lambda \left(\left\lceil \frac{1 + \log L}{\log \frac{1}{r}} \right\rceil \right)$$

Proof
First note that $H_s(p) \leq H_d(p)$ holds by the data processing inequality (Cover and Thomas 1991) because the sentential probability distribution $p(x)$ is a coarser version of the derivational probability distribution $p(x,z)$. Now, consider $p(x,z)$. For simplicity of notation, we use $p(z)$ instead of $p(x,z)$. The yield of z, x, is a function of z, and therefore can be omitted from the distribution. It holds that

$$\begin{aligned} H_d(p) &= -\sum_z p(z) \log p(z) \\ &= -\sum_{z \in Z_1} p(z) \log p(z) - \sum_{z \in Z_2} p(z) \log p(z) \\ &= H_d(p, Z_1) + H_d(p, Z_2) \end{aligned}$$

where $Z_1 = \{z \mid p(z) > 1/e\}$ and $Z_2 = \{z \mid p(z) \leq 1/e\}$. Note that the function $-\alpha \log \alpha$ reaches its maximum for $\alpha = 1/e$. We therefore have

$$H_d(p, Z_1) \leq \frac{|Z_1|}{e}$$

5 For simplicity and consistency with the log-loss, we measure entropy in nats, which means we use the natural logarithm when computing entropy.

We give a bound on $|Z_1|$, the number of "high probability" derivations. Because we have $p(x,z) \leq Lr^{|z|}$, we can find the maximum length of a derivation that has a probability of more than $1/e$ (and hence, it may appear in Z_1) by solving $1/e \leq Lr^{|z|}$ for $|z|$, which leads to $|z| \leq \log(1/eL)/\log r$. Therefore, there are at most $\sum_{k=1}^{\lceil (1+\log L)/\log \frac{1}{r} \rceil} \Lambda(k)$ derivations in $|Z_1|$ and therefore we have

$$|Z_1| \leq \left\lceil (1+\log L)/\log \tfrac{1}{r} \right\rceil \Lambda\left(\left\lceil (1+\log L)/\log \tfrac{1}{r} \right\rceil\right)$$

$$H_d(p, Z_1) \leq \frac{\lceil (1+\log L)/\log \frac{1}{r} \rceil}{e} \Lambda\left(\left\lceil (1+\log L)/\log \tfrac{1}{r} \right\rceil\right) \tag{12}$$

where we use the monotonicity of Λ. Consider $H_d(p, Z_2)$ (the "low probability" derivations). We have:

$$\begin{aligned} H_d(p, Z_2) &\leq -\sum_{z \in Z_2} Lr^{|z|} \log\left(Lr^{|z|}\right) \\ &\leq -\log L - \left(L \log r\right) \sum_{z \in Z_2} |z| r^{|z|} \\ &\leq -\log L - \left(L \log r\right) \sum_{k=1}^{\infty} \Lambda(k) k r^k \\ &\leq -\log L - \left(L \log r\right) \sum_{k=1}^{\infty} k q^k \qquad (13) \\ &= -\log L + \frac{L \log r}{(1-q)^2} \log \frac{1}{q} \qquad (14) \end{aligned}$$

where Equation (13) holds from the assumptions about p. Putting Equation (12) and Equation (14) together, we obtain the result. ■

We note that another common way to quantify the noise in a distribution is through the notion of Tsybakov noise (Tsybakov 2004; Koltchinskii 2006). We discuss this further in Section 7.1, where we show that Tsybakov noise is too permissive, and probabilistic grammars do not satisfy its conditions.

3.2 Limiting the Degree of the Grammar

When approximating a family of probabilistic grammars, it is much more convenient when the degree of the grammar is limited. In this article, we limit the degree of the grammar by making the assumption that all $N_k \leq 2$. This assumption may seem, at first glance, somewhat restrictive, but we show next that for PCFGs (and as a consequence, other formalisms), this assumption does not limit the total generative capacity that we can have across all context-free grammars.

We first show that any context-free grammar with arbitrary degree can be mapped to a corresponding grammar with all $N_k \leq 2$ that generates derivations equivalent to derivations in the original grammar. Such a grammar is also called a "covering grammar" (Nijholt 1980; Leermakers 1989). Let G be a CFG. Let A be the kth nonterminal. Consider the rules $A \to \alpha_i$ for $i \leq N_k$ where A appears on the left side. For each rule

Context-free grammar	**Binarized grammar**	
S → NP VP	S → NP VP | S1	DET → *a* | *the*
S → NP VP NP	S1 → NP VP NP | S2	P → *at* | P1
S → NP VP PP	S2 → NP VP PP	P1 → *on* | P2
PP → P NP	PP → P NP	P2 → *in*
NP → N | DET N	NP → N | DET N	V → *watch*
VP → V	VP → V	
N → *park* | *boy* | *girl* | *I*	N → *park* | N1	
DET → *a* | *the*	N1 → *boy* | N2	
P → *at* | *on* | *in*	N2 → *girl* | N3	
V → *watch*	N3 → *I*	

Figure 2
Example of a context-free grammar and its equivalent binarized form.

$A \to \alpha_i$, $i < N_k$, we create a new nonterminal in G' such that A_i has two rewrite rules: $A_i \to \alpha_i$ and $A_i \to A_{i+1}$. In addition, we create rules $A \to A_1$ and $A_{N_k} \to \alpha_{N_k}$. Figure 2 demonstrates an example of this transformation on a small context-free grammar.

It is easy to verify that the resulting grammar G' has an equivalent capacity to the original CFG, G. A simple transformation that converts each derivation in the new grammar to a derivation in the old grammar would involve collapsing any path of nonterminals added to G' (i.e., all A_i for nonterminal A) so that we end up with nonterminals from the original grammar only. Similarly, any derivation in G can be converted to a derivation in G' by adding new nonterminals through unary application of rules of the form $A_i \to A_{i+1}$. Given a derivation z in G, we denote by $\Upsilon_{G \mapsto G'}(z)$ the corresponding derivation in G' after adding the new non-terminals A_i to z. Throughout this article, we will refer to the normalized form of G' as a "binary normal form."[6]

Note that K', the number of multinomials in the binary normal form, is a function of both the number of nonterminals in the original grammar and the number of rules in that grammar. More specifically, we have that $K' = \sum_{k=1}^{K} N_k + K$. To make the equivalence complete, we need to show that any *probabilistic* context-free grammar can be translated to a PCFG with $\max_k N_k \leq 2$ such that the two PCFGs induce the same equivalent distributions over derivations.

Utility Lemma 1
Let $a_i \in [0,1]$, $i \in \{1,\ldots,N\}$ such that $\sum_i a_i = 1$. Define $b_1 = a_1$, $c_1 = 1 - a_1$, $b_i = \left(\frac{a_i}{a_{i-1}}\right)\left(\frac{b_{i-1}}{c_{i-1}}\right)$, and $c_i = 1 - b_i$ for $i \geq 2$. Then $a_i = \left(\prod_{j=1}^{i-1} c_j\right) b_i$.

See Appendix A for the proof of Utility Lemma 1.

Theorem 1
Let $\langle G, \theta \rangle$ be a probabilistic context-free grammar. Let G' be the binarizing transformation of G as defined earlier. Then, there exists θ' for G' such that for any $z \in D(G)$ we have $p(z \mid \theta, G) = p(\Upsilon_{G \mapsto G'}(z) \mid \theta', G')$.

6 We note that this notion of binarization is different from previous types of binarization appearing in computational linguistics for grammars. Typically in previous work about binarized grammars such as CFGs, the grammars are constrained to have at most two nonterminals in the right side in Chomsky normal form. Another form of binarization for linear context-free rewriting systems is restriction of the *fan-out* of the rules to two (Gómez-Rodríguez and Satta 2009; Gildea 2010). We, however, limit the number of *rules* for each nonterminal (or more generally, the number of elements in each multinomial).

Proof
For the grammar G, index the set $\{1, ..., K\}$ with nonterminals ranging from A_1 to A_K. Define G' as before. We need to define θ'. Index the multinomials in G' by (k, i), each having two events. Let $\mu_{(k,i),1} = \theta_{k,i}$, $\mu_{(k,i),2} = 1 - \theta_{k,i}$ for $i = 1$ and set $\mu_{k,i,1} = \theta_{k,i}/\mu_{(k,i-1),2}$, and $\mu_{(k,i-1),2} = 1 - \mu_{(k,i-1),2}$.

$\langle G', \mu \rangle$ is a *weighted* context-free grammar such that the $\mu_{(k,i),1}$ corresponds to the ith event in the k multinomial of the original grammar. Let z be a derivation in G and $z' = \Upsilon_{G \mapsto G'}(z)$. Then, from Utility Lemma 1 and the construction of g', we have that:

$$
\begin{aligned}
p(z \mid \theta, G) &= \prod_{k=1}^{K} \prod_{i=1}^{N_k} \theta_{k,i}^{\psi_{k,i}(z)} \\
&= \prod_{k=1}^{K} \prod_{i=1}^{N_k} \prod_{l=1}^{\psi_{k,i}(z)} \theta_{k,i} \\
&= \prod_{k=1}^{K} \prod_{i=1}^{N_k} \prod_{l=1}^{\psi_{k,i}(z)} \left(\prod_{j=1}^{i-1} \mu_{(k,j),2} \right) \mu_{(k,i),1} \\
&= \prod_{k=1}^{K} \prod_{i=1}^{N_k} \left(\prod_{j=1}^{i-1} \mu_{(k,j),2}^{\psi_{k,i}(z)} \right) \mu_{(k,i),1}^{\psi_{k,i}(z)} \\
&= \prod_{k=1}^{K} \prod_{j=1}^{N_k} \prod_{i=1}^{2} \mu_{(k,j),i}^{\psi_{k,j}(z')} \\
&= p(z' \mid \mu, G')
\end{aligned}
$$

From Chi (1999), we know that the weighted grammar $\langle G', \mu \rangle$ can be converted to a probabilistic context-free grammar $\langle G', \theta' \rangle$, through a construction of θ' based on μ, such that $p(z' \mid \mu, G') = p(z' \mid \theta', G')$. ■

The proof for Theorem 1 gives a construction the parameters θ' of G' such that $\langle G, \theta \rangle$ is equivalent to $\langle G', \theta' \rangle$. The construction of θ' can also be reversed: Given θ' for G', we can construct θ for G so that again we have equivalence between $\langle G, \theta \rangle$ and $\langle G', \theta' \rangle$.

In this section, we focused on presenting parametrized, empirically justified distributional assumptions about language data that will make the analysis in later sections more manageable. We showed that these assumptions bound the amount of entropy as a function of the assumption parameters. We also made an assumption about the *structure* of the grammar family, and showed that it entails no loss of generality for CFGs. Many other formalisms can follow similar arguments to show that the structural assumption is justified for them as well.

4. Proper Approximations

In order to follow the empirical risk minimization described in Section 2.1, we have to define a series of approximations for $\mathcal{F}$, which we denote by the log-concept spaces $\mathcal{F}_1, \mathcal{F}_2, \ldots$. We also have to replace two-sided uniform convergence (Equation [6]) with convergence on the sequence of concept spaces we defined (Equation [10]). The concept spaces in the sequence vary as a function of the number of samples we have. We next

construct the sequence of concept spaces, and in Section 5 we return to the learning model. Our approximations are based on the concept of *bounded approximations* (Abe, Takeuchi, and Warmuth 1991; Dasgupta 1997), which were originally designed for graphical models.[7] A bounded approximation is a subset of a concept space which is controlled by a parameter that determines its tightness. Here we use this idea to define a series of subsets of the original concept space $\mathcal{F}$ as approximations, while having two asymptotic properties that control the series' tightness.

Let $\mathcal{F}_m$ (for $m \in \{1, 2, \ldots\}$) be a sequence of concept spaces. We consider three properties of elements of this sequence, which should hold for $m > M$ for a fixed M.

The first is **containment** in $\mathcal{F}$:

$$\mathcal{F}_m \subseteq \mathcal{F}$$

The second property is **boundedness**:

$$\exists K_m \geq 0, \forall f \in \mathcal{F}_m, \ \mathbb{E}\left[|f| \times \mathbb{I}\{|f| \geq K_m\}\right] \leq \epsilon_{\text{bound}}(m)$$

where ϵ_{bound} is a non-increasing function such that $\epsilon_{\text{bound}}(m) \underset{m\to\infty}{\longrightarrow} 0$. This states that the expected values of functions from $\mathcal{F}_m$ on values larger than some K_m is small. This is required to obtain uniform convergence results in the revised empirical risk minimization model from Section 2.1. Note that K_m can grow arbitrarily large.

The third property is **tightness**:

$$\exists C_m \in \mathcal{F} \to \mathcal{F}_m, \ p\left(\bigcup_{f\in\mathcal{F}} \{z \mid C_m(f)(z) - f(z) \geq \epsilon_{\text{tail}}(m)\}\right) \leq \epsilon_{\text{tail}}(m)$$

where ϵ_{tail} is a non-increasing function such that $\epsilon_{\text{tail}}(m) \underset{m\to\infty}{\longrightarrow} 0$, and C_m denotes an operator that maps functions in $\mathcal{F}$ to $\mathcal{F}_m$. This ensures that our approximation actually converges to the original concept space $\mathcal{F}$. We will show in Section 4.3 that this is actually a well-motivated characterization of convergence for probabilistic grammars in the supervised setting.

We say that the sequence $\mathcal{F}_m$ *properly approximates* $\mathcal{F}$ if there exist $\epsilon_{\text{tail}}(m)$, $\epsilon_{\text{bound}}(m)$, and C_m such that, for all m larger than some M, containment, boundedness, and tightness all hold.

In a good approximation, K_m would increase at a fast rate as a function of m and $\epsilon_{\text{tail}}(m)$ and $\epsilon_{\text{bound}}(m)$ decrease quickly as a function of m. As we will see in Section 5, we cannot have an arbitrarily fast convergence rate (by, for example, taking a subsequence of $\mathcal{F}_m$), because the size of K_m has a great effect on the number of samples required to obtain accurate estimation.

7 There are other ways to manage the unboundedness of KL divergence in the language learning literature. Clark and Thollard (2004), for example, decompose the KL divergence between probabilistic finite-state automata into several terms according to a decomposition of Carrasco (1997) and then bound each term separately.

Table 1
Example of a PCFG where there is more than a single way to approximate it by truncation with $\gamma = 0.1$, because it has more than two rules. Any value of $\eta \in [0, \gamma]$ will lead to a different approximation.

Rule	θ	General	$\eta = 0$	$\eta = 0.01$	$\eta = 0.005$
$S \to NP\ VP$	0.09	0.01	0.1	0.1	0.1
$S \to NP$	0.11	$0.11 - \eta$	0.11	0.1	0.105
$S \to VP$	0.8	$0.8 - \gamma + \eta$	0.79	0.8	0.795

4.1 Constructing Proper Approximations for Probabilistic Grammars

We now focus on constructing proper approximations for probabilistic grammars whose degree is limited to 2. Proper approximations could, in principle, be used with losses other than the log-loss, though their main use is for unbounded losses. Starting from this point in the article, we focus on using such proper approximations with the log-loss.

We construct $\mathcal{F}_m$. For each $f \in \mathcal{F}$ we define a transformation $T(f, \gamma)$ that shifts every binomial parameter $\theta_k = \langle \theta_{k,1}, \theta_{k,2} \rangle$ in the probabilistic grammar by at most γ:

$$\langle \theta_{k,1}, \theta_{k,2} \rangle \leftarrow \begin{cases} \langle \gamma, & 1-\gamma \rangle & \text{if } \theta_{k,1} < \gamma \\ \langle 1-\gamma, & \gamma \rangle & \text{if } \theta_{k,1} > 1-\gamma \\ \langle \theta_{k,1}, & \theta_{k,2} \rangle & \text{otherwise} \end{cases}$$

Note that $T(f, \gamma) \in \mathcal{F}$ for any $\gamma \leq 1/2$. Fix a constant $s > 1$.[8] We denote by $T(\theta, \gamma)$ the same transformation on θ (which outputs the new shifted parameters) and we denote by $\Theta_G(\gamma) = \Theta(\gamma)$ the set $\{T(\theta, \gamma) \mid \theta \in \Theta_G\}$. For each $m \in \mathbb{N}$, define $\mathcal{F}_m = \{T(f, m^{-s}) \mid f \in \mathcal{F}\}$.

When considering our approach to approximate a probabilistic grammar by increasing its parameter probabilities to be over a certain threshold, it becomes clear why we are required to limit the grammar to have only two rules and why we are required to use the normal from Section 3.2 with grammars of degree 2. Consider the PCFG rules in Table 1. There are different ways to move probability mass to the rule with small probability. This leads to a problem with identifability of the approximation: How does one decide how to reallocate probability to the small probability rules? By binarizing the grammar in advance, we arrive at a single way to reallocate mass when required (i.e., move mass from the high-probability rule to the low-probability rule). This leads to a simpler proof for sample complexity bounds and a single bound (rather than different bounds depending on different smoothing operators). We note, however, that the choices made in binarizing the grammar imply a particular way of smoothing the probability across the original rules.

We now describe how this construction of approximations satisfies the properties mentioned in Section 4, specifically, the boundedness property and the tightness property.

8 By varying s we get a family of approximations. The larger s is, the tighter the approximation is. Also, the larger s is, as we see later, the looser our sample complexity bound will be.

Proposition 2
Let $p \in \mathcal{P}(\alpha, L, r, q, B, G)$ and let $\mathcal{F}_m$ be as defined earlier. There exists a constant $\beta = \beta(L, q, p, N) > 0$ such that $\mathcal{F}_m$ has the boundedness property with $K_m = sN \log^3 m$ and $\epsilon_{\text{bound}}(m) = m^{-\beta \log m}$.

See Appendix A for the proof of Proposition 2.

Next, $\mathcal{F}_m$ is tight with respect to $\mathcal{F}$ with $\epsilon_{\text{tail}}(m) = \frac{N \log^2 m}{m^s - 1}$.

Proposition 3
Let $p \in \mathcal{P}(\alpha, L, r, q, B, G)$ and let $\mathcal{F}_m$ as defined earlier. There exists an M such that for any $m > M$ we have

$$p\left(\bigcup_{f \in \mathcal{F}} \{z \mid C_m(f)(z) - f(z) \geq \epsilon_{\text{tail}}(m)\}\right) \leq \epsilon_{\text{tail}}(m)$$

for $\epsilon_{\text{tail}}(m) = \frac{N \log^2 m}{m^s - 1}$ and $C_m(f) = T(f, m^{-s})$.

See Appendix A for the proof of Proposition 3.

We now have proper approximations for probabilistic grammars. These approximations are defined as a series of probabilistic grammars, related to the family of probabilistic grammars we are interested in estimating. They consist of three properties: containment (they are a subset of the family of probabilistic grammars we are interested in estimating), boundedness (their log-loss does not diverge to infinity quickly), and they are tight (there is a small probability mass at which they are not tight approximations).

4.2 Coupling Bounded Approximations with Number of Samples

At this point, the number of samples n is decoupled from the bounded approximation ($\mathcal{F}_m$) that we choose for grammar estimation. To couple between these two, we need to define m as a function of the number of samples, $m(n)$. As mentioned earlier, there is a clear trade-off between choosing a fast rate for $m(n)$ (such as $m(n) = n^k$ for some $k > 1$) and a slower rate (such as $m(n) = \log n$). The faster the rate is, the tighter the family of approximations that we use for n samples. If the rate is too fast, however, then K_m grows quickly as well. In that case, because our sample complexity bounds are increasing functions of such K_m, the bounds will degrade.

To balance the trade-off, we choose $m(n) = n$. As we see later, this gives sample complexity bounds which are asymptotically interesting for both the supervised and unsupervised case.

4.3 Asymptotic Empirical Risk Minimization

It would be compelling to determine whether the empirical risk minimizer over $\mathcal{F}_n$ is an *asymptotic empirical risk minimizer*. This would mean that the risk of the empirical risk minimizer over $\mathcal{F}_n$ converges to the risk of the maximum likelihood estimate. As a conclusion to this section about proper approximations, we motivate the three requirements that we posed on proper approximations by showing that this is indeed true. We now unify n, the number of samples, and m, the index of the approximation of the concept space $\mathcal{F}$. Let f_n^* be the minimizer of the empirical risk over $\mathcal{F}$, ($f_n^* = \text{argmin}_{f\in\mathcal{F}} \mathbb{E}_{\tilde{p}_n}\left[f\right]$) and let g_n be the minimizer of the empirical risk over $\mathcal{F}_n$ ($g_n = \text{argmin}_{f\in\mathcal{F}_n} \mathbb{E}_{\tilde{p}_n}\left[f\right]$).

Let $D = \{z_1, ..., z_n\}$ be a sample from $p(z)$. The operator ($g_n =$) $\text{argmin}_{f\in\mathcal{F}_n} \mathbb{E}_{\tilde{p}_n}[f]$ is an asymptotic empirical risk minimizer if $\mathbb{E}\left[\mathbb{E}_{\tilde{p}_n}\left[g_n\right] - \mathbb{E}_{\tilde{p}_n}[f_n^*]\right] \to 0$ as $n \to \infty$ (Shalev-Shwartz et al. 2009). Then, we have the following

Lemma 1
Denote by $\mathcal{Z}_{\epsilon,n}$ the set $\bigcup_{f\in\mathcal{F}}\{z \mid C_n(f)(z) - f(z) \geq \epsilon\}$. Denote by $A_{\epsilon,n}$ the event "one of $z_i \in D$ is in $\mathcal{Z}_{\epsilon,n}$." If $\mathcal{F}_n$ properly approximates $\mathcal{F}$, then:

$$\begin{aligned} &\mathbb{E}\left[\mathbb{E}_{\tilde{p}_n}\left[g_n\right] - \mathbb{E}_{\tilde{p}_n}\left[f_n^*\right]\right] \\ &\leq \left|\mathbb{E}\left[\mathbb{E}_{\tilde{p}_n}\left[C_n(f_n^*)\right] \mid A_{\epsilon,n}\right]\right| p(A_{\epsilon,n}) + \left|\mathbb{E}\left[\mathbb{E}_{\tilde{p}_n}\left[f_n^*\right] \mid A_{\epsilon,n}\right]\right| p(A_{\epsilon,n}) + \epsilon_{\text{tail}}(n) \end{aligned} \tag{15}$$

where the expectations are taken with respect to the data set D.

See Appendix A for the proof of Lemma 1.

Proposition 4
Let $D = \{z_1, ..., z_n\}$ be a sample of derivations from G. Then $g_n = \text{argmin}_{f\in\mathcal{F}_n} \mathbb{E}_{\tilde{p}_n}\left[f\right]$ is an asymptotic empirical risk minimizer.

Proof
Let $f_0 \in \mathcal{F}$ be the concept that puts uniform weights over θ, namely, $\theta_k = \langle \frac{1}{2}, \frac{1}{2} \rangle$ for all k. Note that

$$\begin{aligned} &|\mathbb{E}\left[\mathbb{E}_{\tilde{p}_n}\left[f_n^*\right] \mid A_{\epsilon,n}\right]| p(A_{\epsilon,n}) \\ &\leq |\mathbb{E}\left[\mathbb{E}_{\tilde{p}_n}\left[f_0\right] \mid A_{\epsilon,n}\right]| p(A_{\epsilon,n}) = \tfrac{\log 2}{n} \textstyle\sum_{l=1}^{n} \sum_{k,i} \mathbb{E}[\psi_{k,i}(z_l) \mid A_{\epsilon,n}] p(A_{\epsilon,n}) \end{aligned}$$

Let $A_{j,\epsilon,n}$ for $j \in \{1,\ldots,n\}$ be the event "$z_j \in \mathcal{Z}_{\epsilon,n}$". Then $A_{\epsilon,n} = \bigcup_j A_{j,\epsilon,n}$. We have that

$$\begin{aligned}
&\mathbb{E}[\psi_{k,i}(z_l) \mid A_{\epsilon,n}]p(A_{\epsilon,n}) \leq \sum_j \sum_{z_l} p(z_l, A_{j,\epsilon,n})|z_l| \\
&\leq \sum_{j\neq l} \sum_{z_l} p(z_l)p(A_{j,\epsilon,n})|z_l| + \sum_{z_l} p(z_l, A_{l,\epsilon,n})|z_l| \qquad (16) \\
&\leq \left(\sum_{j\neq l} p(A_{j,\epsilon,n})\right) B + \mathbb{E}[\psi_{k,i}(z) \mid z \in \mathcal{Z}_{\epsilon,n}]p(z \in \mathcal{Z}_{\epsilon,n}) \\
&\leq (n-1)Bp(z \in \mathcal{Z}_{\epsilon,n}) + \mathbb{E}[\psi_{k,i}(z) \mid z \in \mathcal{Z}_{\epsilon,n}]p(z \in \mathcal{Z}_{\epsilon,n})
\end{aligned}$$

where Equation (16) comes from z_l being independent. Also, B is the constant from Section 3.1. Therefore, we have:

$$\begin{aligned}
&\frac{1}{n}\sum_{l=1}^{n}\sum_{k,i} \mathbb{E}[\psi_{k,i}(z_l) \mid A_{\epsilon,n}]p(A_{\epsilon,n}) \\
&\leq \sum_{k,i} \left(\mathbb{E}[\psi_{k,i}(z) \mid z \in \mathcal{Z}_{\epsilon,n}]p(z \in \mathcal{Z}_{\epsilon,n}) + (n-1)Bp(z \in \mathcal{Z}_{\epsilon,n})\right)
\end{aligned}$$

From the construction of our proper approximations (Proposition 3), we know that only derivations of length $\log^2 n$ or greater can be in $\mathcal{Z}_{\epsilon,n}$. Therefore

$$\mathbb{E}[\psi_{k,i} \mid \mathcal{Z}_{\epsilon,n}]p(\mathcal{Z}_{\epsilon,n}) \leq \sum_{z:|z|>\log^2 n} p(z)\psi_{k,i}(z) \leq \sum_{l>\log^2 n}^{\infty} L\Lambda(l)r^l l \leq \kappa q^{\log^2 n} = o(1)$$

where $\kappa > 0$ is a constant. Similarly, we have $p(z \in \mathcal{Z}_{\epsilon,n}) = o(n^{-1})$. This means that $|\mathbb{E}[\mathbb{E}_{\tilde{p}_n}[-\log -f_n^*] \mid A_{\epsilon,n}]|p(A_{\epsilon,n}) \underset{n\to\infty}{\longrightarrow} 0$. In addition, it can be shown that $|\mathbb{E}[\mathbb{E}_{\tilde{p}_n}[C_n(f_n^*) \mid A_{\epsilon,n}]|p(A_{\epsilon,n}) \underset{n\to\infty}{\longrightarrow} 0$ using the same proof technique we used here, while relying on the fact that $C_n(f_n^*) \in \mathcal{F}_n$, and therefore $C_n(f_n^*)(z) \leq sN|z|\log n$. ■

5. Sample Complexity Bounds

Equipped with the framework of proper approximations as described previously, we now give our main sample complexity results for probabilistic grammars. These results hinge on the convergence of $\sup_{f\in\mathcal{F}_n} |\mathbb{E}_{\tilde{p}_n}\left[f\right] - \mathbb{E}_p\left[f\right]|$. Indeed, proper approximations replace the use of $\mathcal{F}$ in these convergence results. The rate of this convergence can be fast, if the *covering numbers* for $\mathcal{F}_n$ do not grow too fast.

5.1 Covering Numbers and Bounds on Covering Numbers

We next give a brief overview of covering numbers. A cover provides a way to reduce a class of functions to a much smaller (finite, in fact) representative class such that each function in the original class is represented using a function in the smaller class. Let $\mathcal{G}$

be a class of functions. Let $d(f,g)$ be a distance measure between two functions f,g from $\mathcal{G}$. An ϵ-cover is a subset of $\mathcal{G}$, denoted by $\mathcal{G}'$, such that for every $f \in \mathcal{G}$ there exists an $f' \in \mathcal{G}'$ such that $d(f,f') < \epsilon$. The **covering number** $\mathcal{N}(\epsilon, \mathcal{G}, d)$ is the size of the smallest ϵ-cover of $\mathcal{G}$ for the distance measure d.

We are interested in a specific distance measure which is dependent on the empirical distribution $\tilde{p}_n$ that describes the data $z_1, ..., z_n$. Let $f,g \in \mathcal{G}$. We will use

$$d^{\tilde{p}_n}(f,g) = \mathbb{E}_{\tilde{p}_n}\left[|f-g|\right] = \sum\nolimits_{z \in D(G)} |f(z) - g(z)|\, \tilde{p}_n(z)$$
$$= \frac{1}{n} \sum\nolimits_{i=1}^{n} |f(z_i) - g(z_i)|$$

Instead of using $\mathcal{N}(\epsilon, \mathcal{G}, d^{\tilde{p}_n})$ directly, we bound this quantity with $\mathcal{N}(\epsilon, \mathcal{G}) = \sup_{\tilde{p}_n} \mathcal{N}(\epsilon, \mathcal{G}, d^{\tilde{p}_n})$, where we consider all possible samples (yielding $\tilde{p}_n$). The following is the key result regarding the connection between covering numbers and the double-sided convergence of the empirical process $\sup_{f \in \mathcal{F}_n} |\mathbb{E}_{\tilde{p}_n}\left[f\right] - \mathbb{E}_p\left[f\right]|$ as $n \to \infty$. This result is a general-purpose result that has been used frequently to prove the convergence of empirical processes of the type we discuss in this article.

Lemma 2
Let $\mathcal{F}_n$ be a permissible class[9] of functions such that for every $f \in \mathcal{F}_n$ we have $\mathbb{E}[|f| \times \mathbb{I}\{|f| \leq K_n\}] \leq \epsilon_{\text{bound}}(n)$. Let $\mathcal{F}_{\text{truncated},n} = \{f \times \mathbb{I}\{f \leq K_n\} \mid f \in \mathcal{F}_m\}$, namely, the set of functions from $\mathcal{F}_n$ after being truncated by K_n. Then for $\epsilon > 0$ we have

$$p\left(\sup_{f \in \mathcal{F}_n} |\mathbb{E}_{\tilde{p}_n}\left[f\right] - \mathbb{E}_p\left[f\right]| > 2\epsilon\right) \leq 8\mathcal{N}(\epsilon/8, \mathcal{F}_{\text{truncated},n}) \exp\left(-\frac{1}{128} n\epsilon^2/K_n^2\right) + \epsilon_{\text{bound}}(n)/\epsilon$$

provided $n \geq K_n^2/4\epsilon^2$ and $\epsilon_{\text{bound}}(n) < \epsilon$.

See Pollard (1984; Chapter 2, pages 30–31) for the proof of Lemma 2. See also Appendix A.

Covering numbers are rather complex combinatorial quantities which are hard to compute directly. Fortunately, they can be bounded using the pseudo-dimension (Anthony and Bartlett 1999), a generalization of the Vapnik-Chervonenkis (VC) dimension for real functions. In the case of our "binomialized" probabilistic grammars, the pseudo-dimension of $\mathcal{F}_n$ is bounded by N, because we have $\mathcal{F}_n \subseteq \mathcal{F}$, and the functions in $\mathcal{F}$ are linear with N parameters. Hence, $\mathcal{F}_{\text{truncated},n}$ also has pseudo-dimension that is at most N. We then have the following.

9 The "permissible class" requirement is a mild regularity condition regarding measurability that holds for proper approximations. We refer the reader to Pollard (1984) for more details.

Lemma 3
(From Pollard [1984] and Haussler [1992].) Let $\mathcal{F}_n$ be the proper approximations for probabilistic grammars, for any $0 < \epsilon < K_n$ we have:

$$\mathcal{N}(\epsilon, \mathcal{F}_{\text{truncated},n}) < 2\left(\frac{2eK_n}{\epsilon}\log\frac{2eK_n}{\epsilon}\right)^N$$

5.2 Supervised Case

We turn to give an analysis for the supervised case. This analysis is mostly described as a preparation for the unsupervised case. In general, the families of probabilistic grammars we give a treatment to are parametric families, and the maximum likelihood estimator for these families is a consistent estimator in the supervised case. In the unsupervised case, however, lack of identifiability prevents us from getting these traditional consistency results. Also, the traditional results about the consistency of MLE are based on the assumption that the sample is generated from the parametric family we are trying to estimate. This is not the case in our analysis, where the distribution that generates the data does not have to be a probabilistic grammar.

Lemmas 2 and 3 can be combined to get the following sample complexity result.

Theorem 2
Let G be a grammar. Let $p \in \mathcal{P}(\alpha, L, r, q, B, G)$ (Section 3.1). Let $\mathcal{F}_n$ be a proper approximation for the corresponding family of probabilistic grammars. Let $z_1, \ldots, z_n$ be a sample of derivations. Then there exists a constant $\beta(L, q, p, N)$ and constant M such that for any $0 < \delta < 1$ and $0 < \epsilon < K_n$ and any $n > M$ and if

$$n \geq \max\left\{\frac{128K_n^2}{\epsilon^2}\left(2N\log(16eK_n/\epsilon) + \log\frac{32}{\delta}\right), \frac{\log 4/\delta + \log 1/\epsilon}{\beta(L, q, p, N)}\right\}$$

then we have

$$P\left(\sup_{f\in\mathcal{F}_n} |\mathbb{E}_{\tilde{p}_n}\left[f\right] - \mathbb{E}_p\left[f\right]| \leq 2\epsilon\right) \geq 1 - \delta$$

where $K_n = sN\log^3 n$.

Proof Sketch
$\beta(L, q, p, N)$ is the constant from Proposition 2. The main idea in the proof is to solve for n in the following two inequalities (based on Equation [17] [see the following]) while relying on Lemma 3:

$$8\mathcal{N}(\epsilon/8, \mathcal{F}_{\text{truncated},n})\exp\left(-\frac{1}{128}n\epsilon^2/K_n^2\right) \leq \delta/2$$
$$\epsilon_{\text{bound}}(n)/\epsilon \leq \delta/2$$

■

Theorem 2 gives little intuition about the number of samples required for accurate estimation of a grammar because it considers the "additive" setting: The empirical risk is within ϵ from the expected risk. More specifically, it is not clear how we should pick ϵ for the log-loss, because the log-loss can obtain arbitrary values.

We turn now to converting the additive bound in Theorem 2 to a multiplicative bound. Multiplicative bounds can be more informative than additive bounds when the range of the values that the log-loss can obtain is not known a priori. It is important to note that the two views are equivalent (i.e., it is possible to convert a multiplicative bound to an additive bound and vice versa). Let $\rho \in (0,1)$ and choose $\epsilon = \rho K_n$. Then, substituting this ϵ in Theorem 2, we get that if

$$n \geq \max\left\{\frac{128}{\rho^2}\left(2N\log\frac{16e}{\rho} + \log\frac{32}{\delta}\right), \frac{\log 4/\delta + \log 1/\rho}{\beta(L,q,p,N)}\right\}$$

then, with probability $1-\delta$,

$$\sup_{f\in\mathcal{F}_n}\left|1 - \frac{\mathbb{E}_{\tilde{p}_n}\left[f\right]}{\mathbb{E}_p\left[f\right]}\right| \leq \frac{\rho \times 2sN\log^3(n)}{H(p)} \tag{17}$$

where $H(p)$ is the Shannon entropy of p. This stems from the fact that $\mathbb{E}_p\left[f\right] \geq H(p)$ for any f. This means that if we are interested in computing a sample complexity bound such that the ratio between the empirical risk and the expected risk (for log-loss) is close to 1 with high probability, we need to pick up ρ such that the righthand side of Equation (17) is smaller than the desired accuracy level (between 0 and 1). Note that Equation (17) is an oracle inequality—it requires knowing the entropy of p or some upper bound on it.

5.3 Unsupervised Case

In the unsupervised setting, we have n **yields** of derivations from the grammar, $x_1, ..., x_n$, and our goal again is to identify grammar parameters θ from these yields. Our concept classes are now the sets of log marginalized distributions from $\mathcal{F}_n$. For each $f_\theta \in \mathcal{F}_n$, we define f'_θ as

$$f'_\theta(x) = -\log\sum_{z\in D_x(G)}\exp(-f_\theta(z)) = -\log\sum_{z\in D_x(G)}\exp\left(\sum_{k=1}^{K}\sum_{i=1}^{N_k}\psi_{i,k}(z)\theta_{i,k}\right)$$

We denote the set of $\{f'_\theta\}$ by $\mathcal{F}'_n$. Analogously, we define $\mathcal{F}'$. Note that we also need to define the operator $C'_n(f')$ as a first step towards defining $\mathcal{F}'_n$ as proper approximations (for $\mathcal{F}'$) in the unsupervised setting. Let $f' \in \mathcal{F}'$. Let f be the concept in $\mathcal{F}$ such that $f'(x) = \sum_z f(x,z)$. Then we define $C'_n(f')(x) = \sum_z C_n(f)(x,z)$.

It does not immediately follow that $\mathcal{F}'_n$ is a proper approximation for $\mathcal{F}'$. It is not hard to show that the boundedness property is satisfied with the same K_n and the same form of $\epsilon_{\text{bound}}(n)$ as in Proposition 2 (we would have $\epsilon'_{\text{bound}}(m) = m^{-\beta'\log m}$ for some $\beta'(L,q,p,N) = \beta' > 0$). This relies on the property of bounded derivation length of p (see Appendix A, Proposition 7). The following result shows that we have tightness as well.

Utility Lemma 2
For $a_i, b_i \geq 0$, if $-\log \sum_i a_i + \log \sum_i b_i \geq \epsilon$ then there exists an i such that $-\log a_i + \log b_i \geq \epsilon$.

Proposition 5
There exists an M such that for any $n > M$ we have

$$p\left(\bigcup_{f' \in \mathcal{F}'} \{x \mid C'_n(f')(x) - f'(x) \geq \epsilon_{\text{tail}}(n)\}\right) \leq \epsilon_{\text{tail}}(n)$$

for $\epsilon_{\text{tail}}(n) = \dfrac{N \log^2 n}{n^s - 1}$ and the operator $C'_n(f)$ as defined earlier.

Proof Sketch
From Utility Lemma 2 we have

$$p\left(\bigcup_{f' \in \mathcal{F}'} \{x \mid C'_n(f')(x) - f'(x) \geq \epsilon_{\text{tail}}(n)\}\right) \leq p\left(\bigcup_{f \in \mathcal{F}} \{x \mid \exists z C_n(f)(z) - f(z) \geq \epsilon_{\text{tail}}(n)\}\right)$$

Define $\mathcal{X}(n)$ to be all x such that there exists a z with $\text{yield}(z) = x$ and $|z| \geq \log^2 n$. From the proof of Proposition 3 and the requirements on p, we know that there exists an $\alpha \geq 1$ such that

$$\begin{aligned} p\left(\textstyle\bigcup_{f \in \mathcal{F}} \{x \mid \exists z \text{ s.t. } C_n(f)(z) - f(z) \geq \epsilon_{\text{tail}}(n)\}\right) &\leq \sum_{x \in \mathcal{X}(n)} p(x) \\ \leq \sum_{x : |x| \geq \log^2 n/\alpha} p(x) \leq \sum_{k = \lfloor \log^2 n/\alpha \rfloor}^{\infty} L\Lambda(k) r^k &\leq \epsilon_{\text{tail}}(n) \end{aligned}$$

where the last inequality happens for some n larger than a fixed M. ■

Computing either the covering number or the pseudo-dimension of $\mathcal{F}'_n$ is a hard task, because the function in the classes includes the "log-sum-exp." Dasgupta (1997) overcomes this problem for Bayesian networks with fixed structure by giving a bound on the covering number for (his respective) $\mathcal{F}'$ which depends on the covering number of $\mathcal{F}$.

Unfortunately, we cannot fully adopt this approach, because the derivations of a probabilistic grammar can be arbitrarily large. Instead, we present the following proposition, which is based on the "Hidden Variable Rule" from Dasgupta (1997). This proposition shows that the covering number of $\mathcal{F}'$ (or more accurately, its bounded approximations) can be bounded in terms of the covering number of the bounded

approximations of $\mathcal{F}$, and the constants which control the underlying distribution p mentioned in Section 3.

Utility Lemma 3
For any two positive-valued sequences $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$ we have that $\sum_i |\log a_i/b_i| \geq |\log(\sum a_i / \sum b_i)|$.

Proposition 6 (Hidden Variable Rule for Probabilistic Grammars)
Let $m = \dfrac{\log \dfrac{4K_n}{\epsilon(1-q)}}{\log \dfrac{1}{q}}$. Then, $\mathcal{N}(\epsilon, \mathcal{F}'_{\text{truncated},n}) \leq \mathcal{N}\left(\dfrac{\epsilon}{2\Lambda(m)}, \mathcal{F}_{\text{truncated},n}\right)$.

Proof
Let $\mathcal{Z}(m) = \{z \mid |z| \leq m\}$ be the subset of derivations of length shorter than m. Consider $f, f_0 \in \mathcal{F}_{\text{truncated},n}$. Let f' and f'_0 be the corresponding functions in $\mathcal{F}'_{\text{truncated},n}$. Then, for any distribution p,

$$
\begin{aligned}
d^p(f', f'_0) &= \sum_x |f'(x) - f'_0(x)|\, p(x) \leq \sum_x \sum_z |f(x,z) - f_0(x,z)|\, p(x) \\
&= \sum_x \sum_{z \in \mathcal{Z}(m)} |f(x,z) - f_0(x,z)|\, p(x) + \sum_x \sum_{z \notin \mathcal{Z}(m)} |f(x,z) - f_0(x,z)|\, p(x) \\
&\leq \sum_x \sum_{z \in \mathcal{Z}(m)} |f(x,z) - f_0(x,z)|\, p(x) + \sum_x \sum_{z \notin \mathcal{Z}(m)} 2K_n p(x) \qquad (18) \\
&\leq \sum_x \sum_{z \in \mathcal{Z}(m)} |f(x,z) - f_0(x,z)|\, p(x) + 2K_n \sum_{x:\, |x| \geq m} |D_x(G)| p(x) \\
&\leq \sum_x \sum_{z \in \mathcal{Z}(m)} |f(x,z) - f_0(x,z)|\, p(x) + 2K_n \sum_{k=m}^{\infty} \Lambda^2(k) r^k \\
&\leq d^{p'}(f, f_0) |\mathcal{Z}(m)| + 2K_n \frac{q^m}{1-q}
\end{aligned}
$$

where $p'(x,z)$ is a probability distribution that uniformly divides the probability mass $p(x)$ across all derivations for the specific x, that is:

$$p'(x,z) = \frac{p(x)}{|D_x(G)|}$$

The inequality in Equation (18) stems from Utility Lemma 3.

Set m to be the quantity that appears in the proposition to get the necessary result (f' and f are arbitrary functions in $\mathcal{F}'_{\text{truncated},n}$ and $\mathcal{F}_{\text{truncated},n}$ respectively. Then consider f'_0 and f_0 to be functions from the respective covers.). ■

For the unsupervised case, then, we get the following sample complexity result.

Theorem 3
Let G be a grammar. Let $\mathcal{F}'_n$ be a proper approximation for the corresponding family of probabilistic grammars. Let $p(x, z)$ be a distribution over derivations which satisfies the requirements in Section 3.1. Let $x_1, \ldots, x_n$ be a sample of strings from $p(x)$. Then there exists a constant $\beta'(L, q, p, N)$ and constant M such that for any $0 < \delta < 1$, $0 < \epsilon < K_n$, any $n > M$, and if

$$n \geq \max \left\{ \frac{128 K_n^2}{\epsilon^2} \left(2N \log \left(\frac{32 e K_n \Lambda(m)}{\epsilon} \right) + \log \frac{32}{\delta} \right), \frac{\log 4/\delta + \log 1/\epsilon}{\beta'(L, q, p, N)} \right\} \tag{19}$$

where $m = \dfrac{\log \dfrac{4K_n}{\epsilon(1-q)}}{\log \dfrac{1}{q}}$, we have that

$$p \left(\sup_{f \in \mathcal{F}'_n} |\mathbb{E}_{\tilde{p}_n} \left[f \right] - \mathbb{E}_p \left[f \right]| \leq 2\epsilon \right) \geq 1 - \delta$$

where $K_n = sN \log^3 n$.

Theorem 3 states that the number of samples we require in order to accurately estimate a probabilistic grammar from unparsed strings depends on the level of ambiguity in the grammar, represented as $\Lambda(m)$. We note that this dependence is polynomial, and we consider this a positive result for unsupervised learning of grammars. More specifically, if Λ is an exponential function (such as the case with PCFGs), when compared to the supervised learning, there is an extra multiplicative factor in the sample complexity in the unsupervised setting that behaves like $\mathcal{O}(\log \log \frac{K_n}{\epsilon})$.

We note that the following Equation (20) can again be reduced to a multiplicative case, similarly to the way we described it for the supervised case. Setting $\epsilon = \rho K_n$ ($\rho \in (0, 1)$), we get the following requirement on n:

$$n \geq \max \left\{ \frac{128}{\rho^2} \left(2N \log \left(\frac{32 e \times t(\rho)}{\rho} \right) + \log \frac{32}{\delta} \right), \frac{\log 4/\delta + \log 1/\epsilon}{\beta'(L, q, p, N)} \right\} \tag{20}$$

where $t(\rho) = \dfrac{\log \dfrac{4}{\rho(1-q)}}{\log \dfrac{1}{q}}$.

6. Algorithms for Empirical Risk Minimization

We turn now to describing algorithms and their properties for minimizing empirical risk using the framework described in Section 4.

6.1 Supervised Case

ERM with proper approximations leads to simple algorithms for estimating the probabilities of a probabilistic grammar in the supervised setting. Given an $\epsilon > 0$ and a $\delta > 0$, we draw n examples according to Theorem 2. We then set $\gamma = n^{-s}$. To minimize the log-loss with respect to these n examples, we use the proper approximation $\mathcal{F}_n$.

Note that the value of the empirical log-loss for a probabilistic grammar parametrized by θ is

$$\begin{aligned}\mathbb{E}_{\tilde{p}_n}\left[-\log h(x,z \mid \theta)\right] &= -\sum_{x,z} \tilde{p}_n(x,z) \log h(x,z \mid \theta) \\ &= -\sum_{x,z} \tilde{p}_n(x,z) \sum_{k=1}^{K} \sum_{i=1}^{N_k} \psi_{k,i}(x,z) \log(\theta_{k,i}) \\ &= -\sum_{k=1}^{K} \sum_{i=1}^{N_k} \log(\theta_{k,i}) \mathbb{E}_{\tilde{p}_n}\left[\psi_{k,i}\right]\end{aligned}$$

Because we make the assumption that $\deg(G) \leq 2$ (Section 3.2), we have

$$\mathbb{E}_{\tilde{p}_n}\left[-\log h(x,z \mid \theta)\right] = -\sum_{k=1}^{K} \left(\log(\theta_{k,1}) \mathbb{E}_{\tilde{p}_n}\left[\psi_{k,1}\right] + \log(1-\theta_{k,1}) \mathbb{E}_{\tilde{p}_n}\left[\psi_{k,2}\right]\right) \quad (21)$$

To minimize the log-loss with respect to $\mathcal{F}_n$, we need to minimize Equation (21) under the constraint that $\gamma \leq \theta_{k,i} \leq 1-\gamma$ and $\theta_{k_1} + \theta_{k,2} = 1$. It can be shown that the solution for this optimization problem is

$$\theta_{k,i} = \min\left\{1-\gamma, \max\left\{\gamma, \left(\sum_{j=1}^{n} \hat{\psi}_{j,k,i}\right) \Big/ \left(\sum_{j=1}^{n} \sum_{i'=1}^{2} \hat{\psi}_{j,k,i'}\right)\right\}\right\} \quad (22)$$

where $\hat{\psi}_{j,k,i}$ is the number of times that $\psi_{k,i}$ fires in Example j. (We include a full derivation of this result in Appendix B.) The interpretation of Equation (22) is simple: We count the number of times a rule appears in the samples and then normalize this value by the total number of times rules associated with the same multinomial appear in the samples. This frequency count is the maximum likelihood solution with respect to the full hypothesis class $\mathcal{H}$ (Corazza and Satta 2006; see Appendix B). Because we constrain ourselves to obtain a value away from 0 or 1 by a margin of γ, we need to truncate this solution, as done in Equation (22).

This truncation to a margin γ can be thought of as a smoothing factor that enables us to compute sample complexity bounds. We explore this connection to smoothing with a Dirichlet prior in a Maximum a posteriori (MAP) Bayesian setting in Section 7.2.

6.2 Unsupervised Case

Similarly to the supervised case, minimizing the empirical log-loss in the unsupervised setting requires minimizing (with respect to θ) the following:

$$\mathbb{E}_{\tilde{p}_n}\left[-\log h(x \mid \theta)\right] = -\sum_{x} \tilde{p}_n(x) \log \sum_{z} h(x,z \mid \theta) \quad (23)$$

with the constraint that $\gamma \leq \theta_{k,i} \leq 1-\gamma$ (i.e., $\theta \in \Theta(\gamma)$) where $\gamma = n^{-s}$. This is done after drawing n examples according to Theorem 3.

6.2.1 Hardness of ERM with Proper Approximations. It turns out that minimizing Equation (23) under the specified constraints is actually an NP-hard problem when G is a PCFG. This result follows using a similar proof to the one in Cohen and Smith (2010c) for the hardness of Viterbi training and maximizing log-likelihood for PCFGs. We turn to giving the full derivation of this hardness result for PCFGs and the modification required for adapting the results from Cohen and Smith to the case of having an arbitrary γ margin constraint.

In order to show an NP-hardness result, we need to "convert" the problem of the maximization of Equation (23) to a decision problem. We do so by stating the following decision problem.

Problem 1 (Unsupervised Minimization of the Log-Loss with Margin)
Input: A binarized context-free grammar G, a set of sentences $x_1, \ldots, x_n$, a value $\gamma \in [0, \frac{1}{2})$, and a value $\alpha \in [0, 1]$.
Output: 1 if there exists $\theta \in \Theta(\gamma)$ (and hence, $h \in \mathcal{H}(G)$) such that

$$-\sum_x \tilde{p}_n(x) \log \sum_z h(x, z \mid \theta) \leq -\log(\alpha) \tag{24}$$

and 0 otherwise.

We will show the hardness result both when γ is not restricted at all as well as when we allow $\gamma > 0$. The proof of the hardness result is achieved by reducing the problem 3-SAT (Sipser 2006), known to be NP-complete, to Problem 1. The problem 3-SAT is defined as follows:

Problem 2 (3-SAT)
Input: A formula $\phi = \bigwedge_{i=1}^{m} (a_i \vee b_i \vee c_i)$ in conjunctive normal form, such that each clause has three literals.
Output: 1 if there is a satisfying assignment for ϕ, and 0 otherwise.

Given an instance of the 3-SAT problem, the reduction will, in polynomial time, create a grammar and a single string such that solving Problem 1 for this grammar and string will yield a solution for the instance of the 3-SAT problem.

Let $\phi = \bigwedge_{i=1}^{m} (a_i \vee b_i \vee c_i)$ be an instance of the 3-SAT problem, where a_i, b_i, and c_i are literals over the set of variables $\{Y_1, \ldots, Y_N\}$ (a literal refers to a variable Y_j or its negation, $\bar{Y}_j$). Let C_j be the jth clause in ϕ, such that $C_j = a_j \vee b_j \vee c_j$. We define the following CFG G_ϕ and string to parse s_ϕ:

1. The terminals of G_ϕ are the binary digits $\Sigma = \{0, 1\}$.
2. We create N nonterminals V_{Y_r}, $r \in \{1, \ldots, N\}$ and rules $V_{Y_r} \to 0$ and $V_{Y_r} \to 1$.
3. We create N nonterminals $V_{\bar{Y}_r}$, $r \in \{1, \ldots, N\}$ and rules $V_{\bar{Y}_r} \to 0$ and $V_{\bar{Y}_r} \to 1$.
4. We create $U_{Y_r,1} \to V_{Y_r} V_{\bar{Y}_r}$ and $U_{Y_r,0} \to V_{\bar{Y}_r} V_{Y_r}$.
5. We create the rule $S_1 \to A_1$. For each $j \in \{2, \ldots, m\}$, we create a rule $S_j \to S_{j-1} A_j$ where S_j is a new nonterminal indexed by $\phi_j \triangleq \bigwedge_{i=1}^{j} C_i$ and A_j is also a new nonterminal indexed by $j \in \{1, \ldots, m\}$.

6. Let $C_j = a_j \vee b_j \vee c_j$ be clause j in ϕ. Let $Y(a_j)$ be the variable that a_j mentions. Let (y_1, y_2, y_3) be a satisfying assignment for C_j where $y_k \in \{0, 1\}$ and is the value of $Y(a_j)$, $Y(b_j)$, and $Y(c_j)$, respectively, for $k \in \{1, 2, 3\}$. For each such clause-satisfying assignment, we add the rule

$$A_j \rightarrow U_{Y(a_j),y_1} U_{Y(b_j),y_2} U_{Y(c_j),y_3}$$

   For each A_j, we would have at most seven rules of this form, because one rule will be logically inconsistent with $a_j \vee b_j \vee c_j$.

7. The grammar's start symbol is S_n.

8. The string to parse is $s_\phi = (10)^{3m}$, that is, $3m$ consecutive occurrences of the string 10.

A parse of the string s_ϕ using G_ϕ will be used to get an assignment by setting $Y_r = 0$ if the rule $V_{Y_r} \rightarrow 0$ or $V_{\bar{Y}_r} \rightarrow 1$ is used in the derivation of the parse tree, and 1 otherwise. Notice that at this point we do not exclude "contradictions" that come from the parse tree, such as $V_{Y_3} \rightarrow 0$ used in the tree together with $V_{Y_3} \rightarrow 1$ or $V_{\bar{Y}_3} \rightarrow 0$. To maintain the restriction on the degree of grammars, we convert G_ϕ to the binary normal form described in Section 3.2. The following lemma gives a condition under which the assignment is consistent (so that contradictions do not occur in the parse tree).

Lemma 4
Let ϕ be an instance of the 3-SAT problem, and let G_ϕ be a probabilistic CFG based on the given grammar with weights θ_ϕ. If the (multiplicative) weight of the Viterbi parse (i.e., the highest scoring parse according to the PCFG) of s_ϕ is 1, then the assignment extracted from the parse tree is consistent.

Proof
Because the probability of the Viterbi parse is 1, all rules of the form $\{V_{Y_r}, V_{\bar{Y}_r}\} \rightarrow \{0, 1\}$ which appear in the parse tree have probability 1 as well. There are two possible types of inconsistencies. We show that neither exists in the Viterbi parse:

1. For any r, an appearance of both rules of the form $V_{Y_r} \rightarrow 0$ and $V_{Y_r} \rightarrow 1$ cannot occur because all rules that appear in the Viterbi parse tree have probability 1.

2. For any r, an appearance of rules of the form $V_{Y_r} \rightarrow 1$ and $V_{\bar{Y}_r} \rightarrow 1$ cannot occur, because whenever we have an appearance of the rule $V_{Y_r} \rightarrow 0$, we have an adjacent appearance of the rule $V_{\bar{Y}_r} \rightarrow 1$ (because we parse substrings of the form 10), and then we again use the fact that all rules in the parse tree have probability 1. The case of $V_{Y_r} \rightarrow 0$ and $V_{\bar{Y}_r} \rightarrow 0$ is handled analogously.

Thus, both possible inconsistencies are ruled out, resulting in a consistent assignment. ■

Figure 3 gives an example of an application of the reduction.

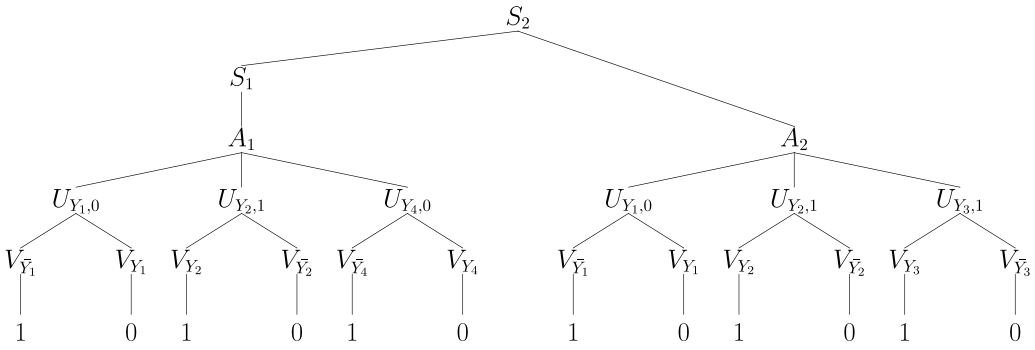


Figure 3
An example of a Viterbi parse tree which represents a satisfying assignment for $\phi = (Y_1 \vee Y_2 \vee \bar{Y}_4) \wedge (\bar{Y}_1 \vee \bar{Y}_2 \vee Y_3)$. In θ_ϕ, all rules appearing in the parse tree have probability 1. The extracted assignment would be $Y_1 = 0, Y_2 = 1, Y_3 = 1, Y_4 = 0$. Note that there is no usage of two different rules for a single nonterminal.

Lemma 5
Define ϕ and G_ϕ as before. There exists θ_ϕ such that the Viterbi parse of s_ϕ is 1 if and only if ϕ is satisfiable. Moreover, the satisfying assignment is the one extracted from the parse tree with weight 1 of s_ϕ under θ_ϕ.

Proof
($\Longrightarrow$) Assume that there is a satisfying assignment. Each clause $C_j = a_j \vee b_j \vee c_j$ is satisfied using a tuple (y_1, y_2, y_3), which assigns values for $Y(a_j)$, $Y(b_j)$, and $Y(c_j)$. This assignment corresponds to the following rule:

$$A_j \rightarrow U_{Y(a_j),y_1} U_{Y(b_j),y_2} U_{Y(c_j),y_3}$$

Set its probability to 1, and set all other rules of A_j to 0. In addition, for each r, if $Y_r = y$, set the probabilities of the rules $V_{Y_r} \rightarrow y$ and $V_{\bar{Y}_r} \rightarrow 1 - y$ to 1 and $V_{\bar{Y}_r} \rightarrow y$ and $V_{Y_r} \rightarrow 1 - y$ to 0. The rest of the weights for $S_j \rightarrow S_{j-1} A_j$ are set to 1. This assignment of rule probabilities results in a Viterbi parse of weight 1.

($\Longleftarrow$) Assume that the Viterbi parse has probability 1. From Lemma 4, we know that we can extract a consistent assignment from the Viterbi parse. In addition, for each clause C_j we have a rule

$$A_j \rightarrow U_{Y(a_j),y_1} U_{Y(b_j),y_2} U_{Y(c_j),y_3}$$

that is assigned probability 1, for some (y_1, y_2, y_3). One can verify that (y_1, y_2, y_3) are the values of the assignment for the corresponding variables in clause C_j, and that they satisfy this clause. This means that each clause is satisfied by the assignment we extracted. ■

We are now ready to prove the following result.

Theorem 4
Problem 1 is NP-hard when either requiring $\gamma > 0$ or when fixing $\gamma = 0$.

Proof
We first describe the reduction for the case of $\gamma = 0$. In Problem 1, set $\gamma = 0$, $\alpha = 1$, $G = G_\phi$, $\gamma = 0$, and $x_1 = s_\phi$. If ϕ is satisfiable, then the left side of Equation (24) can get value 0, by setting the rule probabilities according to Lemma 5, hence we would return 1 as the result of running Problem 1.

If ϕ is unsatisfiable, then we would still get value 0 only if $L(G) = \{s_\phi\}$. If G_ϕ generates a single derivation for $(10)^{3m}$, then we actually do have a satisfying assignment from Lemma 4. Otherwise (more than a single derivation), the optimal θ would have to give fractional probabilities to rules of the form $V_{Y_r} \to \{0,1\}$ (or $V_{\bar{Y}_r} \to \{0,1\}$). In that case, it is no longer true that $(10)^{3m}$ is the only generated sentence, and this is a contradiction to getting value 0 for Problem 1.

We next show that Problem 1 is NP-hard even if we require $\gamma > 0$. Let $\gamma < \frac{1}{20m}$. Set $\alpha = \gamma$, and the rest of the inputs to Problem 1 the same as before. Assume that ϕ is satisfiable. Let θ be the rule probabilities from Equation (5) after being shifted with a margin of γ. Then, because there is a derivation that uses only rules that have probability $1 - \gamma$, we have

$$\begin{aligned} h(x_1 \mid T(\theta,\gamma), G_\phi) &= \sum_z p(x_1, z \mid T(\theta,\gamma), G_\phi) \\ &\geq (1-\gamma)^{10m} \\ &> \alpha \end{aligned}$$

because the size of the parse tree for $(10)^{3m}$ is at most $10m$ (using the binarized G_ϕ) and assuming $\alpha = \gamma < (1-\gamma)^{10m}$. This inequality indeed holds whenever $\gamma < \frac{1}{20m}$. Therefore, we have $-\log h(x_1 \mid \theta) > -\log \alpha$. Problem 1 would return 0 in this case.

Now, assume that ϕ is not satisfiable. That means that any parse tree for the string $(10)^{3m}$ would have to contain two different rules headed by the same non-terminal. This means that

$$\begin{aligned} h(x_1 \mid T(\theta,\gamma), G_\phi) &= \sum_z p(x_1, z \mid T(\theta,\gamma), G_\phi) \\ &\leq \gamma \end{aligned}$$

and therefore $-\log h(x_1 \mid T(\theta,\gamma)) \leq -\log \alpha$, and Problem 1 would return 1. ■

6.2.2 An Expectation-Maximization Algorithm. Instead of solving the optimization problem implied by Equation (21), we propose a rather simple modification to the expectation-maximization (EM) algorithm (Dempster, Laird, and Rubin 1977) to approximate the optimal solution—this algorithm finds a local maximum for the maximum likelihood problem using proper approximations. The modified algorithm is given in Algorithm 1.

The modification from the usual expectation-maximization algorithm is done in the M-step: Instead of using the expected value of the sufficient statistics by counting and normalizing, we truncate the values by γ. It can be shown that if $\theta^{(0)} \in \Theta(\gamma)$, then the likelihood is guaranteed to increase (and hence, the log-loss is guaranteed to decrease) after each iteration of the algorithm.

Algorithm 1: Expectation-Maximization Algorithm with Proper Approximations.

Input: grammar G in binary normal form, initial parameters $\theta^{(0)}$, $\epsilon > 0$, $\delta > 0$, $s > 1$
Output: learned parameters θ
draw $x = \langle x_1, ..., x_n \rangle$ from p following Theorem 3; $t \leftarrow 1$;
$\gamma \leftarrow n^{-s}$;
repeat
  // $\mathbb{E}_{\theta^{(t-1)}}\left[\psi_{k,i}(z) \mid x_j\right]$ denotes the expected counts of event i in multinomial k under the distribution $\tilde{p}_n(x)p(z \mid x, \theta^{(t-1)})$
  Compute for each training example $j \in \{1, \ldots, n\}$, *for each event* $i \in \{1, 2\}$ *in each multinomial* $k \in \{1, \ldots, K\}$: $\hat{\psi}_{j,k,i} \leftarrow \mathbb{E}_{\theta^{(t-1)}}\left[\psi_{k,i}(z) \mid x_j\right]$;
  Set $\theta_{i,k}^{(t)} = \min\{1 - \gamma, \max\{\gamma, \left(\sum_{j=1}^{n} \hat{\psi}_{j,k,i}\right) / \left(\sum_{j=1}^{n} \sum_{i'=1}^{2} \hat{\psi}_{j,k,i'}\right)\}\}$;
  $t \leftarrow t + 1$;
until *convergence*;
return $\theta^{(t)}$

The reason for this likelihood increase stems from the fact that the M-step solves the optimization problem of minimizing the log-loss (with respect to $\theta \in \Theta(\gamma)$) when the posterior calculate at the E-step as the base distribution is used. This means that the M-step minimizes (in iteration t): $\mathbb{E}_r\left[-\log h(x, z \mid \theta^{(t)})\right]$ where the expectation is taken with respect to the distribution $r(x, z) = \tilde{p}_n(x)p(z \mid x, \theta^{(t-1)})$. With this notion in mind, the likelihood increase after each iteration follows from principles similar to those described in Bishop (2006) for the EM algorithm.

7. Discussion

Our framework can be specialized to improve the two main criteria which have a trade-off: the tightness of the proper approximation and the sample complexity. For example, we can improve the tightness of our proper approximations by taking a subsequence of $\mathcal{F}_n$. This will make the sample complexity bound degrade, however, because K_n will grow faster. Table 2 shows the trade-offs between parameters in our model and the effectiveness of learning.

We note that the sample complexity bounds that we give in this article give insight about the asymptotic behavior of grammar estimation, but are not necessarily

Table 2
Trade-off between quantities in our learning model and effectiveness of different criteria. K_n is the constant that satisfies the boundedness property (Theorems 2 and 3) and s is a fixed constant larger than 1 (Section 4.1).

criterion	as K_n increases ...	as s increases ...
tightness of proper approximation	improves	improves
sample complexity bound	degrades	degrades

sufficiently tight to be used in practice. It still remains an open problem to obtain sample complexity bounds which are sufficiently tight in this respect. For a discussion about the connection of grammar learning in theory and practice, we refer the reader to Clark and Lappin (2010).

It is also important to note that MLE is not the only option for estimating finite state probabilistic grammars. There has been some recent advances in learning finite state models (HMMs and finite state transducers) by using spectral analysis of matrices which consist of quantities estimated from observations only (Hsu, Kakade, and Zhang 2009; Balle, Quattoni, and Carreras 2011), based on the observable operator models of Jaeger (1999). These algorithms are not prone to local minima, and converge to the correct model as the number of samples increases, but require some assumptions about the underlying model that generates the data.

7.1 Tsybakov Noise

In this article, we chose to introduce assumptions about distributions that generate natural language data. The choice of these assumptions was motivated by observations about properties shared among treebanks. The main consequence of making these assumptions is bounding the amount of *noise* in the distribution (i.e., the amount of variation in probabilities across labels given a fixed input).

There are other ways to restrict the noise in a distribution. One condition for such noise restriction, which has received considerable recent attention in the statistical literature, is the Tsybakov noise condition (Tsybakov 2004; Koltchinskii 2006). Showing that a distribution satisfies the Tsybakov noise condition enables the use of techniques (e.g., from Koltchinskii 2006) for deriving distribution-dependent sample complexity bounds that depend on the parameters of the noise. It is therefore of interest to see whether Tsybakov noise holds under the assumptions presented in Section 3.1. We show that this is not the case, and that Tsybakov noise is too permissive. In fact, we show that p can be a probabilistic grammar itself (and hence, satisfy the assumptions in Section 3.1), and still not satisfy the Tsybakov noise conditions.

Tsybakov noise was originally introduced for classification problems (Tsybakov 2004), and was later extended to more general settings, such as the one we are facing in this article (Koltchinskii 2006). We now explain the definition of Tsybakov noise in our context.

Let $C > 0$ and $\kappa \geq 1$. We say that a distribution $p(x, z)$ satisfies the (C, κ) Tsybakov noise condition if for any $\epsilon > 0$ and $h, g \in \mathcal{H}$ such that $h, g \in \{h' \mid \mathcal{E}_p(h', \mathcal{H}) \leq \epsilon\}$, we have

$$\mathrm{dist}(g, h) \triangleq \sqrt{\mathbb{E}_p\left[\left(\frac{\log g}{\log h}\right)^2\right]} \leq C\epsilon^{1/\kappa} \tag{25}$$

This interpretation of Tsybakov noise implies that the diameter of the set of functions from the concept class that has small excess risk should shrink to 0 at the rate in Equation (25). Distribution-dependent bounds from Koltchinskii (2006) are monotone with respect to the diameter of this set of functions, and therefore demonstrating that it goes to 0 enables sharper derivations of sample complexity bounds.

We turn now to illustrating that the Tsybakov condition does not hold for probabilistic grammars in most cases. Let G be a probabilistic grammar. Define $A = A_G(\theta)$ as a matrix such that

$$(A_G(\theta))_{(k,i),(k',i')} \triangleq \frac{\mathbb{E}\left[\psi_{k,i} \times \psi_{k',i'}\right]}{\mathbb{E}[\psi_{k,i}]\mathbb{E}[\psi_{k',i'}]}$$

Theorem 5
Let G be a grammar with $K \geq 2$ and degree 2. Assume that p is $\langle G, \theta^* \rangle$ for some θ^*, such that $\theta^*_{1,1} = \theta^*_{2,1} = \mu$ and that $c_1 \leq c_2$. If $A_G(\theta^*)$ is positive definite, then p does not satisfy the Tsybakov noise condition for any (C, κ), where $C > 0$ and $\kappa \geq 1$.
See Appendix C for the proof of Theorem 5.

In Appendix C we show that $A_G(\theta)$ is positive semi-definite for any choice of θ. The main intuition behind the proof is that given a probabilistic grammar p, we can construct an hypothesis h such that the KL divergence between p and h is small, but $\text{dist}(p, h)$ is lower-bounded and is not close to 0.

We conclude that probabilistic grammars, as generative distributions of data, do not generally satisfy the Tsybakov noise condition. This motivates an alternative choice of assumptions that could lead to better understanding of rates of convergences and bounds on the excess risk. Section 3.1 states such assumptions which were also justified empirically.

7.2 Comparison to Dirichlet Maximum A Posteriori Solutions

The transformation $T(\theta, \gamma)$ from Section 4.1 can be thought of as a *smoother* for the probabilities θ: It ensures that the probability of each rule is at least γ (and as a result, the probabilities of all rules cannot exceed $1 - \gamma$). Adding pseudo-counts to frequency counts is also a common way to smooth probabilities in models based on multinomial distributions, including probabilistic grammars (Manning and Schütze 1999). These pseudo-counts can be framed as a maximum a posteriori (MAP) alternative to the maximum likelihood problem, with the choice of Bayesian prior over the parameters in the form of a Dirichlet distribution. In comparison to our framework, with (symmetric) Dirichlet smoothing, instead of truncating the probabilities with a margin γ we would set the probability of each rule (in the supervised setting) to

$$\hat{\theta}_{k,i} = \frac{\sum_{j=1}^{n} \hat{\psi}_{j,k,i} + \alpha - 1}{\sum_{j=1}^{n} \hat{\psi}_{j,k,1} + \sum_{j=1}^{n} \hat{\psi}_{j,k,2} + 2(\alpha - 1)} \tag{26}$$

for $i = 1, 2$, where $\hat{\psi}_{k,i}$ are the counts in the data of event i in multinomial k for Example j. Dirichlet smoothing can be formulated as the result of adding a symmetric Dirichlet prior over the parameters $\theta_{k,i}$ with hyperparameter α. Then Equation (26) is the mode of the posterior after observing $\hat{\psi}_{k,i}$ appearances of event i in multinomial k.

The effect of Dirichlet smoothing becomes weaker as we have more samples, because the frequency counts $\hat{\psi}_{j,k,i}$ become dominant in both the numerator and the denominator when there are more data. In this sense, the prior's effect on learning diminishes as we use more data. A similar effect occurs in our framework: $\gamma = n^{-s}$ where n is the number of samples—the more samples we have, the more we trust the

counts in the data to be reliable. There is a subtle difference, however. With the Dirichlet MAP solution, the smoothing is less dominant only if the counts of the features are large, regardless of the number of samples we have. With our framework, smoothing depends *only* on the number of samples we have. These two scenarios are related, of course: The more samples we have, the more likely it is that the counts of the events will grow large.

7.3 Other Derivations of Sample Complexity Bounds

In this section, we discuss other possible solutions to the problem of deriving sample complexity bounds for probabilistic grammars.

7.3.1 Using Talagrand's Inequality. Our bounds are based on VC theory together with classical results for empirical processes (Pollard 1984). There have been some recent developments to the derivation of rates of convergence in statistical learning theory (Massart 2000; Bartlett, Bousquet, and Mendelson 2005; Koltchinskii 2006), most prominently through the use of Talagrand's inequality (Talagrand 1994), which is a concentration of measure inequality, in the spirit of Lemma 2.

The bounds achieved with Talagrand's inequality are also distribution-dependent, and are based on the diameter of the ϵ-minimal set—the set of hypotheses which have an excess risk smaller than ϵ. We saw in Section 7.1 that the diameter of the ϵ-minimal set does not follow the Tsybakov noise condition, but it is perhaps possible to find meaningful bounds for it, in which case we may be able to get tighter bounds using Talagrand's inequality. We note that it may be possible to obtain *data-dependent* bounds for the diameter of the ϵ-minimal set, following Koltchinskii (2006), by calculating the diameter of the ϵ-minimal set using $\tilde{p}_n$.

7.3.2 Simpler Bounds for the Supervised Case. As noted in Section 6.1, minimizing empirical risk with the log-loss leads to a simple frequency count for calculating the estimated parameters of the grammar. In Corazza and Satta (2006), it has been also noted that to minimize the non-empirical risk, it is necessary to set the parameters of the grammar to the normalized *expected* count of the features.

This means that we can get bounds on the deviation of a certain parameter from the optimal parameter by applying modifications to rather simple inequalities such as Hoeffding's inequality, which determines the probability of the average of a set of i.i.d. random variables deviating from its mean. The modification would require us to split the event space into two cases: one in which the count of some features is larger than some fixed value (which will happen with small probability because of the bounded expectation of features), and one in which they are all smaller than that fixed value. Handling these two cases separately is necessary because Hoeffding's inequality requires that the count of the rules is bounded.

The bound on the deviation from the mean of the parameters (the true probability) can potentially lead to a bound on the excess risk in the supervised case. This formulation of the problem would not generalize to the unsupervised case, however, where the empirical risk minimization does not amount to simple frequency count.

7.4 Open Problems

We conclude the discussion with some directions for further exploration and future work.

7.4.1 Sample Complexity Bounds with Semi-Supervised Learning. Our bounds focus on the supervised case and the unsupervised case. There is a trivial extension to the semi-supervised case. Consider the objective function to be the sum of the likelihood for the labeled data together with the marginalized likelihood of the unlabeled data (this sum could be a weighted sum). Then, use the sample complexity bounds for each summand to derive a sample complexity bound on this sum.

It would be more interesting to extend our results to frameworks such as the one described by Balcan and Blum (2010). In that case, our discussion of sample complexity would attempt to identify how unannotated data can reduce the space of candidate probabilistic grammars to a smaller set, after which we can use the annotated data to estimate the final grammar. This reduction of the space is accomplished through a notion of compatibility, a type of fitness that the learner believes the estimated grammar should have given the distribution that generates the data. The key challenge in the case of probabilistic grammars would be to properly define this compatibility notion such that it fits the log-loss. If this is achieved, then similar machinery to that described in this paper (with proper approximations) can be followed to derive semi-supervised sample complexity bounds for probabilistic grammars.

7.4.2 Sharper Bounds for the Pseudo-Dimension of Probabilistic Grammars. The pseudo-dimension of a probabilistic grammar with the log-loss is bounded by the number of parameters in the grammar, because the logarithm of a distribution generated by a probabilistic grammar is a linear function. Typically the set of counts for the feature vectors of a probabilistic grammar resides in a subspace of a dimension which is smaller than the full dimension specified by the number of parameters, however. The reason for this is that there are usually relationships (which are often linear) between the elements in the feature counts. For example, with HMMs, the total feature count for emissions should equal the total feature count for transitions. With PCFGs, the total number of times that nonterminal rules fire equals the total number of times that features with that nonerminal in the right-hand side fired, again reducing the pseudo-dimension. An open problem that remains is characterization of the exact value pseudo-dimension for a given grammar, determined by consideration of various properties of that grammar. We conjecture, however, that a lower bound on the pseudo-dimension would be rather close to the full dimension of the grammar (the number of parameters).

It is interesting to note that there has been some work to identify the VC dimension and pseudo-dimension for certain types of grammars. Bane, Riggle, and Sonderegger (2010), for example, calculated the VC dimension for constraint-based grammars. Ishigami and Tani (1993, 1997) computed the VC dimension for finite state automata with various properties.

7.5 Conclusion

We presented a framework for performing empirical risk minimization for probabilistic grammars, in which sample complexity bounds, for the supervised case and the unsupervised case, can be derived. Our framework is based on the idea of bounded approximations used in the past to derive sample complexity bounds for graphical models.

Our framework required assumptions about the probability distribution that generates sentences or derivations in the language of the given grammar. These assumptions were tested using corpora, and found to fit the data well.

We also discussed algorithms that can be used for minimizing empirical risk in our framework, given enough samples. We showed that directly trying to minimize empirical risk in the unsupervised case is NP-hard, and suggested an approximation based on an expectation-maximization algorithm.

Appendix A. Proofs

We include in this appendix proofs for several results in the article.

Utility Lemma 1
Let $a_i \in [0,1]$, $i \in \{1,\ldots,N\}$ such that $\sum_i a_i = 1$. Define $b_1 = a_1$, $c_1 = 1 - a_1$, $b_i = \left(\frac{a_i}{a_{i-1}}\right)\left(\frac{b_{i-1}}{c_{i-1}}\right)$, and $c_i = 1 - b_i$ for $i \geq 2$. Then $a_i = \left(\prod_{j=1}^{i-1} c_j\right) b_i$.

Proof
Proof by induction on $i \in \{1,\ldots,N\}$. Clearly, the statement holds for $i = 1$. Assume it holds for arbitrary $i < N$. Then:

$$a_{i+1} = \left(\frac{a_i}{a_i}\right) a_{i+1} = \left(\left(\prod_{j=1}^{i-1} c_j\right) b_i\right) \frac{a_{i+1}}{a_i} = \left(\left(\prod_{j=1}^{i-1} c_j\right) b_i\right) \frac{c_i b_{i+1}}{b_i}$$
$$= \left(\prod_{j=1}^{i} c_j\right) b_{i+1}$$

and this completes the proof. ■

Lemma 1
Denote by $\mathcal{Z}_{\epsilon,n}$ the set $\bigcup_{f\in\mathcal{F}}\{z \mid C_n(f)(z) - f(z) \geq \epsilon\}$. Denote by $A_{\epsilon,n}$ the event "one of $z_i \in D$ is in $\mathcal{Z}_{\epsilon,n}$." If $\mathcal{F}_n$ properly approximates $\mathcal{F}$, then:

$$\mathbb{E}\left[\mathbb{E}_{\tilde{p}_n}\left[g_n\right] - \mathbb{E}_{\tilde{p}_n}\left[f_n^*\right]\right] \tag{A.1}$$
$$\leq \left|\mathbb{E}\left[\mathbb{E}_{\tilde{p}_n}\left[C_n(f_n^*)\right] \mid A_{\epsilon,n}\right]\right| p(A_{\epsilon,n}) + \left|\mathbb{E}\left[\mathbb{E}_{\tilde{p}_n}\left[f_n^*\right] \mid A_{\epsilon,n}\right]\right| p(A_{\epsilon,n}) + \epsilon_{\text{tail}}(n)$$

where the expectations are taken with respect to the data set D.

Proof
Consider the following:

$$\mathbb{E}\left[\mathbb{E}_{\tilde{p}_n}\left[g_n\right] - \mathbb{E}_{\tilde{p}_n}\left[f_n^*\right]\right]$$
$$= \mathbb{E}\left[\mathbb{E}_{\tilde{p}_n}\left[g_n\right] - \mathbb{E}_{\tilde{p}_n}\left[C_n(f_n^*)\right] + \mathbb{E}_{\tilde{p}_n}\left[C_n(f_n^*)\right] - \mathbb{E}_{\tilde{p}_n}\left[f_n^*\right]\right]$$
$$= \mathbb{E}\left[\mathbb{E}_{\tilde{p}_n}\left[g_n\right] - \mathbb{E}_{\tilde{p}_n}\left[C_n(f_n^*)\right]\right] + \mathbb{E}\left[\mathbb{E}_{\tilde{p}_n}\left[C_n(f_n^*)\right] - \mathbb{E}_{\tilde{p}_n}\left[f_n^*\right]\right]$$

Note first that $\mathbb{E}\left[\mathbb{E}_{\tilde{p}_n}\left[g_n\right] - \mathbb{E}_{\tilde{p}_n}\left[C_n(f_n^*)\right]\right] \leq 0$, by the definition of g_n as the minimizer of the empirical risk. We next bound $\mathbb{E}\left[\mathbb{E}_{\tilde{p}_n}\left[C_n(f_n^*)\right] - \mathbb{E}_{\tilde{p}_n}\left[f_n^*\right]\right]$. We know from the requirement of proper approximation that we have

$$\begin{aligned}
&\mathbb{E}\Big[\mathbb{E}_{\tilde{p}_n}\left[C_n(f_n^*)\right] - \mathbb{E}_{\tilde{p}_n}\left[f_n^*\right]\Big] = \mathbb{E}\Big[\mathbb{E}_{\tilde{p}_n}\left[C_n(f_n^*)\right] - \mathbb{E}_{\tilde{p}_n}\left[f_n^*\right] \mid A_{\epsilon,n}\Big]p(A_{\epsilon,n}) \\
&+ \mathbb{E}\Big[\mathbb{E}_{\tilde{p}_n}\left[C_n(f_n^*)\right] - \mathbb{E}_{\tilde{p}_n}\left[f_n^*\right] \mid \neg A_{\epsilon,n}\Big](1 - p(A_{\epsilon,n})) \\
&\leq |\mathbb{E}\Big[\mathbb{E}_{\tilde{p}_n}\left[C_n(f_n^*)\right] \mid A_{\epsilon,n}\Big]|p(A_{\epsilon,n}) + |\mathbb{E}\Big[\mathbb{E}_{\tilde{p}_n}\left[f_n^*\right] \mid A_{\epsilon,n}\Big]|p(A_{\epsilon,n}) + \epsilon_{\text{tail}}(n)
\end{aligned}$$

and that equals the right side of Equation (Appendix A.1). ■

Proposition 2
Let $p \in \mathcal{P}(\alpha, L, r, q, B, G)$ and let $\mathcal{F}_m$ be as defined earlier. There exists a constant $\beta = \beta(L, q, p, N) > 0$ such that $\mathcal{F}_m$ has the boundedness property with $K_m = sN\log^3 m$ and $\epsilon_{\text{bound}}(m) = m^{-\beta \log m}$.

Proof
Let $f \in \mathcal{F}_m$. Let $\mathcal{Z}(m) = \{z \mid |z| \leq \log^2 m\}$. Then, for all $z \in \mathcal{Z}(m)$ we have $|f(z)| = -\sum_{i,k} \psi(k,i) \log \theta_{k,i} \leq \sum_{i,k} \psi(k,i)(p \log m) \leq sN \log^3 m = K_m$, where the first inequality follows from $f \in \mathcal{F}_m$ ($\theta_{k,i} \geq m^{-s}$) and the second from $|z| \leq \log^2 m$. In addition, from the requirements on p we have

$$\mathbb{E}\left[|f| \times \mathbb{I}\left\{|f| \geq K_m\right\}\right] \leq \left(sN\log^3 m\right) \times \left(\sum_{k > \log^2 m} L\Lambda(k) r^k k\right) \leq \left(\kappa \log^3 m\right) \times \left(q^{\log^2 m}\right)$$

for $\kappa = \frac{sNL}{(1-q)^2}$. Finally, for $\beta(L, q, p, N) \triangleq \log \kappa + 1 + \log \frac{1}{q} = \beta > 0$ and if $m > 1$ then $\left(\kappa \log^3 m\right)\left(q^{\log^2 m}\right) \leq m^{-\beta \log m}$. ■

Utility Lemma 4
(From [Dasgupta 1997].) Let $a \in [0,1]$ and let $b = a$ if $a \in [\gamma, 1-\gamma]$, $b = \gamma$ if $a \leq \gamma$, and $b = 1 - \gamma$ if $a \geq 1 - \gamma$. Then for any $\epsilon \leq 1/2$ such that $\gamma \leq \epsilon/(1+\epsilon)$ we have $\log a/b \leq \epsilon$.

Proposition 3
Let $p \in \mathcal{P}(\alpha, L, r, q, B, G)$ and let $\mathcal{F}_m$ as defined earlier. There exists an M such that for any $m > M$ we have

$$p\left(\bigcup_{f \in \mathcal{F}} \{z \mid C_m(f)(z) - f(z) \geq \epsilon_{\text{tail}}(m)\}\right) \leq \epsilon_{\text{tail}}(m)$$

for $\epsilon_{\text{tail}}(m) = \dfrac{N \log^2 m}{m^s - 1}$ and $C_m(f) = T(f, m^{-s})$.

Proof
Let $\mathcal{Z}(m)$ be the set of derivations of size bigger than $\log^2 m$. Let $f \in \mathcal{F}$. Define $f' = T(f, m^{-s})$. For any $z \notin \mathcal{Z}(m)$ we have that

$$f'(z) - f(z) = -\sum_{k=1}^{K} \left(\phi_{k,1}(z)\log\theta_{k,1} + \phi_{k,2}(z)\log\theta_{k,2} - \phi_{k,1}(z)\log\theta'_{k,1} - \phi_{k,1}(z)\log\theta'_{k,2}\right)$$
$$\leq \sum_{k=1}^{K} \log^2 m \left(\max\{0, \log(\theta'_{k,1}/\theta_{k,1})\} + \max\{0, \log(\theta'_{k,2}/\theta_{k,2})\}\right) \quad \text{(A.2)}$$

Without loss of generality, assume $\epsilon_{\text{tail}}(n)/N\log^2 m \leq 1/2$. Let $\gamma = \dfrac{\epsilon_{\text{tail}}(m)/N\log^2 m}{1+\epsilon_{\text{tail}}(m)/N\log^2 m} = 1/m^s$. From Utility Lemma 4 we have that $\log(\theta'_{k,i}/\theta_{k,i}) \leq \epsilon_{\text{tail}}(m)/N\log m$. Plug this into Equation A.2 ($N = 2K$) to get that for all $z \notin \mathcal{Z}(m)$ we have $f'(z) - f(z) \leq \epsilon_{\text{tail}}(m)$. It remains to show that the measure $p(\mathcal{Z}(m)) \leq \epsilon_{\text{tail}}(m)$. Note that $\sum_{z\in\mathcal{Z}(m)} p(z) \leq \sum_{k>\log^2 m} L\Lambda(k)r^k \leq L\sum_{k>\log^2 m} q^k = Lq^{\log^2 m}/(1-q) < \epsilon_{\text{tail}}(m)$ for $m > M$ where M is fixed. ■

Proposition 7
There exists a $\beta'(L, p, q, N) > 0$ such that $\mathcal{F}'_m$ has the boundedness property with $K_m = sN\log^3 m$ and $\epsilon_{\text{bound}}(m) = m^{-\beta'\log m}$.

Proof
From the requirement of p, we know that for any x we have a z such that $\text{yield}(z) = x$ and $|z| \leq \alpha|x|$. Therefore, if we let $\mathcal{X}(m) = \{x \mid |x| \leq \log^2 m/\alpha\}$, then we have for any $f \in \mathcal{F}'_m$ and $x \in \mathcal{X}(m)$ that $f(x) \leq sN\log^3 m = K_m$ (similarly to the proof of Proposition 2). Denote by $f_1(x, z)$ the function in $\mathcal{F}_m$ such that $f(x) = -\log\sum_z \exp(-f_1(x, z))$.

In addition, from the requirements on p and the definition of K_m we have

$$\mathbb{E}\left[|f| \times \mathbb{I}\{|f| \geq K_m\}\right] = \sum_x p(x)f(x)\mathbb{I}\{f \geq K_m\}$$
$$= \sum_{x:|x|>\log^2 m/\alpha} p(x)f(x)$$
$$\leq \sum_{x:|x|>\log^2 m/\alpha} p(x)f_1(x, z(x))$$

where $z(x)$ is some derivation for x. We have

$$\begin{aligned}\sum_{x:|x|>\log^2 m/\alpha} p(x) f_1(x, z(x)) &\leq \sum_{x:|x|\geq\log^2 m/\alpha} \sum_{z\in D_x(G)} p(x,z) f_1(x, z(x)) \\ &\leq sN \log m \sum_{x:|x|>\log^2 m/\alpha} \sum_{z} p(x,z)|z(x)| \\ &\leq sN \log m \sum_{k>\log^2 m} \Lambda(k) r^k k \\ &\leq sN \log m \sum_{k>\log^2 m} q^k k \leq \kappa \log m q^{log^2 m}\end{aligned}$$

for some constant $\kappa > 0$. Finally, for some $\beta'(L, p, q, N) = \beta' > 0$ and some constant M, if $m > M$ then $\kappa \log m \left(q^{\log^2 m}\right) \leq m^{-\beta' \log m}$. ■

Utility Lemma 2
For $a_i, b_i \geq 0$, if $-\log \sum_i a_i + \log \sum_i b_i \geq \epsilon$ then there exists an i such that $-\log a_i + \log b_i \geq \epsilon$.

Proof
Assume $-\log a_i + \log b_i < \epsilon$ for all i. Then, $b_i/a_i < e^\epsilon$, therefore $\sum_i b_i / \sum_i a_i < e^\epsilon$, therefore $-\log \sum_i a_i + \log \sum_i b_i < \epsilon$ which is a contradiction to $-\log \sum_i a_i + \log \sum_i b_i \geq \epsilon$. ■

The next lemma is the main concentation of measure result that we use. Its proof requires some simple modification to the proof given for Theorem 24 in Pollard (1984, pages 30–31).

Lemma 2
Let $\mathcal{F}_n$ be a permissible class of functions such that for every $f \in \mathcal{F}_n$ we have $\mathbb{E}[|f| \times \mathbb{I}\{|f| \leq K_n\}] \leq \epsilon_{\text{bound}}(n)$. Let $\mathcal{F}_{\text{truncated},n} = \{f \times \mathbb{I}\{f \leq K_n\} \mid f \in \mathcal{F}_m\}$, that is, the set of functions from $\mathcal{F}_n$ after being truncated by K_n. Then for $\epsilon > 0$ we have

$$p\left(\sup_{f\in\mathcal{F}_n} |\mathbb{E}_{\tilde{p}_n}\left[f\right] - \mathbb{E}_p\left[f\right]| > 2\epsilon\right) \leq 8\mathcal{N}(\epsilon/8, \mathcal{F}_{\text{truncated},n}) \exp\left(-\frac{1}{128} n\epsilon^2/K_n^2\right) + \epsilon_{\text{bound}}(n)/\epsilon$$

provided $n \geq K_n^2/4\epsilon^2$ and $\epsilon_{\text{bound}}(n) < \epsilon$.

Proof
First note that

$$\begin{aligned}\sup_{f\in\mathcal{F}_n} |\mathbb{E}_{\tilde{p}_n}\left[f\right] - \mathbb{E}_p\left[f\right]| &\leq \sup_{f\in\mathcal{F}_n} |\mathbb{E}_{\tilde{p}_n}\left[f\mathbb{I}\{|f| \leq K_n\}\right] - \mathbb{E}_p\left[f\mathbb{I}\{|f| \leq K_n\}\right]| \\ &\quad + \sup_{f\in\mathcal{F}_n} \mathbb{E}_{\tilde{p}_n}\left[|f|\mathbb{I}\{|f| \leq K_n\}\right] + \sup_{f\in\mathcal{F}_n} \mathbb{E}_p\left[|f|\mathbb{I}\{|f| \leq K_n\}\right]\end{aligned}$$

We have $\sup_{f \in \mathcal{F}_n} \mathbb{E}_p \left[|f| \mathbb{I}\{|f| \leq K_n\} \right] \leq \epsilon_{\text{bound}}(n) < \epsilon$, and also, from Markov inequality, we have

$$P(\sup_{f \in \mathcal{F}_n} \mathbb{E}_{\tilde{p}_n} \left[|f| \mathbb{I}\{|f| \leq K_n\} \right] > \epsilon) \leq \epsilon_{\text{bound}}(n)/\epsilon$$

At this point, we can follow the proof of Theorem 24 in Pollard (1984), and its extension on pages 30–31 to get Lemma 2, using the shifted set of functions $\mathcal{F}_{\text{truncated},n}$. ■

Appendix B. Minimizing Log-Loss for Probabilistic Grammars

Central to our algorithms for minimizing the log-loss (both in the supervised case and the unsupervised case) is a convex optimization problem of the form

$$\min_{\theta} \sum_{k=1}^{K} c_{k,1} \log \theta_{k,1} + c_{k,2} \log \theta_{k,2}$$

$$\text{such that } \forall k \in \{1, \ldots, K\} :$$

$$\theta_{k,1} + \theta_{k,2} = 1$$

$$\gamma \leq \theta_{k,1} \leq 1 - \gamma$$

$$\gamma \leq \theta_{k,2} \leq 1 - \gamma$$

for constants $c_{k,i}$ which depend on $\tilde{p}_n$ or some other intermediate distribution in the case of the expectation-maximization algorithm and γ which is a margin determined by the number of samples. This minimization problem can be decomposed into several optimization problems, one for each k, each having the following form:

$$\max_{\beta} c_1 \beta_1 + c_2 \beta_2 \tag{B.1}$$

$$\text{such that } \exp(\beta_1) + \exp(\beta_2) = 1 \tag{B.2}$$

$$\gamma \leq \beta_1 \leq 1 - \gamma \tag{B.3}$$

$$\gamma \leq \beta_2 \leq 1 - \gamma \tag{B.4}$$

where $c_i \geq 0$ and $1/2 > \gamma \geq 0$. Ignore for a moment the constraints $\gamma \leq \beta_i \leq 1 - \gamma$. In that case, this can be thought of as a regular maximum likelihood estimation problem, so $\beta_i = c_i/(c_1 + c_2)$. We give a derivation of this result in this simple case for completion. We use Lagranian multipliers to solve this problem. Let $F(\beta 1, \beta 2) = c_1 \beta_1 + c_2 \beta_2$. Define the Lagrangian:

$$\begin{aligned} g(\lambda) &= \inf_{\beta} L(\lambda, \beta) \\ &= \inf_{\beta} c_1 \beta_1 + c_2 \beta_2 + \lambda(\exp(\beta_1) + \exp(\beta_2) - 1) \end{aligned}$$

Taking the derivative of the term we minimize in the Lagrangian, we have

$$\frac{\partial L}{\partial \beta_i} = c_i + \lambda \exp(\beta_i)$$

Setting the derivatives to 0 for minimization, we have

$$g(\lambda) = c_1 \log(-c_1/\lambda) + c_2 \log(-c_2/\lambda) + \lambda(-c_1/\lambda - c_2/\lambda - 1) \tag{B.5}$$

$g(\lambda)$ is the objective function of the dual problem of Equation (B.1)–Equation (B.2). We would like to minimize Equation (B.5) with respect to λ. The derivative of $g(\lambda)$ is

$$\frac{\partial g}{\partial \lambda} = -c_1/\lambda - c_2/\lambda - 1$$

hence when equating the derivative of $g(\lambda)$ to 0, we get $\lambda = -(c_1 + c_2)$, and therefore the solution is $\beta_i^* = \log(c_i/(c_1 + c_2))$. We need to verify that the solution to the dual problem indeed gets the optimal value for the primal. Because the primal problem is convex, it is sufficient to verify that the Karush-Kuhn-Tucker (KKT) conditions hold (Boyd and Vandenberghe 2004). Indeed, we have

$$\begin{aligned}\frac{\partial F}{\partial \beta_i}(\beta^*) + \lambda \frac{\partial h}{\partial \beta_i}(\beta^*) &= c_i - (c_1 + c_2) \times \frac{c_i}{c_1 + c_2} \\ &= 0\end{aligned}$$

where $h(\beta) \triangleq \exp(\beta) + \exp(\beta) - 1$ stands for the equality constraint. The rest of the KKT conditions trivially hold, therefore β^* is the optimal solution for Equations (B.1)–(B.2).

Note that if $1 - \gamma < c_i/(c_1 + c_2) < \gamma$, then this is the solution even when again adding the constraints in Equation (B.3) and (B.4). When $c_1/(c_1 + c_2) < \gamma$, then the solution is $\beta_1^* = \gamma$ and $\beta_2^* = 1 - \gamma$. Similarly, when $c_2/(c_1 + c_2) < \gamma$ then the solution is $\beta_2^* = \gamma$ and $\beta_1^* = 1 - \gamma$. We describe why this is true for the first case. The second case follows very similarly. Assume $c_1/(c_1 + c_2) < \gamma$. We want to show that for any choice of $\beta \in [0, 1]$ such that $\beta > \gamma$ we have

$$c_1 \log \gamma + c_2 \log(1 - \gamma) \geq c_1 \log \beta + c_2 \log(1 - \beta)$$

Divide both sides of the inequality by $c_1 + c_2$ and we get that we need to show that

$$\frac{c_1}{c_1 + c_2} \log(\gamma/\beta) + \frac{c_2}{c_1 + c_2} \log\left(\frac{1 - \gamma}{1 - \beta}\right) \geq 0$$

Because we have $\beta > \gamma$, and we also have $c_1/(c_1 + c_2) < \gamma$, it is sufficient to show that

$$\gamma \log(\gamma/\beta) + (1 - \gamma) \log\left(\frac{1 - \gamma}{1 - \beta}\right) \geq 0 \tag{B.6}$$

Equation (B.6) is precisely the definition of the KL divergence between the distribution of a coin with probability γ of heads and the distribution of a coin with probability β

of heads, and therefore the right side in Equation (B.6) is positive, and we get what we need.

Appendix C. Counterexample to Tsybakov Noise (Proofs)

Lemma 6
$A = A_G(\theta)$ is positive semi-definite for any probabilistic grammar $\langle G, \theta \rangle$.

Proof
Let $d_{k,i}$ be a collection of constants. Define the random variable:

$$R(z) = \sum_{i,k} \frac{d_{k,i}}{\mathbb{E}\left[\psi_{k,i}\right]} \psi_{k,i}(z)$$

We have that

$$\mathbb{E}\left[R^2\right] = \sum_{i,i'} \sum_{k,k'} A_{(k,i),(k',i')} d_{k,i} d_{k',i'}$$

which is always larger or equal to 0. Therefore, A is positive semi-definite. ■

Lemma 7
Let $0 < \mu < 1/2$, $c_1, c_2 \geq 0$. Let $\kappa, C > 0$. Also, assume that $c_1 \leq c_2$. For any $\epsilon > 0$, define:

$$a = \mu \left(\exp\left(\frac{C\epsilon^{1/\kappa} + \epsilon/2}{c_1} \right) \right) = \alpha_1 \mu$$

$$b = \mu \left(\exp\left(\frac{-C\epsilon^{1/\kappa} + \epsilon/2}{c_2} \right) \right) = \alpha_2 \mu$$

$$t(\epsilon) = c_1 \left(\frac{1-\mu}{1-a} \right) + c_2 \left(\frac{1-\mu}{1-b} \right) - (c_1 + c_2) \exp(\epsilon/2)$$

Then, for small enough ϵ, we have $t(\epsilon) \leq 0$.

Proof
We have that $t(\epsilon) \leq 0$ if

$$\begin{aligned} ac_2 + bc_1 &\geq -\frac{(c_1 + c_2)(1-a)(1-b)}{1-\mu} \exp(\epsilon/2) + c_1 + c_2 \\ &= (c_1 + c_2) \left(1 - \frac{(1-a)(1-b)}{(1-\mu)\exp(-\epsilon/2)} \right) \end{aligned} \tag{C.1}$$

First, show that

$$\frac{(1-a)(1-b)}{(1-\mu)\exp(-\epsilon/2)} \geq 1 - \mu \tag{C.2}$$

which happens if (after substituting $a = \alpha_1\mu$, $b = \alpha_2\mu$)

$$\mu \leq (\alpha_1 + \alpha_2 - 2)/(1 - \alpha_1\alpha_2)$$

Note we have $\alpha_1\alpha_2 > 1$ because $c_1 \leq c_2$. In addition, we have $\alpha_1 + \alpha_2 - 2 \geq 0$ for small enough ϵ (can be shown by taking the derivative, with respect to ϵ of $\alpha_1 + \alpha_2 - 2$, which is always positive for small enough ϵ, and in addition, noticing that the value of $\alpha_1 + \alpha_2 - 2$ is 0 when $\epsilon = 0$.) Therefore, Equation (C.2) is true.

Substituting Equation (C.2) in Equation (C.1), we have that $t(\epsilon) \leq 0$ if

$$ac_2 + bc_1 \geq (c_1 + c_2)\mu$$

which is equivalent to

$$c_2\alpha_1 + c_1\alpha_2 \geq c_1 + c_2 \tag{C.3}$$

Taking again the derivative of the left side of Equation (C.3), we have that it is an increasing function of ϵ (if $c_1 \leq c_2$), and in addition at $\epsilon = 0$ it obtains the value $c_1 + c_2$. Therefore, Equation (C.3) holds, and therefore $t(\epsilon) \leq 0$ for small enough ϵ. ■

Theorem 5
Let G be a grammar with $K \geq 2$ and degree 2. Assume that p is $\langle G, \theta^* \rangle$ for some θ^*, such that $\theta^*_{1,1} = \theta^*_{2,1} = \mu$ and that $c_1 \leq c_2$. If $A_G(\theta^*)$ is positive definite, then p does not satisfy the Tsybakov noise condition for any (C, κ), where $C > 0$ and $\kappa \geq 1$.

Proof
Define λ to be the eigenvalue of $A_G(\theta)$ with the smallest value (λ is positive). Also, define $\mathbf{v}(\theta)$ to be a vector indexed by k, i such that

$$v_{k,i}(\theta) = \mathbb{E}\left[\psi_{k,i}\right] \log \frac{\theta^*_{k,i}}{\theta_{k,i}}.$$

Simple algebra shows that for any $h \in \mathcal{H}(G)$ (and the fact that $p \in \mathcal{H}(G)$), we have

$$\mathcal{E}_p(h) = D_{\mathrm{KL}}(p\|h) = \sum_{k=1}^{K} \left(\mathbb{E}_p\left[\psi_{k,1}\right] \log \frac{\theta^*_{k,1}}{\theta_{k,1}} + \mathbb{E}_p\left[\psi_{k,1}\right] \log \left(\frac{1 - \theta^*_{k,1}}{1 - \theta_{k,1}} \right) \right)$$

For a $C > 0$ and $\kappa \geq 1$, define $\alpha = C\epsilon^{1/\kappa}$. Let $\epsilon < \alpha$. First, we construct an h such that $D_{\mathrm{KL}}(p\|h) < \epsilon + \epsilon/2$ but $\mathrm{dist}(p, h) > C\epsilon^{1/\kappa}$ as $\epsilon \to 0$. The construction follows. Parametrize h by θ such that θ is identical to θ^* except for $k = 1, 2$, in which case we have

$$\theta_{1,1} = \theta^*_{1,1}\left(\exp\left(\frac{\alpha + \epsilon/2}{c_1}\right)\right) = \mu\left(\exp\left(\frac{\alpha + \epsilon/2}{c_1}\right)\right) \tag{C.4}$$

$$\theta_{2,1} = \theta^*_{2,1}\left(\exp\left(\frac{-\alpha + \epsilon/2}{c_2}\right)\right) = \mu\left(\exp\left(\frac{-\alpha + \epsilon/2}{c_2}\right)\right) \tag{C.5}$$

Note that $\mu \leq \theta_{1,1} \leq 1/2$ and $\theta_{2,1} < \mu$. Then, we have that

$$\begin{aligned} D_{\mathrm{KL}}(p\|h) &= \sum_{k=1}^{K} \left(\mathbb{E}_p \left[\psi_{k,1}\right] \log \frac{\theta^*_{k,1}}{\theta_{k,1}} + \mathbb{E}_p \left[\psi_{k,1}\right] \log \left(\frac{1-\theta^*_{k,1}}{1-\theta_{k,1}} \right) \right) \\ &= \epsilon + c_1 \log \frac{1-\theta^*_{k,1}}{1-\theta_{1,1}} + c_2 \log \frac{1-\theta^*_{k,2}}{1-\theta_{2,1}} \\ &= \epsilon + c_1 \log \frac{1-\mu}{1-\theta_{1,1}} + c_2 \log \frac{1-\mu}{1-\theta_{2,1}} \end{aligned}$$

We also have

$$c_1 \log \frac{1-\mu}{1-\theta_{1,1}} + c_2 \log \frac{1-\mu}{1-\theta_{2,1}} \leq 0 \tag{C.6}$$

if

$$c_1 \times \frac{1-\mu}{1-\theta_{1,1}} + c_2 \times \frac{1-\mu}{1-\theta_{2,1}} \leq c_1 + c_2 \tag{C.7}$$

(This can be shown by dividing Equation [C.6] by $c_1 + c_2$ and then using the concavity of the logarithm function.) From Lemma 7, we have that Equation (C.7) holds. Therefore,

$$D_{\mathrm{KL}}(p\|h) \leq 2\epsilon$$

Now, consider the following, which can be shown through algebraic manipulation:

$$\mathrm{dist}(p,h) = \mathbb{E}\left[\left(\log \frac{p}{h} \right)^2 \right] = \sum_{k,k'} \sum_{i,i'} \mathbb{E}\left[\psi_{k,i} \times \psi_{k',i'}\right] \left(\log \frac{\theta^*_{k,i}}{\theta_{k,i}} \right) \left(\log \frac{\theta^*_{k',i'}}{\theta_{k',i'}} \right)$$

Then, additional algebraic simplification shows that

$$\mathbb{E}\left[\left(\log \frac{p}{h} \right)^2 \right] = \mathbf{v}(\theta) A \mathbf{v}(\theta)^\top$$

A fact from linear algebra states that

$$\mathbf{v}(\theta) A \mathbf{v}(\theta)^\top \geq \lambda ||\mathbf{v}(\theta)||_2^2$$

where λ is the smallest eigenvalue in A. From the construction of θ and Equation (C.4)–(C.5), we have that $||\mathbf{v}(\theta)||_2^2 > \alpha^2$. Therefore,

$$\mathbb{E}\left[\left(\log \frac{p}{h} \right)^2 \right] \geq \lambda \alpha^2$$

which means $\mathrm{dist}(p,h) \geq \sqrt{\lambda} C \epsilon^{1/\kappa}$. Therefore, p does not satisfy the Tsybakov noise condition with parameters (D, κ) for any $D > 0$. ■

Appendix D. Notation

Table D.1 gives a table of notation for symbols used throughout this article.

Table 1
Table of notation symbols used in this article.

	Symbol	Description	1st Mention
ERM	$\mathcal{X}$	Instance space (natural language sentences)	Sec. 2
	$\mathcal{Z}$	Output space (grammar derivations)	Sec. 2
	p	Distribution generating the data	Sec. 2
	$\mathcal{Q}$	Concept space, a family of distributions	Sec. 2
	q	An estimated distribution	Sec. 2
	q_{opt}	Risk minimizer	Eq. 1
	n	Number of available samples	Sec. 2
	$\tilde{p}_n$	Empirical distribution	Sec. 2
	q^*	Empirical risk minimizer	Eq. 2
	$\mathcal{E}_p(q;\mathcal{Q})$	Excess risk	Eq. 4
	$R_n(\mathcal{Q})$	Empirical process for the log-loss	Eq. 5
Grammars	G	Grammar (for example, CFG rules)	Sec. 3
	θ	Probabilistic grammar parameters	Sec. 3
	K	Number of multinomials in the probabilistic grammar	Eq. 11
	N_k	Size of the kth multinomial of the probabilistic grammar	Eq. 11
	N	$\sum_{k=1}^{K} N_k$	Sec. 3
	x	Sentence in the language of the grammar	Sec. 3
	z	Derivation in the grammar	Sec. 3
	$\psi_{k,i}(x,z)$	Count of the ith event firing in the kth multinomial in x and z	Eq. 11
	Θ_G	Parameter space for a given probabilistic grammar G	Eq. 11
	θ	Parameters for a probabilistic grammar	Eq. 11
	$\deg(G)$	The degree of G, $\max_k N_k$	Sec. 3
	$D_x(G)$	The set of derivations for string x	Sec. 3
	$\mathcal{H}, \mathcal{H}(G)$	Concept space, a set of probabilistic grammars	Sec. 3
	$\mathcal{F}, \mathcal{F}(G)$	Negated log-concept space, $\{-\log h \mid h \in \mathcal{H}(G)\}$	Sec. 3
	L	Constant determining distributional assumption	Sec. 3.1
	q	Constant determining distributional assumption	Sec. 3.1
	r	Constant determining distributional assumption	Sec. 3.1
Proper Approximations	$\mathcal{F}_n$	Element n in a proper approximation (contained in $\mathcal{F}$)	Sec. 4
	$\epsilon_{\text{tail}}(n)$	Convergence rate for the boundedness property	Sec. 4
	$\epsilon_{\text{bound}}(n)$	Convergence rate for the tightness property	Sec. 4
	$C_n(f)$	A map for $f \in \mathcal{F}$ to $f' \in \mathcal{F}_n$	Sec. 4
	$T(\theta,\gamma)$	Parameters θ with shifted probabilities	Sec. 4.1
	$T(f,\gamma)$	$f \in \mathcal{F}$ with shifted probabilities	Sec. 4.1
	$\Theta_G(\gamma)$	Set of parameters $\{T(\theta,\gamma) \mid \theta \in \Theta_G\}$ for a given G	Sec. 4.1
	s	A constant larger than 1 on which boundedness property depends	Sec. 4.1
	$\beta(L,q,p,N)$	A constant on which sample complexity depends for the supervised case	Prop. 2
	$\mathcal{F}'_n$	Element n in a proper approximation (contained in $\mathcal{F}$)	Sec. 4
	$C'_n(f)$	A map for $f \in \mathcal{F}$ to $f' \in \mathcal{F}_n$	Sec. 4
	$\epsilon'_{\text{tail}}(n)$	Convergence rate for the soundness property	Sec. 4
	$\epsilon'_{\text{bound}}(n)$	Convergence rate for the tightness property	Sec. 4
	$\beta'(L,q,p,N)$	A constant on which sample complexity depends for the unsupervised case	Sec. 5.3

Acknowledgments

The authors thank the anonymous reviewers for their comments and Avrim Blum, Steve Hanneke, Mark Johnson, John Lafferty, Dan Roth, and Eric Xing for useful conversations. This research was supported by National Science Foundation grant IIS-0915187.

References

Abe, N., J. Takeuchi, and M. Warmuth. 1991. Polynomial learnability of probabilistic concepts with respect to the Kullback-Leiber divergence. In *Proceedings of the Conference on Learning Theory*, pages 277–289.

Abe, N. and M. Warmuth. 1992. On the computational complexity of approximating distributions by probabilistic automata. *Machine Learning*, 2:205–260.

Angluin, D. 1987. Learning regular sets from queries and counterexamples. *Information and Computation*, 75:87–106.

Anthony, M. and P. L. Bartlett. 1999. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press.

Balcan, M. and A. Blum. 2010. A discriminative model for semi-supervised learning. *Journal of the Association for Computing Machinery*, 57(3):1–46.

Balle, B., A. Quattoni, and X. Carreras. 2011. A spectral learning algorithm for finite state transducers. In *Proceedings of the European Conference on Machine Learning/the Principles and Practice of Knowledge Discovery in Databases*, pages 156–171.

Bane, M., J. Riggle, and M. Sonderegger. 2010. The VC dimension of constraint-based grammars. *Lingua*, 120(5):1194–1208.

Bartlett, P., O. Bousquet, and S. Mendelson. 2005. Local Rademacher complexities. *Annals of Statistics*, 33(4):1497–1537.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer, Berlin.

Boyd, S. and L. Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

Carrasco, R. 1997. Accurate computation of the relative entropy between stochastic regular grammars. *Theoretical Informatics and Applications*, 31(5):437–444.

Carroll, G. and E. Charniak. 1992. Two experiments on learning probabilistic dependency grammars from corpora. Technical report, Brown University, Providence, RI.

Charniak, E. 1993. *Statistical Language Learning*. MIT Press, Cambridge, MA.

Charniak, E. and M. Johnson. 2005. Coarse-to-fine *n*-best parsing and maxent discriminative reranking. In *Proceedings of the Association for Computational Linguistics*, pages 173–180.

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

Clark, A., R. Eyraud, and A. Habrard. 2008. A polynomial algorithm for the inference of context free languages. In *Proceedings of the International Colloquium on Grammatical Inference*, pages 29–42.

Clark, A. and S. Lappin. 2010. Unsupervised learning and grammar induction. In Alexander Clark, Chris Fox, and Shalom Lappin, editors, *The Handbook of Computational Linguistics and Natural Language Processing*. Wiley-Blackwell, London, pages 197–220.

Clark, A. and F. Thollard. 2004. PAC-learnability of probabilistic deterministic finite state automata. *Journal of Machine Learning Research*, 5:473–497.

Cohen, S. B. and N. A. Smith. 2010a. Covariance in unsupervised learning of probabilistic grammars. *Journal of Machine Learning Research*, 11:3017–3051.

Cohen, S. B. and N. A. Smith. 2010b. Empirical risk minimization with approximations of probabilistic grammars. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 424–432.

Cohen, S. B. and N. A. Smith. 2010c. Viterbi training for PCFGs: Hardness results and competitiveness of uniform initialization. In *Proceedings of the Association for Computational Linguistics*, pages 1502–1511.

Collins, M. 2003. Head-driven statistical models for natural language processing. *Computational Linguistics*, 29:589–637.

Collins, M. 2004. Parameter estimation for statistical parsing models: Theory and practice of distribution-free methods. In H. Bunt, J. Carroll, and G. Satta, *Text, Speech and Language Technology (New Developments in Parsing Technology)*. Kluwer, Dordrecht, pages 19–55.

Corazza, A. and G. Satta. 2006. Cross-entropy and estimation of probabilistic context-free

grammars. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 335–342.
Cover, T. M. and J. A. Thomas. 1991. *Elements of Information Theory*. Wiley, London.
Dasgupta, S. 1997. The sample complexity of learning fixed-structure bayesian networks. *Machine Learning*, 29(2–3):165–180.
de la Higuera, C. 2005. A bibliographical study of grammatical inference. *Pattern Recognition*, 38:1332–1348.
Dempster, A., N. Laird, and D. Rubin. 1977. Maximum likelihood estimation from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38.
Gildea, D. 2010. Optimal parsing strategies for linear context-free rewriting systems. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 769–776.
Gómez-Rodríguez, C. and G. Satta. 2009. An optimal-time binarization algorithm for linear context-free rewriting systems with fan-out two. In *Proceedings of the Association for Computational Linguistics-International Joint Conference on Natural Language Processing*, pages 985–993.
Grenander, U. 1981. *Abstract Inference*. Wiley, New York.
Haussler, D. 1992. Decision-theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100:78–150.
Hsu, D., S. M. Kakade, and T. Zhang. 2009. A spectral algorithm for learning hidden Markov models. In *Proceedings of the Conference on Learning Theory*.
Ishigami, Y. and S. Tani. 1993. The VC-dimensions of finite automata with *n* states. In *Proceedings of Algorithmic Learning Theory*, pages 328–341.
Ishigami, Y. and S. Tani. 1997. VC-dimensions of finite automata and commutative finite automata with *k* letters and *n* states. *Applied Mathematics*, 74(3):229–240.
Jaeger, H. 1999. Observable operator models for discrete stochastic time series. *Neural Computation*, 12:1371–1398.
Kearns, M. and L. Valiant. 1989. Cryptographic limitations on learning Boolean formulae and finite automata. In *Proceedings of the 21st Association for Computing Machinery Symposium on the Theory of Computing*, pages 433–444.
Kearns, M. J. and U. V. Vazirani. 1994. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA.
Klein, D. and C. D. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the Association for Computational Linguistics*, pages 478–487.
Koltchinskii, V. 2006. Local Rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics*, 34(6):2593–2656.
Leermakers, R. 1989. How to cover a grammar. In *Proceedings of the Association for Computational Linguistics*, pages 135–142.
Manning, C. D. and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.
Massart, P. 2000. Some applications of concentration inequalities to statistics. *Annales de la Faculté des Sciences de Toulouse*, IX(2):245–303.
Nijholt, A. 1980. *Context-Free Grammars: Covers, Normal Forms, and Parsing* (volume 93 of *Lecture Notes in Computer Science*). Springer-Verlag, Berlin.
Palmer, N. and P. W. Goldberg. 2007. PAC-learnability of probabilistic deterministic finite state automata in terms of variation distance. In *Proceedings of Algorithmic Learning Theory*, pages 157–170.
Pereira, F. C. N. and Y. Schabes. 1992. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the Association for Computational Linguistics*, pages 128–135.
Pitt, L. 1989. Inductive inference, DFAs, and computational complexity. *Analogical and Inductive Inference*, 397:18–44.
Pollard, D. 1984. *Convergence of Stochastic Processes*. Springer-Verlag, New York.
Ron, D. 1995. *Automata Learning and Its Applications*. Ph.D. thesis, Hebrew University of Jerusalem.
Ron, D., Y. Singer, and N. Tishby. 1998. On the learnability and usage of acyclic probabilistic finite automata. *Journal of Computer and System Sciences*, 56(2):133–152.

Shalev-Shwartz, S., O. Shamir, K. Sridharan, and N. Srebro. 2009. Learnability and stability in the general learning setting. In *Proceedings of the Conference on Learning Theory*.

Sipser, M. 2006. *Introduction to the Theory of Computation, Second Edition*. Thomson Course Technology, Boston, MA.

Talagrand, M. 1994. Sharper bounds for Gaussian and empirical processes. *Annals of Probability*, 22:28–76.

Terwijn, S. A. 2002. On the learnability of hidden Markov models. In P. Adriaans, H. Fernow, & M. van Zaane. *Grammatical Inference: Algorithms and Applications* (Lecture Notes in Computer Science). Springer, Berlin, pages 344–348.

Tsybakov, A. 2004. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166.

Vapnik, V. N. 1998. *Statistical Learning Theory*. Wiley-Interscience, New York.