Weighted and Probabilistic Context-Free Grammars Are Equally Expressive

Noah A. Smith*
Carnegie Mellon University

Mark Johnson**
Brown University

This paper studies the relationship between weighted context-free grammars (WCFGs), where each production is associated with a positive real-valued weight, and probabilistic context-free grammars (PCFGs), where the weights of the productions associated with a nonterminal are constrained to sum to one. Since the class of WCFGs properly includes the PCFGs, one might expect that WCFGs can describe distributions that PCFGs cannot. However, Chi (1999) and Abney, McAllester, and Pereira (1999) proved that every WCFG distribution is equivalent to some PCFG distribution. We extend their results to conditional distributions, and show that every WCFG conditional distribution of parses given strings is also the conditional distribution defined by some PCFG, even when the WCFG's partition function diverges. This shows that any parsing or labeling accuracy improvement from conditional estimation of WCFGs or CRFs over joint estimation of PCFGs or HMMs is due to the estimation procedure rather than the change in model class, since PCFGs and HMMs are exactly as expressive as WCFGs and chain-structured CRFs respectively.

Introduction

In recent years the field of computational linguistics has turned to machine learning to aid in the development of accurate tools for language processing. A widely used example, applied to parsing and tagging tasks of various kinds, is a *weighted grammar*. Adding weights to a formal grammar allows disambiguation (more generally, ranking of analyses) and can lead to more efficient parsing. Machine learning comes in when we wish to choose those weights empirically.

The predominant approach for many years was to select a probabilistic model—such as a hidden Markov model (HMM) or probabilistic context-free grammar (PCFG)—that defined a distribution over the structures allowed by a grammar. Given a treebank, maximum likelihood estimation can be applied to learn the probability values in the model.

More recently, new machine learning methods have been developed or extended to handle models of grammatical structure. Notably, conditional estimation (Ratnaparkhi, Roukos, and Ward 1994; Johnson et al. 1999; Lafferty, McCallum, and Pereira 2001), maximum margin estimation (Taskar et al. 2004), and unsupervised contrastive

* School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15217, USA. E-mail: nasmith@cs.cmu.edu
** Department of Cognitive and Linguistic Sciences, Brown University, Providence, RI 02912, USA. E-mail: Mark_Johnson@brown.edu

Submission received: 30 November 2005; revised submission received: 11 January 2007; accepted for publication: 30 March 2007.

estimation (Smith and Eisner 2005) have been applied to structured models. Weighted grammars learned in this way differ in two important ways from traditional, generative models. First, the weights can be any positive value; they need not sum to one. Second, features can "overlap," and it can be difficult to design a generative model that uses such features. The benefits of new features and discriminative training methods are widely documented and recognized.

This article focuses specifically on the first of these differences. It compares the expressive power of Weighted Context-Free Grammars (WCFGs), where each rule is associated with a positive weight, to that of the corresponding Probabilistic Context-Free Grammars (PCFGs), i.e., with the same rules but where the weights of the rules expanding a nonterminal must sum to one.

One might expect that since normalization removes one or more degrees of freedom, unnormalized models should be more expressive than normalized, probabilistic models. Perhaps counter-intuitively, previous work has shown that the classes of probability distributions defined by WCFGs and PCFGs are the same (Chi 1999; Abney, McAllester, and Pereira 1999).

However, this result does not completely settle the question about the expressive power of WCFGs and PCFGs. As we show below, a WCFG can define a *conditional distribution* from strings to trees even if it does not define a probability distribution over trees. Since these conditional distributions are what are used in classification tasks and related tasks such as parsing, we need to know the relationship between the classes of *conditional distributions* defined by WCFGs and PCFGs. In fact we extend the results of Chi and of Abney et al., and show that WCFGs and PCFGs both define the same class of conditional distribution. Moreover, we present an algorithm for converting an arbitrary WCFG that defines a conditional distribution over trees given strings but possibly without a finite partition function into a PCFG with the same rules as the WCFG and that defines the same conditional distribution over trees given strings.

This means that maximum conditional likelihood WCFGs are non-identifiable, since there are an infinite number of rule weights all of which maximize the conditional likelihood.

1. Weighted CFGs

A CFG G is a tuple $\langle N, S, \Sigma, R\rangle$ where N is a finite set of nonterminal symbols, $S \in N$ is the start symbol, Σ is a finite set of terminal symbols (disjoint from N), and R is a set of production rules of the form $X \rightarrow \alpha$ where $X \in N$ and $\alpha \in (N \cup \Sigma)^\star$. A WCFG associates a positive number called the *weight* with each rule in R.[1] We denote by $\theta_{X\rightarrow\alpha}$ the weight attached to the rule $X \rightarrow \alpha$, and the vector of rule weights by $\Theta = \{\theta_{A\rightarrow\alpha} : A \rightarrow \alpha \in R\}$. A weighted grammar is the pair $G_\Theta = \langle G, \Theta\rangle$.

Unless otherwise specified, we assume a fixed underlying context-free grammar G. Let $\Omega(G)$ be the set of (finite) trees that G generates. For any $\tau \in \Omega(G)$, the *score* $s_\Theta(\tau)$ of τ is defined as follows:

$$s_\Theta(\tau) \;=\; \prod_{(X\rightarrow\alpha)\in R} (\theta_{X\rightarrow\alpha})^{f(X\rightarrow\alpha;\tau)} \tag{1}$$

where $f(X \rightarrow \alpha; \tau)$ is the number of times $X \rightarrow \alpha$ is used in the derivation of the tree τ.

1 Assigning a weight of zero to a rule equates to excluding it from R.

The *partition function* $Z(\Theta)$ is the sum of the scores of the trees in $\Omega(G)$.

$$Z(\Theta) = \sum_{\tau \in \Omega(G)} s_\Theta(\tau)$$

Since we have imposed no constraints on Θ, the partition function need not equal one; indeed, as we show below the partition function need not even exist. If $Z(\Theta)$ is finite then we say that the WCFG is *convergent*, and we can define a Gibbs probability distribution over $\Omega(G)$ by dividing by $Z(\Theta)$:

$$\mathrm{P}_\Theta(\tau) = \frac{s_\Theta(\tau)}{Z(\Theta)}$$

A *probabilistic* CFG, or PCFG, is a WCFG in which the sum of the weights of the rules expanding each nonterminal is one:

$$\forall X \in N, \sum_{(X \to \alpha) \in R} \theta_{X \to \alpha} = 1 \tag{2}$$

It is easy to show that if G_Θ is a PCFG then $Z(\Theta) \leq 1$. A *tight PCFG* is a PCFG G_Θ for which $Z(\Theta) = 1$. Necessary conditions and sufficient conditions for a PCFG to be tight are given in several places, including Booth and Thompson (1973) and Wetherell (1980).

We now describe the results of Chi (1999) and Abney, McAllester, and Pereira (1999). Let $\mathcal{G} = \{G_\Theta\}$ denote the set of the WCFGs based on the CFG G (i.e., the WCFGs in $\mathcal{G}$ all have the same underlying grammar G but differ in their rule weight vectors Θ). Let $\mathcal{G}_{Z<\infty}$ be the subset of $\mathcal{G}$ for which the partition function $Z(\Theta)$ is finite, and let $\mathcal{G}_{Z=\infty} = \mathcal{G} \setminus \mathcal{G}_{Z<\infty}$ be the subset of $\mathcal{G}$ with an infinite partition function. Further let $\mathcal{G}_{\text{PCFG}}$ denote the set of PCFGs based on G. In general, $\mathcal{G}_{\text{PCFG}}$ is a proper subset of $\mathcal{G}_{Z<\infty}$, i.e., every probabilistic context-free grammar is also a weighted context-free grammar, but because there are weight vectors Θ that don't obey Equation 2, not all WCFGs are PCFGs.

However, this does not mean that WCFGs are more expressive than PCFGs. As noted above, the WCFGs $\mathcal{G}_{Z<\infty}$ define Gibbs distributions. Again, for a fixed G, let $\mathcal{P}_{Z<\infty}$ be the probability distributions over the trees $\Omega(G)$ defined by the WCFGs $\mathcal{G}_{Z<\infty}$ and let $\mathcal{P}_{\text{PCFG}}$ be the probability distributions defined by the PCFGs $\mathcal{G}_{\text{PCFG}}$. Chi (1999, Proposition 4) and Abney, McAllester, and Pereira (1999, Lemma 5) showed that $\mathcal{P}_{Z<\infty} = \mathcal{P}_{\text{PCFG}}$, i.e., that every WCFG probability distribution is in fact generated by some PCFG. There is no "$\mathcal{P}_{Z=\infty}$" because there is no finite normalizing term $Z(\Theta)$ for such WCFGs.

1.1 Chi's Algorithm for converting WCFGs to equivalent PCFGs

Chi (1999) describes an algorithm for converting a WCFG to an equivalent PCFG. Let G_Θ be a WCFG in $\mathcal{G}_{Z<\infty}$. If $X \in N$ is a nonterminal, let $\Omega^X(G)$ be the set of trees rooted in X that can be built using G. Then define:

$$Z^X(\Theta) = \sum_{\tau \in \Omega^X(G)} s_\Theta(\tau)$$

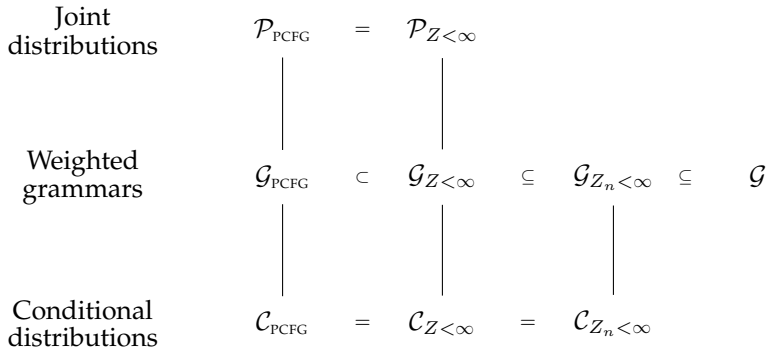


Figure 1
A graphical depiction of the primary result of this paper. Given a fixed set of productions, $\mathcal{G}$ is the set of Weighted Context-Free Grammars (WCFGs) with exactly those productions (i.e., they vary only in the production weights), $\mathcal{G}_{Z<\infty}$ is the subset of $\mathcal{G}$ that define (joint) probability distributions over trees (i.e., that have a finite partition function Z) and $\mathcal{P}_{Z<\infty}$ is the set of probability distributions defined by grammars in $\mathcal{G}_{Z<\infty}$. Chi and Geman (1998) and Abney, McAllester and Pereira (1999) proved that $\mathcal{P}_{Z<\infty}$ is the same as $\mathcal{P}_{\text{PCFG}}$, the set of probability distributions defined by the Probabilistic Context-Free Grammars $\mathcal{G}_{\text{PCFG}}$ with the same productions as $\mathcal{G}$. Thus even though the set of WCFGs properly includes the set of PCFGs, WCFGs define exactly the same probability distributions over trees as PCFGs. This paper extends these results to conditional distributions over trees conditioned on their strings. Even though the set $\mathcal{G}_{Z_n<\infty}$ of WCFGs that define conditional distributions may be larger than $\mathcal{G}_{Z<\infty}$ and properly includes $\mathcal{G}_{\text{PCFG}}$, the set of conditional distributions $\mathcal{C}_{Z_n<\infty}$ defined by $\mathcal{G}_{Z_n<\infty}$ is equal to the set of conditional distributions $\mathcal{C}_{\text{PCFG}}$ defined by PCFGs. Our proof is constructive: we give an algorithm which takes as input a WCFG $G \in \mathcal{G}_{Z_n<\infty}$ and returns a PCFG which defines the same conditional distribution over trees given strings as G.

For simplicity, let $Z^t(\Theta) = 1$ for all $t \in \Sigma$. Chi demonstrated that $G_\Theta \in \mathcal{G}_{Z<\infty}$ implies that $Z^X(\Theta)$ is finite for all $X \in N \cup \Sigma$.

For every rule $X \rightarrow \alpha$ in R define:

$$\theta'_{X\rightarrow\alpha} = \frac{\theta_{X\rightarrow\alpha} \prod_{i=1}^{|\alpha|} Z^{\alpha_i}(\Theta)}{Z^X(\Theta)}$$

where α_i is the ith element of α and $|\alpha|$ is the length of α. Chi proved that $G_{\Theta'}$ is a PCFG and that $\mathrm{P}_{\Theta'}(\tau) = s_\Theta(\tau)/Z(\Theta)$ for all trees $\tau \in \Omega(G)$.

Chi did not describe how to compute the nonterminal-specific partition functions $Z^X(\Theta)$. The $Z^X(\Theta)$ are related by equations of the form:

$$Z^X(\Theta) = \sum_{\alpha: X\rightarrow\alpha\in R} \theta_{X\rightarrow\alpha} \prod_{i=1}^{|\alpha|} Z^{\alpha_i}(\Theta)$$

which constitute a set of non-linear polynomial equations in $Z^X(\Theta)$. While a numerical solver might be employed to find the $Z^X(\Theta)$, we have found that in practice iterative

propagation of weights following the method described by Stolcke (1995, Section 4.7.1) converges quickly when $Z(\Theta)$ is finite.

2. Classifiers and Conditional Distributions

A common application of weighted grammars is parsing. One way to select a parse tree for a sentence x is to choose the maximum weighted parse that is consistent with the observation x:

$$\tau^*(x) = \operatorname*{argmax}_{\tau \in \Omega(G): y(\tau) = x} s_\Theta(\tau) \tag{3}$$

where $y(\tau)$ is the yield of τ. Other decision criteria exist, including minimum-loss decoding and re-ranked n-best decoding. All of these classifiers use some kind of dynamic programming algorithm to optimize over trees, and they also exploit the *conditional* distribution of trees given sentence observations. A WCFG defines such a conditional distribution as follows:

$$\mathrm{P}_\Theta(\tau \mid x) \;=\; \frac{s_\Theta(\tau)}{\sum_{\tau' \in \Omega(G): y(\tau') = x} s_\Theta(\tau')} = \frac{s_\Theta(\tau)}{Z_x(\Theta)} \tag{4}$$

where $Z_x(\Theta)$ is the sum of scores for all parses of x. Note that the above will be ill-defined when $Z_x(\Theta)$ diverges. Because $Z_x(\Theta)$ is constant for a given x, solving Equation 3 is equivalent to choosing τ to maximize $\mathrm{P}_\Theta(\tau \mid x)$.

We turn now to classes of these conditional distribution families. Let $\mathcal{C}_{Z<\infty}$ ($\mathcal{C}_{\text{PCFG}}$) be the class of conditional distribution families that can be expressed by grammars in $\mathcal{G}_{Z<\infty}$ ($\mathcal{G}_{\text{PCFG}}$, respectively). It should be clear that, because $\mathcal{P}_{Z<\infty} = \mathcal{P}_{\text{PCFG}}$, $\mathcal{C}_{Z<\infty} = \mathcal{C}_{\text{PCFG}}$ since a conditional family is derived by normalizing a joint distribution by its marginals.

We now define another subset of $\mathcal{G}$. Let $\mathcal{G}_{Z_n<\infty}$ contain every WCFG $G_\Theta = \langle G, \Theta \rangle$ such that, for all $n \geq 0$,

$$Z_n(\Theta) = \sum_{\tau \in \Omega(G): |y(\tau)| = n} s_\Theta(\tau) < \infty \tag{5}$$

(Note that, to be fully rigorous, we should quantify n in $\mathcal{G}_{Z_n<\infty}$, writing "$\mathcal{G}_{\forall n Z_n(\Theta)<\infty}$." We use the abbreviated form to keep the notation crisp.) For any $G_\Theta \in \mathcal{G}_{Z_n<\infty}$, it also follows that, for any $x \in L(G)$, $Z_x(\Theta) < \infty$; the converse holds as well.

It follows that any WCFG in $\mathcal{G}_{Z_n<\infty}$ can be used to construct a conditional distribution of trees given the sentence, for any sentence $x \in L(G)$. To do so, we only need to normalize $s_\Theta(\tau)$ by $Z_x(\Theta)$ (Equation 4). Let $\mathcal{G}_{Z_n=\infty}$ contain the WCFGs where some $Z_n(\Theta)$ diverge; this is a subset of $\mathcal{G}_{Z=\infty}$.[2] To see that $\mathcal{G}_{Z=\infty} \cap \mathcal{G}_{Z_n<\infty} \neq \emptyset$, consider Example 1.

Example 1

$$\theta_{A \to A\ A} = 1, \qquad \theta_{A \to a} = 1$$

2 Here, full rigor would require quantification of n, writing "$\mathcal{G}_{\exists n Z_n(\Theta)=\infty}$."

This grammar produces binary structures over strings in a^+. Every such tree receives score 1. Since there are infinitely many trees, $Z(\Theta)$ diverges. But for any fixed string a^n, the number of parse trees is finite. This grammar defines a uniform conditional distribution over all binary trees, given the string.

For a grammar G_Θ to be in $\mathcal{G}_{Z_n<\infty}$, it is sufficient that, for every nonterminal $X \in N$, the sum of scores of all cyclic derivations $X \Rightarrow^+ X$ be finite. Conservatively, this can be forced by eliminating epsilon rules and unary rules or cycles altogether, or by requiring the sum of cyclic derivations for every nonterminal X to sum to strictly less than one. Example 2 gives a grammar in $\mathcal{G}_{Z_n=\infty}$ with a unary cyclic derivation that does not "dampen."

Example 2

$$\theta_{A\to A\ A} = 1, \qquad \theta_{A\to A} = 1, \qquad \theta_{A\to a} = 1$$

For any given a^n, there are infinitely many equally-weighted parse trees, so even the set of trees for a^n cannot be normalized into a distribution ($Z_n(\Theta) = \infty$). Generally speaking, if there exists a string $x \in L(G)$ such that the set of trees that derive x is not finite (i.e., there is no finite bound on the number of derivations for strings in $L(G)$; the grammar in Example 2 is a simple example), then $\mathcal{G}_{Z_n<\infty}$ and $\mathcal{G}_{Z<\infty}$ are separable.[3]

For a given CFG G, a conditional distribution over trees given strings is a function $\Sigma^* \to (\Omega(G) \to [0,1])$. Our notation for the set of conditional distributions that can be expressed by $\mathcal{G}_{Z_n<\infty}$ is $\mathcal{C}_{Z_n<\infty}$. Note that there is no "$\mathcal{C}_{Z_n=\infty}$" since an infinite $Z_n(\Theta)$ implies an infinite $Z(x)$ for some sentence x and therefore an ill-formed conditional family. Indeed, it is difficult to imagine a scenario in computational linguistics in which non-dampening cyclic derivations (WCFGs in $\mathcal{G}_{Z_n=\infty}$) are desirable, since no linguistic explanations depend crucially on arbitrary lengthening of cyclic derivations.

We now state our main theorem.

Theorem 1
For a given CFG G, $\mathcal{C}_{Z_n<\infty} = \mathcal{C}_{Z<\infty}$.

Proof 1
Suppose we are given weights Θ for G such that $G_\Theta \in \mathcal{G}_{Z_n<\infty}$. We will show that the sequence $Z_1(\Theta), Z_2(\Theta), ...$ is bounded by an exponential function of n, then describe a transformation on Θ resulting in a new grammar, $G_{\Theta'}$ that is in $\mathcal{G}_{Z<\infty}$ and defines the same family of conditional distributions (i.e., $\forall \tau \in \Omega(G), \forall x \in L(G), \mathrm{P}_\Theta(\tau \mid x) = \mathrm{P}_{\Theta'}(\tau \mid x)$).

First we prove that for all $n \geq 1$ there exists some c such that $Z_n(\Theta) \leq c^n$. Given G_Θ, we construct $\bar{G}_{\bar{\Theta}}$ in CNF that preserves the total score for any $x \in L(G)$. The existence of $\bar{G}_{\bar{\Theta}}$ was demonstrated by Goodman (1998, Section 2.6), who gives an algorithm for constructing the value-preserving weighted grammar $\bar{G}_{\bar{\Theta}}$ from G_Θ.

Note that $\bar{G} = \langle \bar{N}, S, \Sigma, \bar{R} \rangle$, containing possibly more nonterminals and rules than G. The set of (finite) trees $\Omega\left(\bar{G}\right)$ is different from $\Omega(G)$; the new trees must be binary and may include new nonterminals.

3 We are grateful to an anonymous reviewer for pointing this out, and an even stronger point: for a given G, $\mathcal{G}$ and $\mathcal{G}_{Z_n<\infty}$ have a non-empty set-difference if and only if G has infinite ambiguity (some $x \in L(G)$ has infinitely many parse trees).

Next, collapse the nonterminals in $\bar{N}$ into one nonterminal, S. The resulting grammar is $\breve{G}_{\breve{\Theta}} = \langle\langle\{S\}, S, \Sigma, \breve{R}\rangle, \breve{\Theta}\rangle$. $\breve{R}$ contains the rule $S \rightarrow S\ S$ and rules of the form $S \rightarrow a$ for $a \in \Sigma$. The weights of these rules are

$$\breve{\theta}_{S \rightarrow S\ S} = \beta = \max(1, \sum_{(X \rightarrow Y\ Z) \in \bar{R}} \bar{\theta}_{X \rightarrow Y\ Z}) \tag{6}$$

$$\breve{\theta}_{S \rightarrow a} = \upsilon = \max(1, \sum_{(X \rightarrow b) \in \bar{R}} \bar{\theta}_{X \rightarrow b}) \tag{7}$$

The grammar $\breve{G}_{\breve{\Theta}}$ will allow every tree allowed by $\bar{G}_{\bar{\Theta}}$ (modulo labels on nonterminal nodes, which are now all S). It may allow some additional trees. The score of a tree under $\breve{G}_{\breve{\Theta}}$ will be at least as great as the sum of scores of all structurally-equivalent trees under $\bar{G}_{\bar{\Theta}}$, because β and υ are defined to be large enough to absorb all such scores. It follows that, for all $x \in L(G)$:

$$s_{\breve{\Theta}}(x) \geq s_{\bar{\Theta}}(x) = s_{\Theta}(x) \tag{8}$$

Summing over all trees of any given yield length n, we have

$$Z_n(\breve{\Theta}) \geq Z_n(\bar{\Theta}) = Z_n(\Theta) \tag{9}$$

$\breve{G}$ generates all possible binary trees (with internal nodes undifferentiated) over a given sentence x in $L(G)$. Every tree generated by $\breve{G}$ with yield length n will have the same score: $\beta^{n-1}\upsilon^n$, since every binary tree with n terminals has exactly $n-1$ nonterminals. Each tree corresponds to a way of bracketing n items, so the total number of parse trees generated by $\breve{G}$ for a string of length n is the number of different ways of bracketing a sequence of n items. The total number of unlabeled binary bracketings of an n-length sequence is the nth Catalan number C_n (Graham, Knuth, and Patashnik 1994), which in turn is bounded above by 4^n (Vardi 1991). The total number of strings of length n is $|\Sigma|^n$. Therefore

$$Z_n(\breve{\Theta}) = C_n|\Sigma|^n\beta^{n-1}\upsilon^n \leq 4^n|\Sigma|^n\beta^{n-1}\upsilon^n \leq (4|\Sigma|\beta\upsilon)^n \tag{10}$$

We now transform the original weights Θ as follows. For every rule $(X \rightarrow \alpha) \in R$, let

$$\theta'_{X \rightarrow \alpha} \leftarrow \frac{\theta_{X \rightarrow \alpha}}{(8|\Sigma|\beta\upsilon)^{t(\alpha)}} \tag{11}$$

where $t(\alpha)$ is the number of Σ symbols appearing in α. The above transformation results in every n-length sentence having its score divided by $(8|\Sigma|\beta\upsilon)^n$. The relative scores of trees with the same yield are unaffected, because they are all scaled equally. Therefore $G_{\Theta'}$ defines the same conditional distribution over trees given sentences as G_{Θ}, which implies that G_{Θ} and $G_{\Theta'}$ have the same highest scoring parses. Note that *any* sufficiently large value could stand in for $8|\Sigma|\beta\upsilon$ to both (a.) preserve the conditional distribution and (b.) force $Z_n(\Theta)$ to converge. We have not found the *minimum* such value, but $8|\Sigma|\beta\upsilon$ is sufficiently large.

The sequence of $Z_n(\Theta)$ now converges:

$$Z_n(\Theta') \leq \frac{Z_n(\Theta)}{(8|\Sigma|\beta v)^n} \leq \left(\frac{1}{2}\right)^n \tag{12}$$

Hence $Z(\Theta') = \sum_{n=0}^{\infty} Z_n(\Theta') \leq 2$ and $G_{\Theta'} \in \mathcal{G}_{Z<\infty}$. ■

Corollary 1
Given a CFG G, $\mathcal{C}_{Z_n<\infty} = \mathcal{C}_{\text{PCFG}}$.

Proof 2
By Theorem 1, $\mathcal{C}_{Z_n<\infty} = \mathcal{C}_{Z<\infty}$. We know that $\mathcal{P}_{Z<\infty} = \mathcal{P}_{\text{PCFG}}$, from which it follows that $\mathcal{C}_{Z<\infty} = \mathcal{C}_{\text{PCFG}}$. Hence $\mathcal{C}_{Z_n<\infty} = \mathcal{C}_{\text{PCFG}}$. To convert a WCFG in $\mathcal{C}_{Z_n<\infty}$ into a PCFG, first apply the transformation in the proof of Theorem 1 to get a convergent WCFG, then apply Chi's method (our Section 1.1). ■

3. HMMs and Related Models

Hidden Markov models (HMMs) are a special case of PCFGs. The structures they produce are labeled sequences, which are equivalent to right-branching trees. We can write an HMM as a PCFG with restricted types of rules. We will refer to the unweighted, finite-state grammars that HMMs stochasticize as "right-linear grammars." Rather than using the production rule notation of PCFGs, we will use more traditional HMM notation and refer to states (interchangeable with nonterminals) and paths (interchangeable with parse trees).

In the rest of the paper we distinguish between hidden Markov models (HMMs), which are probabilistic finite-state automata locally normalized just like a PCFG, and chain-structured Markov random fields (MRFs; Section 3.1), in which moves or transitions are associated with positive weights and which are globally normalized like a WCFG.[4] We also distinguish two different types of dependency structures in these automata. Abusing the standard terminology somewhat, in a Mealy automaton arcs are labeled with output or terminal symbols, while in a Moore automaton the states emit terminal symbols.[5]

A Mealy HMM defines a probability distribution over pairs $\langle \vec{x}, \vec{\pi} \rangle$, where $\vec{x}$ is a length-n sequence $\langle x_1, x_2, ..., x_n \rangle \in \Sigma^n$ and $\vec{\pi} = \langle \pi_0, \pi_1, \pi_2, ..., \pi_n \rangle \in N^{n+1}$ is a state (or nonterminal) path. The distribution is given by

$$\mathrm{P}_{\text{HMM}}(\vec{x}, \vec{\pi}) = \left(\prod_{i=1}^{n} p(x_i, \pi_i \mid \pi_{i-1}) \right) p(\text{STOP} \mid \pi_n) \tag{13}$$

π_0 is assumed, for simplicity, to be constant and known; we also assume that every state transition emits a symbol (no ϵ arcs), an assumption made in typical tagging and chunking applications of HMMs. We can convert a Mealy HMM to a PCFG by including, for every tuple $\langle x, \pi, \phi \rangle$ ($x \in \Sigma$ and $\pi, \phi \in N$) such that $p(x, \pi \mid \phi) > 0$, the rule $\pi \rightarrow x\ \phi$,

4 We admit that these names are somewhat misleading, since as we will show, chain-structured MRFs also have the Markov property and define the same joint and conditional distributions as HMMs.

5 In formal language theory both Mealy and Moore machines are finite-state transducers (Mealy 1955; Moore 1956); we ignore the input symbols here.

with the same probability as the corresponding HMM transition. For every π such that $p(\text{STOP} \mid \pi)$, we include the rule $\pi \rightarrow \epsilon$, with probability $p(\text{STOP} \mid \pi)$.

A Moore HMM factors the distribution $p(x, \pi \mid \phi)$ into $p(x \mid \pi) \cdot p(\pi \mid \phi)$. A Moore HMM can be converted to a PCFG by adding a new nonterminal $\bar{\pi}$ for every state π and including the rules $\phi \rightarrow \bar{\pi}$ (with probability $p(\pi \mid \phi)$) and $\bar{\pi} \rightarrow x\ \pi$ (with probability $p(x \mid \pi)$). Stop probabilities are added as in the Mealy case. For a fixed number of states, Moore HMMs are less probabilistically expressive than Mealy HMMs, though we can convert between the two with a change in the number of states.

We consider Mealy HMMs primarily from here on. If we wish to define the distribution over paths given words, we conditionalize:

$$\mathrm{P}_{\text{HMM}}(\vec{\pi} \mid \vec{x}) = \frac{\left(\prod_{i=1}^{n} p(x_i, \pi_i \mid \pi_{i-1})\right) p(\text{STOP} \mid \pi_n)}{\sum_{\vec{\pi}' \in N^{n+1}} \left(\prod_{i=1}^{n} p(x_i, \pi'_i \mid \pi'_{i-1})\right) p(\text{STOP} \mid \pi'_n)} \tag{14}$$

This is how scores are assigned when selecting the best path given a sequence.

For a grammar G that is right-linear, we can therefore talk about the set of HMM (right-linear) grammars $\mathcal{G}_{\text{HMM}}$, the set of probability distributions $\mathcal{P}_{\text{HMM}}$ defined by those grammars, and $\mathcal{C}_{\text{HMM}}$, the set of conditional distributions over state paths (trees) that they define.[6]

3.1 Mealy Markov random fields

When the probabilities in Mealy HMMs are replaced by arbitrary positive weights, the production rules can be seen as features in a Gibbs distribution. The resulting model is a type of Markov random field (MRF) with a chain structure; these have recently become popular in natural language processing (Lafferty, McCallum, and Pereira 2001). Lafferty et al.'s formulation defined a conditional distribution over paths given sequences by normalizing for each sequence $\vec{x}$:

$$\mathrm{P}_{\text{cMRF}}(\vec{\pi} \mid \vec{x}) = \frac{\left(\prod_{i=1}^{n} \theta_{\pi_{i-1}, x_i, \pi_i}\right) \theta_{\pi_n, \text{STOP}}}{Z_x(\Theta)} \tag{15}$$

Using a single normalizing term $Z(\Theta)$, we can also define a *joint* distribution over states and paths:

$$\mathrm{P}_{\text{cMRF}}(\vec{x}, \vec{\pi}) = \frac{\left(\prod_{i=1}^{n} \theta_{\pi_{i-1}, x_i, \pi_i}\right) \theta_{\pi_n, \text{STOP}}}{Z(\Theta)} \tag{16}$$

Let $\mathcal{G} = \{G_\Theta\}$ denote the set of weighted grammars based on the unweighted right-linear grammar G. We call these weighted grammars "Mealy MRFs." As in the WCFG case, we can add the constraint $Z_n(\Theta) < \infty$ (for all n), giving the class $\mathcal{G}_{Z_n < \infty}$.

Recall that, in the WCFG case, the move from $\mathcal{G}$ to $\mathcal{G}_{Z_n < \infty}$ had to do with cyclic derivations. The analogous move in the right-linear grammar case involves ϵ emissions

6 Of course, the right-linear grammar is a CFG, so we could also use the notation $\mathcal{G}_{\text{PCFG}}$, $\mathcal{P}_{\text{PCFG}}$, and $\mathcal{C}_{\text{PCFG}}$.

(production rules of the form $X \rightarrow Y$). If, as in typical applications of finite-state models to natural language processing, there are no rules of the form $X \rightarrow Y$, then $\mathcal{G}_{Z_n<\infty}$ is empty and $\mathcal{G}_{Z_n<\infty} = \mathcal{G}$. Our formulae above, in fact, assume that there are no ϵ emissions.

Because Mealy MRFs are a special case of WCFGs, Theorem 1 applies to them. This means that any random field using Mealy HMM features (Mealy MRF) such that $\forall n, Z_n(\Theta) < \infty$ can be transformed into a Mealy HMM that defines the same conditional distribution of tags given words:[7]

Corollary 2
For a given right-linear grammar G, $\mathcal{C}_{\text{HMM}} = \mathcal{C}_{Z<\infty} = \mathcal{C}_{Z_n<\infty}$.

Lafferty, McCallum, and Pereira's *conditional* random fields are typically trained to optimize a different objective function than HMMs (conditional likelihood and joint likelihood, respectively). Our result shows that optimizing either objective on the set of Mealy HMMs as opposed to Mealy MRFs will achieve the same result, modulo imperfections in the numerical search for parameter values.

3.2 Maximum-entropy Markov models

While HMMs and chain MRFs represent the same set of conditional distributions, we can show that the "maximum-entropy Markov models" (MEMMs) of McCallum, Freitag, and Pereira (2000) represent a strictly smaller class of distributions.

An MEMM is a similar model with a different event structure. It defines the distribution over paths *given* words as:

$$\mathrm{P}_{\text{MEMM}}(\vec{\pi} \mid \vec{x}) = \prod_{i=1}^{n} p(\pi_i \mid \pi_{i-1}, x_i) \tag{17}$$

Unlike an HMM, the MEMM does *not* define a distribution over output sequences x. The name "maximum entropy Markov model" comes from the fact that the conditional distributions $p(\cdot \mid \pi, x)$ typically have a log-linear form, rather than a multinomial form, and are trained to maximize entropy.

Lemma 1
For every MEMM, there is a Mealy MRF that represents the same conditional distribution over paths given symbols.

Proof 3
By definition, the features of the MRF include triples $\langle \pi_{i-1}, x_i, \pi_i \rangle$. Assign to the weight $\theta_{\pi_i, x_j, \pi_k}$ the value $p_{\text{MEMM}}(\pi_i \mid \pi_k, x_j)$. Assign to $\theta_{\pi_i, \text{STOP}}$ the value 1. In computing $\mathrm{P}_{\text{cMRF}}(\pi \mid x)$ (Equation 15), the normalizing term for each x will be equal to 1. ■

MEMMs, like HMMs, are defined by locally-normalized conditional multinomial distributions. This has computational advantages (no potentially-infinite $Z(\Theta)$ terms to compute). However, the set of conditional distributions of labels given terminals that

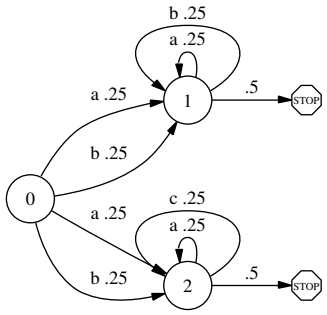
7 What if we allow additional features? It can be shown that, as long as the vocabulary Σ is finite and known, we can convert any such MRF with potential functions on state transitions and emissions into an HMM functioning equivalently as a classifier. If Σ is not fully known, then we cannot sum over all emissions from each state, and we cannot use Chi's method (Section 1.1) to convert to a PCFG (HMM).

can be expressed by MEMMs is strictly smaller than those expressible by HMMs (and by extension, Mealy MRFs).

Theorem 2
For a given right-linear grammar G, $\mathcal{C}_{\text{MEMM}} \subset \mathcal{C}_{\text{HMM}}$.

Proof 4
We give an example of a Mealy HMM whose conditional distribution over paths (trees) given sentences cannot be represented by an MEMM. We thank Michael Collins for pointing out to us the existence of examples like this one. Define a Mealy HMM with three states named 0, 1, and 2, over an alphabet $\{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$, as follows. State 0 is the start state.

Example 3



Under this model, $\mathrm{P}_{\text{HMM}}(0, 1, 1 \mid \mathrm{a}, \mathrm{b}) = \mathrm{P}_{\text{HMM}}(0, 2, 2 \mid \mathrm{a}, \mathrm{c}) = 1$. These conditional distributions cannot both be met by any MEMM. To see why, consider:

$$p(1 \mid 0, \mathrm{a}) \cdot p(1 \mid 1, \mathrm{b}) = p(2 \mid 0, \mathrm{a}) \cdot p(2 \mid 2, \mathrm{c}) = 1$$

implies that

$$p(1 \mid 0, \mathrm{a}) = p(1 \mid 1, \mathrm{b}) = p(2 \mid 0, \mathrm{a}) = p(2 \mid 2, \mathrm{c}) = 1$$

But it is impossible for $p(1 \mid 0, \mathrm{a}) = p(2 \mid 0, \mathrm{a}) = 1$. This holds regardless of the form of the distribution $p(\cdot \mid \pi, x)$ (e.g., multinomial or log-linear).

Since $\mathrm{P}(0, 1, 1 \mid \mathrm{a}, \mathrm{b}) = \mathrm{P}(0, 2, 2 \mid \mathrm{a}, \mathrm{c})$ cannot be met by any MEMM, there are distributions in the family allowed by HMMs that cannot be expressed as MEMMs, and the latter are less expressive. ■

It is important to note that the above result applies to Mealy HMMs; our result compares models with the same dependencies among random variables. If the HMM's distribution $p(x_i, \pi_i \mid \pi_{i-1})$ is factored into $p(x_i \mid \pi_i) \cdot p(\pi_i \mid \pi_{i-1})$ (i.e., it is a Moore HMM), then there may exist a MEMM with the same number of states that can represent some distributions that the Moore HMM cannot.[8]

8 The HMM shown in Example 3 can be factored into a Moore HMM without any change to the distribution.

One can also imagine MEMMs in which $p(\pi_i \mid \pi_{i-1}, x_i, ...)$ is conditioned on *more* surrounding context (x_{i-1} or x_{i+1}, or the entire sequence $\vec{x}$, for example). Conditioning on more context can be done by increasing the *order* of the Markov model—all of our models so far have been first-order, with a memory of only the previous state. Our result can be extended to include higher-order MEMMs. Suppose we allow the MEMM to "look ahead" n words, factoring its distribution into $p(\pi_i \mid \pi_{i-1}, x_i, x_{i+1}, ..., x_{i+n})$.

Corollary 3
A first-order Mealy HMM can represent some classifiers that no MEMM with finite lookahead can represent.

Proof 5
Consider again Example 3. Note that, for all $m \geq 1$, it sets

$$\mathrm{P}_{\text{HMM}}(0, \overbrace{1, ..., 1}^{m\ 1\text{'s}} \mid \mathrm{a}^m\mathrm{b}) = 1$$
$$\mathrm{P}_{\text{HMM}}(0, \underbrace{2, ..., 2}_{m\ 2\text{'s}} \mid \mathrm{a}^m\mathrm{c}) = 1$$

Suppose we wish to capture this in a MEMM with n symbols of look-ahead. Letting $m = n + 1$,

$$p(1 \mid 0, \mathrm{a}^{n+1}) \cdot p(1 \mid 1, \mathrm{a}^n\mathrm{b}) \cdot \prod_{i=1}^{n} p(1 \mid 1, \mathrm{a}^{n-i}\mathrm{b}) = 1$$
$$p(2 \mid 0, \mathrm{a}^{n+1}) \cdot p(2 \mid 2, \mathrm{a}^n\mathrm{c}) \cdot \prod_{i=1}^{n} p(2 \mid 2, \mathrm{a}^{n-i}\mathrm{c}) = 1$$

The same issue arises as in the proof of Theorem 2: it cannot be that $p(1 \mid 0, \mathrm{a}^{n+1}) = p(2 \mid 0, \mathrm{a}^{n+1}) = 1$, and so this MEMM does not exist. Note that even if we allow the MEMM to "look back" and condition on earlier symbols (or states), it cannot represent the distribution in Example 3. ■

Generally speaking, this limitation of MEMMs has nothing to do with the estimation procedure (we have committed to no estimation procedure in particular) but rather with the conditional *structure* of the model. That some model structures work better than others at real NLP tasks was discussed by Johnson (2001) and Klein and Manning (2002). Our result—that the class of distributions allowed by MEMMs is a strict subset of those allowed by Mealy HMMs—makes this unsurprising.

4. Practical Implications

Our result is that weighted generalizations of classical probabilistic grammars (PCFGs and HMMs) are *no more powerful* than the probabilistic models. This means that, insofar as log-linear models for NLP tasks like tagging and parsing are more successful than their probabilistic cousins, it is due to either (a.) additional features added to the model, (b.) improved estimation procedures (e.g., maximum conditional likelihood estimation or contrastive estimation), or both. (Note that the choice of estimation procedure (b) is in principal orthogonal to the choice of model, and conditional estimation should

not be conflated with log-linear modeling.) For a given estimation criterion, weighted CFGs and Mealy MRFs, in particular, cannot be expected to behave any differently than PCFGs and HMMs, respectively, unless they are augmented with more features.

5. Related Work

Abney, McAllester, and Pereira (1999) addressed the relationship between PCFGs and probabilistic models based on push-down automaton operations (e.g., the structured language model of Chelba and Jelinek, 1998). They proved that, while the conversion may not be simple (indeed, a blow-up in the automaton's size may be incurred), given G, $\mathcal{P}_{\text{PCFG}}$ and the set of distributions expressible by shift-reduce probabilistic push-down automata are weakly equivalent. Importantly, the standard conversion of a CFG into a *shift-reduce* PDA, when applied in the *stochastic* case, does not always preserve the probability distribution over trees. Our Theorem 2 bears a resemblance to that result. Further work on the relationship between weighted CFGs and weighted PDAs is described in Nederhof and Satta (2004).

MacKay (1996) proved that linear Boltzmann chains (a class of weighted models that is essentially the same as Moore MRFs) express the same set of distributions as Moore HMMs, under the condition that the Boltzmann chain has a single specific end state. MacKay avoided the divergence problem by defining the Boltzmann chain always to condition on the length of the sequence; he tacitly requires all of his models to be in $\mathcal{G}_{Z_n<\infty}$. We have suggested a more applicable notion of model equivalence (equivalence of the conditional distribution) and our Theorem 1 generalizes to context-free models.

6. Conclusion

We have shown that weighted CFGs that define finite scores for all sentences in their languages have no greater expressivity than PCFGs, when used to define distributions over trees given sentences. This implies that the standard Mealy MRF formalism is no more powerful than Mealy hidden Markov models, for instance. We have also related "maximum entropy Markov models" to Mealy Markov random fields, showing that the former is a strictly less expressive weighted formalism.

Acknowledgments

This work was supported by a Fannie and John Hertz Foundation fellowship to N. Smith at Johns Hopkins University. The views expressed are not necessarily endorsed by the sponsors. We are grateful to three anonymous reviewers for feedback that improved the article, to Michael Collins for encouraging exploration of this matter and helpful comments on a draft, and to Jason Eisner and Dan Klein for insightful conversations. Any errors are the sole responsibility of the authors.

References

Abney, Steven P., David A. McAllester, and Fernando Pereira. 1999. Relating probabilistic grammars and automata. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 542–9, College Park, Maryland, USA, June.

Booth, Taylor L. and Richard A. Thompson. 1973. Applying probability measures to abstract languages. *IEEE Transactions on Computers*, 22(5):442–450, May.

Chelba, Ciprian and Frederick Jelinek. 1998. Exploiting syntactic structure for language modeling. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 325–31, Montreal, Canada, August.

Chi, Zhiyi. 1999. Statistical properties of probabilistic context-free grammars. *Computational Linguistics*, 25(1):131–60.

Goodman, Joshua T. 1998. *Parsing Inside-Out*. Ph.D. thesis, Harvard University, May.

Graham, Ronald L., Donald E. Knuth, and Oren Patashnik. 1994. *Concrete Mathematics*. Addison-Wesley, Reading, Massachusetts, USA.

Johnson, Mark. 2001. Joint and conditional estimation of tagging and parsing models. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 314–21, Toulouse, France, July.

Johnson, Mark, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic "unification-based" grammars. In *Proceedings of the 37th Annual Conference of the Association for Computational Linguistics*, pages 535–41, College Park, Maryland, USA, June.

Klein, Dan and Christopher D. Manning. 2002. Conditional structure versus conditional estimation in NLP models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 9–16, Philadelphia, Pennsylvania, USA, July.

Lafferty, John, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–9, Williams College, Massachusetts, USA, June/July.

MacKay, David J. C. 1996. Equivalence of linear Boltzmann chains and hidden Markov models. *Neural Computation*, 8(1):178–181.

McCallum, Andrew, Dayne Freitag, and Fernando Pereira. 2000. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning*, pages 591–8, Stanford University, California, USA, June/July.

Mealy, G. H. 1955. A method for synthesizing sequential circuits. *Bell System Technology Journal*, 34:1045–1079, September.

Moore, Edward F. 1956. Gedanken-experiments on sequential machines. In *Automata Studies*, number 34 in Annals of Mathematical Studies. Princeton University Press, Princeton, New Jersey, USA, pages 129–153.

Nederhof, Mark-Jan and Giorgio Satta. 2004. Probabilistic parsing strategies. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 543–50, Barcelona, Spain, July.

Ratnaparkhi, Adwait, Salim Roukos, and R. Todd Ward. 1994. A maximum entropy model for parsing. In *Proceedings of the International Conference on Spoken Language Processing*, pages 803–806, Yokohama, Japan.

Smith, Noah A. and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 354–62, Ann Arbor, Michigan, USA, June.

Stolcke, Andreas. 1995. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–201.

Taskar, Ben, Dan Klein, Michael Collins, Daphne Koller, and Christopher Manning. 2004. Max-margin parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1–8, Barcelona, Spain, July.

Vardi, Ilan. 1991. *Computational Recreations in Mathematica*. Addison-Wesley, Redwood City, California, USA.

Wetherell, C. S. 1980. Probabilistic languages: A review and some open questions. *Computing Surveys*, 12:361–379.