The VC dimension of constraint-based grammars

Max Bane[a], Jason Riggle[a], Morgan Sonderegger[b]

[a] *University of Chicago, Department of Linguistics, 1010 East 59th Street, Chicago, IL 60637.*
[b] *University of Chicago, Department of Computer Science, 1100 East 58th Street, Chicago, IL 60637.*

Abstract

We analyze the complexity of Harmonic Grammar (HG), a linguistic model in which licit underlying-to-surface-form mappings are determined by optimization over weighted constraints. We show that the Vapnik-Chervonenkis Dimension of HG grammars with k constraints is $k-1$. This establishes a fundamental bound on the complexity of HG in terms of its capacity to classify sets of linguistic data that has significant ramifications for learnability. The VC dimension of HG is the same as that of Optimality Theory (OT), which is similar to HG, but uses ranked rather than weighted constraints in optimization. The parity of the VC dimension in these two models is somewhat surprising because OT defines finite classes of grammars—there are at most $k!$ ways to rank k constraints—while HG can define infinite classes of grammars because the weights associated with constraints are real-valued. The parity is also surprising because HG permits groups of constraints that interact through so-called 'gang effects' to generate languages that cannot be generated in OT. The fact that the VC dimension grows linearly with the number of constraints in both models means that, even in the worst case, the number of randomly chosen training samples needed to weight/rank a known set of constraints is a linear function of k. We conclude that though there may be factors that favor one model or the other, the complexity of learning weightings/rankings is not one of them.

Key words: complexity, learnability, Optimality Theory, Harmonic Grammar, VC dimension

1. Introduction

Harmonic Grammar (HG; Legendre et al., 1990; Smolensky and Legendre, 2006; Goldsmith, 1990, 1991, 1993a,b) is a constraint-based linguistic model in which the licit mappings from underlying forms to surface forms are those that optimally satisfy a linearly weighted set of constraints.[1] It is closely related to Optimality Theory (OT; Prince and Smolensky, 1993/2004), with the crucial difference that the parameters of an HG grammar are a vector of numerical constraint weights rather than a total ordering of the constraint set. For a given threshold on the number of violations, HG grammars include OT grammars in the sense that there are patterns generated by weighting that cannot be generated by rankings, while any ranking can be approximated by a weighting.

Though research in HG all but ceased with the emergence of OT, a growing body of work has been reexamining HG in a variety of cases. Weighted constraints have been argued to be desirable for describing "gang effects" (as in Potts et al., 2008) in which the action of a strong constraint may be overwhelmed by a group of weaker constraints, or multiple violations of a single weaker constraint (for a range of proposed empirical cases, see Itô and Mester, 2003; Jäger and Rosenbach, 2006; Kager and Shatzman, 2007; Farris-Trimble, 2008a,b, in press). This is straightforward in HG, but requires special mechanisms such as constraint conjunction (Smolensky, 1995; Zhang, 2007) in OT. Albright (2007, 2008) has also argued that weighted constraints are useful in analyzing gradient well-formedness judgments, which seem to show both gang effects and "anti-bottleneck" effects where violations beyond what would be the fatal violation in OT contribute to relative ill-formedness. Finally, Jesney and Tessier (2008) and Albright et al. (2008) have argued that weighted constraints better represent aspects of phonological acquisition.

Email addresses: bane@uchicago.edu (Max Bane), jriggle@uchicago.edu (Jason Riggle), morgan@cs.uchicago.edu (Morgan Sonderegger)

[1] See Pater (2009) for a thorough and accessible introduction to HG, as well as a review of previous work.

Previous computational work on HG has focused on algorithms for learning HG grammars. Potts et al. (2008) formulate HG as linear programming and use the Simplex Algorithm, Boersma and Pater (2008) uses a variant of the Perceptron Learning Algorithm, and Goldwater and Johnson (2003); Hayes and Wilson (2008) use Maximum Entropy methods. By contrast, our focus in this paper is not on any particular algorithm or method, but on a property of the class of HG grammars itself that has ramifications for *any* learning algorithm. We characterize a basic formal property of HG called the Vapnik-Chervonenkis dimension Vapnik and Chervonenkis (1971). The VC dimension of a class of grammars has implications for its learnability and expressiveness. Put simply, if a class of grammars has infinite VC dimension then it is not learnable, and if a class of grammars has finite VC dimension then its members can be learned from a sample whose size is a linear function of the VC dimension (Blumer et al., 1989). Of particular interest in models like OT and HG is the way that the dimensionality of the grammar class (i.e. the number of constraints) relates to the VC dimension.

Because constraint-based grammars for human languages are generally assumed to require many constraints, if the VC dimension of HG/OT grows too quickly with the number of constraints, then grammars will not be learnable without further restrictions. However, Riggle (2009) has shown that the VC dimension of OT grammars with k constraints is $k-1$. This tightens the inherent bound of $\log_2(k!)$ that comes from the finiteness of concept classes in OT ($k!$ rankings generate at most $k!$ languages). In HG, on the other hand, the range of positive real valued weights is infinite, so there is no *a priori* bound on the number of languages generated by k constraints and thus no *a priori* bound on the VC dimension of HG. Furthermore, because it is the case that for any given lexicon HG generates all the OT languages[2] and may also generate languages with gang effects that involve subtle interactions among groups of constraints, one might expect that HG would be harder to learn in the worst case. It turns out, however, that the VC dimension of HG with k constraints is also $k-1$. In this paper, we prove this result and discuss its ramifications.

Section 2 provides a brief overview and formal description of HG as a grammatical concept class, and illustrates the correspondence between HG tableaux and systems of linear inequalities. Section 3 introduces the Vapnik-Chervonenkis dimension and establishes that one need only consider tableaux of two candidates ("binary tableaux") in order to reason about the VC dimension of HG. Section 4 provides a mathematical description of binary tableaux, then shows that the VC dimension of HG is $k-1$ for k constraints. Section 5 concludes with a discussion of this result's implications for the learnability of HG.

2. Background

Given a lexicon D of input forms, both OT and HG define a language L over D by determining for each input $i \in D$ one or more optimal output forms according to a set of constraints $\textsc{Con} = \{c_1, \ldots, c_k\}$ and either an ordering of CON, in the case of OT, or a vector of positive real constraint weights $\vec{w} \in \mathbb{R}_+^k$ in the case of HG. Thus for fixed D and CON, an OT grammar is fully specified by any total ordering (or "ranking") of the constraints, and an HG grammar is defined by any k-dimensional non-negative real vector (or "weighting") of the constraints.

Given an alphabet Σ from which underlying forms and surface forms are constructed, each constraint $c_i \in \textsc{Con}$ is a function from "candidate" (i,o)-pairs in $\Sigma^* \times \Sigma^*$ to non-negative integers:

$$c_i : \Sigma^* \times \Sigma^* \to \mathbb{Z}_+. \tag{1}$$

For constraint $c_i \in \textsc{Con}$, if $c_i((i,o)) = m$ then we say that the candidate mapping (i,o) "violates" constraint c_i m times. For every candidate input-output mapping in $\Sigma^* \times \Sigma^*$, we can then define a vector $\vec{v} \in \mathbb{Z}_+^k$ of its violations of each constraint:

$$\vec{v} = \langle v_1, \ldots, v_k \rangle = \langle c_1((i,o)), \ldots, c_k((i,o)) \rangle. \tag{2}$$

It is in terms of these violation vectors that the grammar specifies the output(s) for each input form. The rankings and weightings of OT and HG grammars respectively impose partial orderings on $\mathbb{Z}_+^k$ according to a property called

[2] An OT tableau can contain at most $k!$ violation vectors that are not harmonically bounded. Thus, among the non-harmonically-bounded candidates for any finite lexicon there is a finite maximum number of violations for any constraint. As Bane and Riggle (to appear) point out, this maximum means that any ranking can be simulated by a constraint weighting (e.g. by using a weighting scheme like that described by Prince (2007)).

"harmony" (in the case of OT, it is in fact a total order). We refer to the harmony orderings on violation vectors as $\succ_{\mathcal{R}}$ for OT and $\succ_{\vec{w}}$ for HG; these are defined in (3) and (4).

In Optimality Theory, we denote the fact that constraint c_i outranks c_j according to constraint ranking $\mathcal{R}$ as $c_i \gg_{\mathcal{R}} c_j$ (or $c_i \gg c_j$ if $\mathcal{R}$ is clear from context). For a given $\mathcal{R}$, violation vector $\vec{u}$ is more harmonic than vector $\vec{v}$ if $\vec{u}$ has fewer violations of the highest ranked constraint for which $\vec{u}$ and $\vec{v}$ have different numbers of violations:

(3)
$$\vec{u} \succ_{\mathcal{R}} \vec{v} \text{ iff } \forall i \text{ such that } u_i > v_i,\ \exists c_j \text{ such that } c_j \gg_{\mathcal{R}} c_i \text{ and } u_j < v_j$$
or equivalently
$$\vec{u} \succ_{\mathcal{R}} \vec{v} \text{ iff } \forall i \text{ such that } (\vec{u}-\vec{v})_i > 0,\ \exists c_j \text{ such that } c_j \gg_{\mathcal{R}} c_i \text{ and } (\vec{u}-\vec{v})_j < 0.$$

In defining harmony for Harmonic Grammar, we follow Prince (2002a) in treating violations as positive integers and optimization as the selection of violation vectors so as to minimize the *dot product* of the violation vector and the weight vector.[3] This is equivalent to the characterization in work such as Legendre et al. (2006) that represents violations as negative integers and optimization as maximization of the dot product. We make the additional assumption that all of the weights are positive and we denote a 'weighting' drawn from $\mathbb{R}_+^k$ as $\vec{w}$.[4] For a given weighting $\vec{w}$, violation vector $\vec{u}$ is more harmonic than violation vector $\vec{v}$ if the weighted sum of the components of $\vec{u}$ is less than the weighted sum of the components of $\vec{v}$:

(4)
$$\vec{u} \succ_{\vec{w}} \vec{v} \quad \text{iff} \quad \vec{u}\cdot\vec{w} < \vec{v}\cdot\vec{w}$$
or equivalently
$$\vec{u} \succ_{\vec{w}} \vec{v} \quad \text{iff} \quad (\vec{u}-\vec{v})\cdot\vec{w} > 0.$$

A weighting in HG (and respectively a ranking in OT) defines a mapping that selects for each input form the output form(s) that maximize harmony as defined in (3) and (4). Given a candidate-generating function $gen(i)$ that maps each input $i \in \Sigma^*$ to a set of candidates paired with their constraint violations (i.e. $gen : \Sigma^* \to \Sigma^* \times \mathbb{Z}_+^k$), the (i,o)-pairs that appear in the language $L_{\mathcal{R}}$ defined by ranking $\mathcal{R}$ or language $L_{\vec{w}}$ defined by weighting $\vec{w}$ are those whose violation vectors are maximally harmonic.

(5)
$$L = \left\{ (i,o) \mid i \in D, (o,v) \in gen(i), \text{ and } \nexists (o',v') \in gen(i) \text{ such that } o' \succ o \right\}$$

Note that it is possible for multiple candidates for the same input form to 'tie' with the same violation vector and that in HG (but not OT) it is possible for candidates that have different violation vectors to be equi-harmonic (e.g. for any $\vec{v},\vec{w} \in \mathbb{Z}_+^k$ there is an $\vec{x} \in \mathbb{R}_+^k$ such that $\vec{v}\cdot\vec{x} = \vec{w}\cdot\vec{x}$).

Analyses in OT and HG are usually presented with illustrative tableaux that represent the competition among a set of carefully chosen candidates. In Fig. 1 we present four hypothetical candidates along with their violation vectors for three constraints in a standard OT/HG tableau. If the constraints in this tableau are ranked $\mathcal{R} = \langle c_1 \gg c_2 \gg c_3 \rangle$ then candidate *b* is optimal according to $\mathcal{R}$ because it has the fewest violations of the top-ranked constraint c_1 along with candidate *a*, which it beats with fewer violations of constraint c_3. On the other hand, if the constraints are weighted by $\vec{w} = \langle 0.1, 0.3, 0.5 \rangle$ then candidate *d* is optimal according to $\vec{w}$ because the weighted sum of its violations, $\vec{w}\cdot\langle 3,0,1\rangle$, is 0.8 while those of candidates *a*, *b*, and *c* are 1.9, 1.4, and 0.9. Candidate *a* is included in Fig. 1 to illustrate an instance of a candidate that could never be optimal under any ranking or weighting of the constraints. This occurs because *a* has a constraint-wise superset of the violations of candidate *b* and thus will be less harmonic than *b* under *every* ranking/weighting. Prince and Smolensky (1993/2004) call this "harmonic bounding."

Candidate *c* is of particular interest. Note that there is no ranking of the constraints that can make *c* optimal because it is harmonically bounded by candidates *b* and *d* together, though it is not bounded by either one individually. Samek-Lodovici and Prince (1999) call this "collective" harmonic bounding. There is, however, a constraint weighting $\vec{w} = \langle 0.4, 0.5, 0.1 \rangle$ that makes candidate *c* most harmonic, with a weighted sum of 1.0, compared to 1.7, 1.6, and 1.3

[3] The dot product of vectors $(a_1,\ldots,a_n)$ and $(b_1,\ldots,b_n)$ is the sum of their component-wise products: $\vec{a}\cdot\vec{b} = \sum_{i=1}^n a_i b_i$.

[4] We adopt the restriction that weights may not be negative because it provides a sufficient condition to ensure that optima are well defined. We return to the possibility of mixed positive and negative weights in §5.

input	c_1	c_2	c_3
candidate *a*	0	3	2
candidate *b*	0	3	1
candidate *c*	1	1	1
candidate *d*	3	0	1

Figure 1: A tableau with four candidates and three constraints.

for candidates *a*, *b*, and *d*. Candidate *c* exemplifies the kind of input-output pair that that can be generated by HG but not OT. With just two constraints, which allow at most a two-way choice among candidates in an OT tableau, it is possible to construct an HG tableau with arbitrarily many rows that are each optimal under a different weighting (see Legendre et al. (2006) for a concrete example). The fact that HG can generate infinitely many languages with only two constraints and the fact that there are constraint interactions in HG that are impossible in OT suggests, *prima facie*, that learning HG grammars could be more complex than learning OT grammars.

3. The Vapnik Chervonenkis Dimension

The VC dimension comes from the work of Vapnik and Chervonenkis (Vapnik and Chervonenkis, 1971).[5] It is a powerful metric that is often used in computational learning theory to quantify the maximum degree of independence among the data that can be classified by a given set of classification functions. VC dimension is usually described in terms of "concept classes," where a concept class $\mathcal{C}$ is a (possibly infinite) set of "concepts" or "classifiers" that evaluate the elements of a (possibly infinite) set $\mathcal{X}$ called the "instance space." Often $\mathcal{C}$ is taken to contain boolean classifiers, so for each $c \in \mathcal{C}$, and for each $x \in \mathcal{X}$, the concept c maps x to 1 if x is "in" the concept and to 0 if it is "out."

A simple example may illustrate the terminology. Suppose that we wish to characterize any person by two quantities: weight and height. A person p_i is then represented by a pair of postive real numbers (w_i, h_i), where w_i and h_i are her weight and height (in some units). The set of all possible such pairs (i.e. $\mathbb{R}^2_+$) is then our instance space $\mathcal{X}$, the set of all possible weight-height combinations. A boolean concept on this space is any function from these pairs to 0 or 1 (i.e. any "classification function"). For instance, we might define a concept c_{stout} of "stoutness," such that $c_{\text{stout}}((w_i, h_i)) = 1$ if a person with weight w_i and height h_i is heavy and short enough to be called "stout," and 0 otherwise. We could define an analogous concept of "slenderness" that is true of people sufficiently light and tall, or infinitely many other binary concepts for this instance space. One possible definition of c_{stout} might be:

$$c_{\text{stout}}((w_i, h_i)) = \begin{cases} 1 \text{ if } \frac{w_i}{h_i} > 2; \\ 0 \text{ otherwise.} \end{cases} \tag{6}$$

Here any point in the instance space is classified as "stout" if the weight is more than twice the height. The choice of 2 as the threshold is arbitrary, and different values would result in different concepts (corresponding to different definitions of stoutness). A *concept class* on $\mathcal{X}$ is simply any set of concepts on $\mathcal{X}$; for instance the class of all concepts with the form in (6), for different threshold values. We might call this the "weight-to-height threshold" concept class, $\mathcal{C}_{w/h>k}$, and define it as:

$$\mathcal{C}_{w/h>k} = \{c : \mathcal{X} \to \{0,1\} \mid \exists k \text{ such that } \forall (w_i, h_i) \in \mathcal{X},\ c((w_i, h_i)) = \left\{ \begin{array}{l} 1 \text{ if } \frac{w_i}{h_i} > k; \\ 0 \text{ otherwise.} \end{array} \right\}\}. \tag{7}$$

There are infinitely many other possible concept classes on this space, for example the set of concepts defined by some linear combination of w_i and h_i being greater than some threshold (i.e. "linear classification functions"). Any

[5] See Kearns and Vazirani (1994) and Vapnik (1998) for introductions to computational learning theory.

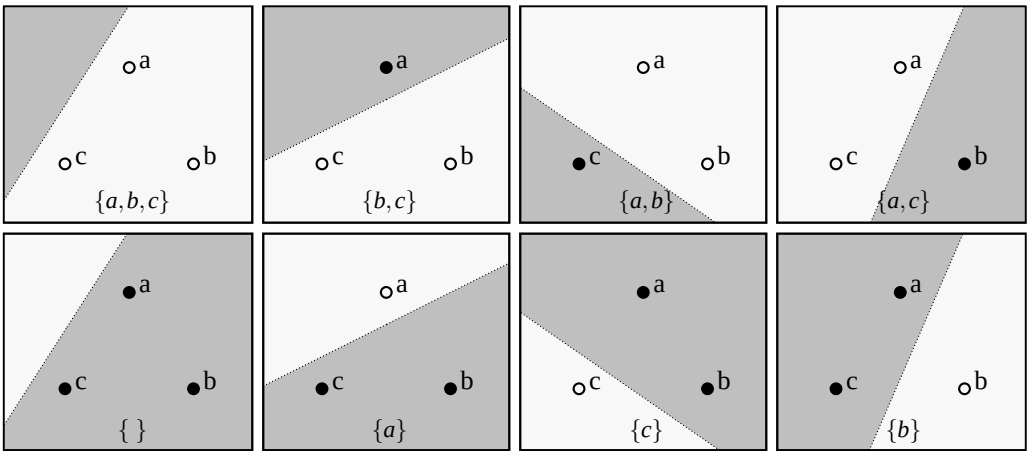

Figure 2: A set of three points that is shatterable by half-spaces in $\mathbb{R}^2$.

set of concepts constitutes a concept class, but usually we are interested in concept classes that are defined by some parametrization—that is, sets of concepts that result from varying the value(s) of some parameter(s) in a defining formula, such as k in (7)—and how a concept class' properties, like the VC dimension, change as its parameters are varied.

The VC dimension of a binary concept class $\mathcal{C}$ is the largest set of elements $S = \{a_1, \ldots, a_n\} \subseteq \mathcal{X}$ that is *shatterable* as in (8):

(8) $\{a_1, \ldots, a_n\}$ is shatterable iff $\forall (v_1, \ldots, v_n) \in \{0,1\}^n, \exists c \in \mathcal{C}$ such that $c(a_i) = v_i$.

Less formally, for any division of the members of S into two discrete groups, there must be a concept in $\mathcal{C}$ that classifies all the members of one group as "in" and all the members of the other group as "out." The *VC dimension* of concept class $\mathcal{C}$ is the size (cardinality) of the largest shatterable $S \subseteq \mathcal{X}$.

A classic example used to illustrate shatterability of boolean concept classes is half-spaces in the real plane. Let $\mathcal{X} = \mathbb{R}^2$ (the x-y plane) and let $\mathcal{C}$ be all half-spaces in the x-y plane. We will show that many (but not all) sets of 3 points in the plane are shatterable, while no set of 4 points is shatterable, so the VC dimension of half-spaces in $\mathbb{R}^2$ is 3.

Consider a set of three points $a, b, c \in \mathbb{R}^2$ that are not collinear (i.e. do not all lie on one line). A concept $c \in \mathcal{C}$ that includes all the points $\{a,b,c\}$ is obtained by drawing a line off to one side of them and selecting the half-space on the side of the line facing the points. For any pair of points $\{b,c\}$, $\{a,b\}$, or $\{a,c\}$, a $c \in \mathcal{C}$ that includes that pair but excludes the third point is obtained by drawing a line between the excluded point and the pair, then selecting the half-space on the side of the line facing the pair. These four cases are illustrated in the top row of Fig. 2. The other four cases—a concept $c \in \mathcal{C}$ that includes none of the points, and concepts that uniquely include $\{a\}$, $\{c\}$, and $\{b\}$—are obtained by inverting the half-spaces in the first four cases; these are illustrated in the bottom row of Fig. 2.

Because there is a shatterable set of three points, the VC dimension is at least 3. This does not mean that all 3-point sets must be shatterable: any set of three collinear points cannot be shattered by half-spaces because one of the points lies between the other two, making it impossible to include the outer points while excluding the middle one.

This situation also illustrates why no set of four points can be shattered. With four points it is either the case that one point lies in the interior of the triangle whose corners are the other three points, or that the four points are the corners of a 4-sided convex polygon. In the former case, no half-space can include the three corners while excluding the interior point; in the latter case, if we label the corners clockwise a, b, c, d then no half-space can include a and c while excluding b and d. (If this is too abstract, we suggest the exercise of trying to replicate Fig. 2 for four points.)

In many cases, the point of analyzing the VC dimension of a concept class is to determine how the VC dimension grows as a function of some parameters of the concept class. Particularly important is the way that the "dimensionality" of the concept class relates to its VC dimension. For half-spaces in $\mathbb{R}^n$, the dimensionality of the concepts is n and the VC dimension of is $n+1$ (the $n = 2$ case was illustrated above; see Kearns and Vazirani (1994) for the general

input 1	c_1	c_2	c_3
candidate *a*	1	4	0
candidate *b*	0	0	4
candidate *c*	0	1	2

input 2	c_1	c_2	c_3
candidate *d*	2	2	0
candidate *e*	3	1	0
candidate *f*	0	5	1

Figure 3: Two tableaux with three candidates and three constraints, shatterable in HG.

candidate *a*	1	4	0
candidate *b*	0	0	4

candidate *d*	2	2	0
candidate *e*	3	1	0

candidate *a*	1	4	0
candidate *c*	0	1	2

candidate *d*	2	2	0
candidate *f*	0	5	1

candidate *b*	0	0	4
candidate *c*	0	1	2

candidate *e*	3	1	0
candidate *f*	0	5	1

Figure 4: Representation of the tableaux in Fig. 3 as two sets of three binary tableaux.

case). The fact that the growth of the VC dimension is a linear function of dimensionality of the concepts is important because it means that half-spaces can be learned from sets of randomly drawn samples whose size is a linear function of their dimensionality (Blumer et al., 1989).

For OT or HG with a given constraint set CON, we take the concept class $\mathcal{C}$ to be the set of all possible grammars over CON (i.e. all rankings or weightings of the constraints). In this case the dimensionality of $\mathcal{C}$ is the number of constraints in CON, and the 'samples' that the learner must classify are sets of tableaux where each tableau consists of a set of output candidates for the same input form, where each output candidate is paired with its violations of the constraints (see Fig. 3).

Thus far, we have been characterizing concepts as boolean classifiers that take points in some multidimensional space and map them to 1 if they are 'in' the concept and 0 if they are 'out.' Though tableaux in HG and OT usually represent competition among more than two candidates they can be straightforwardly reduced to sets of binary decisions. A *binary* tableau is one that contains just two candidates. Any tableau with n candidates can be seen as comprising $(n^2-n)/2$ binary tableaux (one for each pair of candidates). Thus the two tableaux in Fig. 3 can be recast as two sets of three binary tableaux as in Fig. 4.

If we adopt the convention that binary tableaux represent the proposition that the first candidate is more harmonic than the second candidate, then a grammar (i.e. a concept) in OT/HG maps each binary tableau to 1 if the first candidate is more harmonic and to 0 if the second is more harmonic. If the candidates are equi-harmonic, either because they have the same violation vector or because the dot product of their violations with the constraint weighting are equal, then the grammar will return a third value such as 'undecided.' These cases are not relevant to the discussion at hand because tableaux that contain candidates with the same violation profile cannot occur in shatterable samples—no grammar could include one of two tied candidates while excluding the other—and the fact that some weighting might fail to distinguish between two candidates is irrelevant as long as there exists a weighting that does so.

The VC dimension of a concept class comprising all weightings/rankings of k constraints in HG/OT is the cardinality of the largest set of tableaux that can be shattered in the sense that, for every possible way of selecting one winner per tableau, there is a grammar that chooses that pattern of winners. This quantity is relevant for learning because it represents an absolute worst-case bound on any learner's ability to generalize. Given a shatterable set of n tableaux, if we present the tableaux in any order and tell the learner which candidates are optimal in the first $n-1$ of them, the learner will still be left with no recourse but to guess which candidate is optimal in the n-th tableau.

Generally speaking, tableaux with more candidates give the learner more information, so the worst cases will involve tableaux with only two candidates. In fact, by deconstructing tableaux into sets of binary tableaux as in Fig. 4, it is simple to demonstrate that every set of n shatterable tableaux must contain a shatterable set of n binary tableaux.

This means that establishing the VC dimension of sets of binary tableaux will bound the VC dimension for all sets of tableaux.

Theorem 3.1. *Every shatterable set of n tableaux contains a shatterable set of n binary tableaux.*

Proof. Assume, for a contradiction, that T is a shatterable set of n tableaux that contains non-binary tableaux but not a shatterable set of n binary tableaux. Consider the set of tableaux T_2 where each tableau in T_2 is the first two rows of a tableau in T. If T_2 is *not* shatterable then there is some choice of winners that is not supported by a ranking/weighting. This means that for some choice of winners $\mathcal{W}$ it is the case that for some $R \subset T$ and for some tableau $t \in T - R$ the ranking/weighting entailed by $\mathcal{W}$ makes one of the candidates in t more harmonic than the other. If this were so, then there would be a pattern of winners in T that is not supported by any ranking/weighting, contrary to the assumption that T is shatterable. □

The VC dimension for a specific set of constraints, a specific set of input forms, or a specific set of tableaux, will usually be far lower than the bound that we derive for the general case.[6] The advantage of making no assumptions about the constraints (other than that there are k of them) and no assumptions about what kinds of inputs or tableaux are possible, is that we establish a bound on the complexity of the class of HG grammars that is independent of these details. Moreover, because the general worst-case VC dimension is so tame, there is no need to attempt to formulate variants of the weighting/ranking problem to ensure that grammars are learnable.

Riggle (2009) reduces sets of binary tableaux in OT to sets of statements in the three-valued logic of Prince's (2002b) Elementary Ranking Conditions (ERCs) and shows that the cardinality of the largest shatterable set of ERCs for k constraints is $k-1$. In the next section, we take a parallel approach and reduce sets of binary tableaux in HG to sets of linear inequalities over constraint weightings, then show that the largest shatterable set of linear inequalities for k constraints is $k-1$ if all constraint weights are positive and k if weights are positive and negative.

4. The VC dimension of Harmonic Grammar

As noted by Potts et al. (2008), tableaux containing sets of candidates correspond to systems of linear inequalities. Consider a tableau of n candidates in an HG grammar over k constraints, with weights $\vec{w} = \langle w_1, \ldots, w_k \rangle$. Once a winner is specified, there are $n-1$ winner/loser pairs, each of which specifies a linear inequality in the weights. For a set of tableaux T, any choice of winners (one winner for each $t \in T$) yields system of linear inequalities that select each winner.

For example, if the winners in Fig. 5 are candidates a, c, and e, we obtain three linear inequalities (one for each winner/loser pair). Taken together with the condition that weights be positive, this set of tableaux and choice of winners imply six inequalities:

$$\begin{aligned} a \succ_{\vec{w}} b \quad &\Longrightarrow \quad w_1 - w_3 > 0 \\ c \succ_{\vec{w}} d \quad &\Longrightarrow \quad -w_1 + w_2 - w_3 + w_4 > 0 \\ e \succ_{\vec{w}} f \quad &\Longrightarrow \quad 2w_2 - w_4 > 0 \\ \text{positive weights} \quad &\Longrightarrow \quad w_1 > 0,\ w_2 > 0,\ w_3 > 0 \end{aligned}$$

A choice of winners is selected by any vector of weights satisfying the system of inequalities implied by that choice of winners; the set of such vectors is called the *feasible region* of the choice of winners. If the feasible region for a choice of winners is empty, then there is no setting of weights that selects those winners. If every choice of winners in a set of tableaux has a non-empty region of the weighting space that supports it, then that set of tableaux is shatterable.

Given a set of n binary tableaux over k constraints, each tableau defines a *hyperplane* which splits the space $\mathbb{R}^k$ of possible weights into two open *half-spaces*, one corresponding to the set of weightings under which the first candidate

[6] This result applies to tableaux with *any* number of candidates, even abstract tableaux containing an infinite range of candidates (say, one for each language in an infinite HG typology). Tableaux with fewer than two candidates do not require learners to make decisions and are thus irrelevant to the VC dimension, and if a set of tableaux is to be shattered then even if one of its members has infinitely many rows it must still be the case that any pair of rows we extract from the infinite tableau can be included among a shatterable set of binary tableaux.

input 1	c_1	c_2	c_3	c_4	
candidate a	1	1	1	0	Implication: $a \succ b$ iff $w_1 > w_3$
candidate b	0	1	2	0	Implication: $b \succ a$ iff $w_3 > w_1$

input 2	c_1	c_2	c_3	c_4	
candidate c	1	0	2	0	Implication: $c \succ d$ iff $w_2 + w_4 > w_1 + w_3$
candidate d	0	1	1	1	Implication: $d \succ c$ iff $w_1 + w_3 > w_2 + w_4$

input 3	c_1	c_2	c_3	c_4	
candidate e	0	0	2	1	Implication: $e \succ f$ iff $2w_2 > w_4$
candidate f	0	2	2	0	Implication: $f \succ e$ iff $w_4 > 2w_2$

Figure 5: A shatterable set of three tableaux and linear inequalities for possible winners.

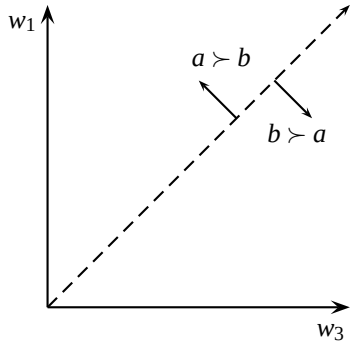

Figure 6: The feasible regions for $a \succ b$ and $b \succ a$.

is more harmonic and the other to the weightings under which the second is more harmonic.[7] This representation, used by Potts et al. (2008), affords a straightforward interpretation where the linear inequalities for the half-spaces correspond to intuitive statements like "a is more harmonic than b if the weight assigned to c_1 is greater than that assigned to c_3." (An equivalent but less intuitive representation can be obtained by casting the half-spaces as points and the constraint weightings as linear classifiers separating them. We discuss this "point representation" in the appendix.) Each hyperplane at the boundary between opposing half-spaces passes through the origin of the weighting space (i.e. the point 0^k) because any pair of candidates that violate no constraints at all must be equally harmonic.

As a concrete example, consider in Fig. 6 the half-spaces corresponding to the first tableau of Fig. 5. When $a \succ_{\vec{w}} b$, the inequality is $w_1 - w_3 < 0$; when $b \succ_{\vec{w}} a$, it is $w_1 - w_3 > 0$. In Fig. 6 we consider only the w_1 and w_3 dimensions because they are the only relevant dimensions in the comparison. For larger sets of tableaux over many constraints we will have regions in high-dimensional space. For n tableaux this is formalized as follows.

An *arrangement* is a set $\mathcal{A} = \{P_1, \ldots, P_n\}$ of n hyperplanes in $\mathbb{R}^k$. Each hyperplane in an arrangement splits $\mathbb{R}^k$ into two open half-spaces. The arrangement partitions $\mathbb{R}^k$ into a set $\mathcal{S}(\mathcal{A})$ of 2^n *sectors*, each of which is an intersection of n open half-spaces. Put otherwise, each sector corresponds to choosing a vector

$$\vec{o} = \langle o_1, \ldots, o_n \rangle, \qquad o_i \in \{-1, 1\}$$

designating which side of each of the n hyperplanes the sector lies on. Each $\vec{o} \in \{-1,1\}^n$ is called an *orientation* of the n hyperplanes. If all members of $\mathcal{A}$ intersect at the origin, the arrangement is *homogeneous*. Because each

[7] The weightings on the boundary between the two half-spaces are those where the candidates are equi-harmonic.

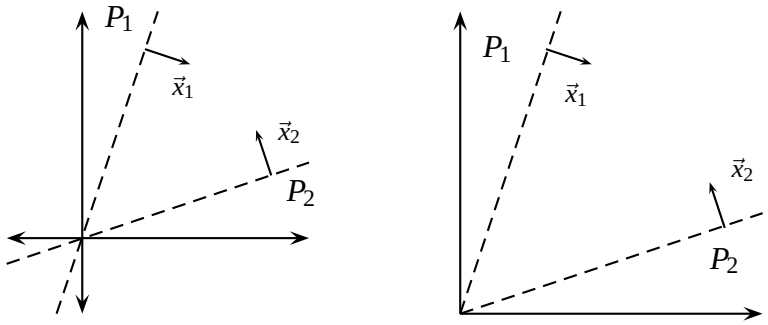

Figure 7: The arrangement $\{P_1, P_2\}$ is shatterable (left), but not shatterable in the positive orthant (right). $\vec{x}_1, \vec{x}_2$ are the normal vectors of P_1, P_2.

binary tableau maps to a hyperplane including the origin, we are interested only in homogeneous arrangements. For a homogeneous arrangement $\mathcal{A}$, the hyperplanes P_i are completely specified by their normal vectors $\vec{x}_i$. Specifying on which side of P_i a point $\vec{p} \in \mathbb{R}^k$ lies corresponds to specifying the sign of $\vec{p} \cdot \vec{x}_i$ ($\in \{-1, 1\}$).[8] We can therefore say that a point $\vec{p}$ lies in the sector of arrangement $\mathcal{A}$ specified by orientation $\vec{o}$ if and only if $\text{sign}(\vec{p} \cdot \vec{x}_i) = o_i$ for each $i = 1, \ldots, n$. This identification is used in the proofs below.

We can now define shatterability for a homogeneous arrangement of n hyperplanes in $\mathbb{R}^k$, which is equivalent to shatterability for a set of n binary tableaux in an HG grammar with k constraints.

Definition 1. A homogeneous arrangement $\mathcal{A}$ of n hyperplanes in k dimensions is *shatterable* if all its sectors ($S \in \mathcal{S}(\mathcal{A})$) are non-empty.

In HG terms, each sector $S \in \mathcal{S}(\mathcal{A})$ corresponds to a vector $\vec{W} \in \{0, 1\}^n$ specifying a winner for each of the n binary tableaux. Non-empty S means there exist weights $\vec{w} \in \mathbb{R}^k$ under which the winners are $\vec{W}$. Our assumption that constraint weights are non-negative adds one final condition to the definition of shatterability in HG.

Definition 2. A arrangement $\mathcal{A}$ of n hyperplanes in k dimensions is *shatterable in the positive orthant* $\mathbb{R}^k_+ = \{(x_1, \ldots, x_k) \,|\, x_i > 0, i = 1, \ldots, k\}$ if, for each sector $S \in \mathcal{S}(\mathcal{A})$, $S \cap \mathbb{R}^k_+ \neq \emptyset$.

Fig. 7 illustrates that for $k = 2$, a homogeneous arrangement of 2 non-identical hyperplanes ($n = 2$) will be shatterable, but not shatterable in the positive orthant. In this example, any arrangement $\{P_1, P_2\}$ of non-identical, homogeneous planes P_1 and P_2 define four non-empty sectors in the $x-y$ plane, but at most three of these intersect the positive orthant.[9] From here on we use "shatterable" to mean "shatterable in the positive orthant."

Theorem 4.1. *The VC dimension of homogeneous arrangements of hyperplanes in* $\mathbb{R}^k_+$ *is* $n - 1$.

Proof of lower bound. We show that there is a homogeneous arrangement of $k-1$ hyperplanes in $\mathbb{R}^k_+$ which is shatterable. Consider the set of $n-1$ hyperplanes $P_1 : w_1 = w_2, \ldots, P_{n-1} : w_{n-1} = w_n$. Let $\vec{o} = (o_1, \ldots, o_{n-1})$ be an orientation of these hyperplanes defining a sector. Construct a weight vector $\vec{w} = (w_1, \ldots, w_k)$ as follows:

1. Choose $w_1 > 0$.
2. For $i = 2, \ldots, k$: If $o_{i-1} = 1$, choose w_i such that $w_{i-1} > w_i$ and $w_i > 0$. If $o_{i-1} = -1$, choose w_i such that $w_{i-1} < w_i$ and $w_i > 0$.

For any $\vec{o}$, $\vec{w}$ lies in the positive orthant and in the sector defined by $\vec{o}$. □

[8] This is because $\frac{\vec{p} \cdot \vec{x}}{|\vec{p}||\vec{x}|} = \cos\theta$, where θ is the angle between $\vec{p}$ and $\vec{x}$, $\theta \in (0°, 180°)$. If $\theta \in (0°, 90°)$, then $\vec{p} \cdot \vec{x} > 0$, while if $\theta \in (90°, 180°)$, $\vec{p} \cdot \vec{x} < 0$

[9] The case where both P_1 and P_2 cross the positive orthant is shown. If only one of P_1 and P_2 crosses the positive orthant, two sectors intersect it; if neither plane crosses the positive orthant, only one sector intersects it.

input 1	c_1	c_2	c_3	c_4	
candidate a	0	1	0	0	Implication: $a \succ b$ iff $w_1 > w_2$
candidate b	1	0	0	0	Implication: $b \succ a$ iff $w_2 > w_1$

input 2	c_1	c_2	c_3	c_4	
candidate c	0	0	1	0	Implication: $c \succ d$ iff $w_2 > w_3$
candidate d	0	1	0	0	Implication: $d \succ c$ iff $w_3 > w_2$

input 3	c_1	c_2	c_3	c_4	
candidate e	0	0	0	1	Implication: $e \succ f$ iff $w_3 > w_4$
candidate f	0	0	1	0	Implication: $f \succ e$ iff $w_4 > w_3$

Figure 8: A shatterable set of three tableaux.

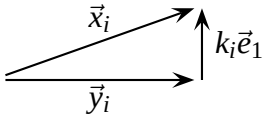

Figure 9: $\vec{y}_i$ is the component of $\vec{x}_i$ perpendicular to $\vec{e}_1$

The idea is that for k constraints we can construct a set of tableaux $\{t_1,...,t_{k-1}\}$ in which each t_i has equal but opposite non-zero violations for pairs of constraints (c_i, c_{i+1}). For such a set of tableaux, all that is necessary for shatterability is that the relative weight of constraints c_i and c_{i+1} be set to favor the desired winner in binary tableaux i and $i+1$. A shatterable set of three tableaux for $k = 4$ that uses this construction is given in Fig. 8.

Proof of upper bound. We show that a homogeneous arrangement of $n = k$ hyperplanes in k dimensions is not shatterable in the positive orthant.

The $n = 1$ case is trivial. Assume $n \geq 2$ and that there exists such an arrangement $\mathcal{A} = \{P_1, \ldots, P_k\}$, and let $\vec{x}_i$ be the normal vector of hyperplane P_i. Let $\vec{e}_1$ be the unit vector in the direction of the first coordinate axis, and let $\vec{y}_i$ be the component of $\vec{x}_i$ perpendicular to $\vec{e}_1$ (Fig. 9), i.e.

$$\vec{y}_i = \vec{x}_i - k_i\vec{e}_1 \quad (i = 1, \ldots, k) \tag{9}$$

where k_i is $\vec{x}_i \cdot \vec{e}_1$, the first coordinate of $\vec{x}_i$. Since $\vec{y}_1, \ldots, \vec{y}_k$ lie in a $k-1$ dimensional subspace, they span a subspace of dimension at most $k-1$, so one of the $\vec{y}_i$ is linearly dependent on the others. Without loss of generality, assume $\vec{y}_1$ is linearly dependent on $\vec{y}_2, \ldots, \vec{y}_k$, so that

$$\vec{y}_1 = c_2\vec{y}_2 + \cdots + c_k\vec{y}_k$$

for some $c_2, \ldots, c_k \in \mathbb{R}$. Substituting from (9),

$$\begin{aligned} \vec{x}_1 - k_1\vec{e}_1 &= \sum_{i=2}^{k} c_i(\vec{x}_i - k_i\vec{e}_1) \\ \implies \vec{x}_1 &= K\vec{e}_1 + \sum_{i=2}^{k} c_i\vec{x}_i \end{aligned}$$

for $K = k_1 - \sum_{i=2}^{k} c_i k_i$. For any $\vec{z} \in \mathbb{R}^k$,

$$\vec{z} \cdot \vec{x}_1 = K(\vec{z} \cdot \vec{e}_1) + \sum_{i=2}^{k} c_i(\vec{z} \cdot \vec{x}_i) \tag{10}$$

input 1	c_1	c_2	c_3	
candidate a	0	1	0	Implication: $a \succ b$ iff $w_1 > w_2$
candidate b	1	0	0	Implication: $b \succ a$ iff $w_2 > w_1$

input 2	c_1	c_2	c_3	
candidate c	0	0	1	Implication: $c \succ d$ iff $w_2 > w_3$
candidate d	0	1	0	Implication: $d \succ c$ iff $w_3 > w_2$

input 3	c_1	c_2	c_3	
candidate e	0	0	1	Implication: $e \succ f$ iff $w_1 > w_3$
candidate f	1	0	0	Implication: $f \succ e$ iff $w_3 > w_1$

Figure 10: A set of three tableaux that cannot be shattered.

Now assume $K \geq 0$ and consider the sector S defined by the orientation $(-1, \text{sign}(c_2), \ldots, \text{sign}(c_k))$. By shatterability in the positive orthant, S has non-empty intersection with the positive orthant $\mathbb{R}^k_+$, so choose $\vec{y} \in S \cap \mathbb{R}^k_+$. As above, $c_i(\vec{y} \cdot \vec{x}_i) \geq 0$ ($i = 2, \ldots, k$) and $\vec{y} \cdot \vec{x}_1 < 0$, while $\vec{z}$ has a positive first coordinate $\implies \vec{z} \cdot \vec{e}_1 > 0$, so from (10),

$$0 > \vec{z} \cdot \vec{x}_1 = K(\vec{z} \cdot \vec{e}_1) + \sum_{i=2}^{k} c_i(\vec{z} \cdot \vec{x}_i) \geq 0 \tag{11}$$

a contradiction.

The $K < 0$ case is analogous.[10] □

We illustrate the argument used in this proof by example, using the set of three binary tableaux ($k = 3$, $n = 3$) in Fig. 10. Subtracting the second from the first row of each tableau gives the normal vectors defining 3 planes in $\mathbb{R}^3$,

$$\vec{x}_1 = (-1, 1, 0), \quad \vec{x}_2 = (0, -1, 1), \quad \vec{x}_3 = (-1, 0, 1)$$

Using (9) then gives vectors $\vec{y}_i$ perpendicular to $\vec{e}_1 = (1, 0, 0)$:

$$\vec{y}_1 = (0, 1, 0), \quad \vec{y}_2 = (0, -1, 1), \quad \vec{y}_3 = (0, 0, 1)$$

and $k_1 = -1$, $k_2 = 0$, $k_3 = -1$. $\vec{y}_1$ can be written as a linear combination of $\vec{y}_2$ and $\vec{y}_3$ ($\vec{y}_1 = -\vec{y}_2 + \vec{y}_3$), giving $c_2 = -1$, $c_3 = 1$ and $K = k_1 - c_2 k_2 - c_3 k_3 = -1 + 0 + 1 = 0$.

Eqns. 10–11 show how the fact that $\vec{y}_1$ can be written in terms of $\vec{y}_2, \ldots, \vec{y}_n$ implies that the n tableaux are not shatterable, by showing that no $w_1, \ldots, w_k$ satisfy one of the 2^n orientations. For our example, consider the orientation $(-1, \text{sign}(c_2), \text{sign}(c_3)) = (-1, -1, 1)$, meaning $a \succ b$, $c \succ d$, $f \succ e$. Then $w_1 > w_2, w_2 > w_3 \implies w_1 > w_3$ (from tableaux 1 and 2), but $w_3 > w_1$ (from tableau 3), a contradiction.

[10] The upper bound proof is similar to Vapnik's (1998) proof of the VC dimension of indicator variables of linear combinations of functions. This is because the (coefficients of) hyperplanes corresponding to binary tableaux in HG can equivalently be considered as points, and weight vectors as hyperplanes separating them (see appendix), which form a linear family of functions. The HG case is slightly different because not all points (binary tableaux) are in the domain, only those not in the positive or negative orthants, and not all weight vectors are considered, only those that define hyperplanes which pass through the positive orthant. We characterize tableaux as orientations of hyperplanes here because it corresponds more transparently to the consequences of different optima for the constraint weightings (i.e. in Fig. 10 candidate a is more harmonic than b just in case w_1 is greater than w_2).

5. Discussion

Our result establishes an upper bound on the VC dimension that is based only on the dimensionality (i.e. number of constraints) of an HG grammar. Specific sets of constraints and/or specific sets of tableaux can have much lower VC dimension. Though results for specific cases will surely be of interest, our results show that in the general case the complexity of the weighting/ranking problem is already quite manageable. It is reasonable to ask whether lifting the restriction to positive weights changes the VC dimension of the problem. As it turns out, the VC dimension of arrangements of homogeneous half-spaces in $\mathbb{R}^k$ is k, just one greater than the VC dimension with all positive weights.[11] If the concept space is recast in the point representation described in the appendix, this question is addressed by a relatively well-known result in machine learning (Vapnik, 1998, 156), that the VC dimension of homogeneous linear classifiers in k dimensions is k. This connection illustrates an important aspect of the kind of analysis presented here. By formalizing linguistically-familiar problems in ways that show them to be familiar problems in other disciplines, we make it easier to take advantage of research that has already been done in those disciplines.

The VC dimension of a concept class has broad ramifications for the learnability of that class. First, note that all finite concept classes have finite VC dimension because it takes 2^n concepts to shatter a sample of size n.[12] Infinite concept classes can have infinite VC dimension, but not all do. For example, the main result of this paper is that the infinite class of languages generated by positive real-valued weightings of a set of k constraints under HG is $k-1$. On the other hand, the VC dimension of the infinite class of languages generated by HG/OT grammars made up of arbitrary sets of constraints is clearly infinite (for n binary tableaux, a set of 2^n constraints can select each patten of winners). For concept classes with infinite VC dimension, there is no upper bound on the number of samples that might be required for a learner to converge on the correct hypothesis. On the other hand, for concept classes with finite VC dimension d, a sequence of training samples whose size is essentially linear in d is sufficient to learn any concept in the class in the framework of *PAC-learnability* (Probably Approximately Correct; Valiant, 1984)

Blumer et al. (1989) show that in the PAC model, for arbitrarily small δ and ε, with probability $1-\delta$, a hypothesis whose probability of misclassification is less than ε will be found by any learning algorithm that is *consistent* in the sense that it always correctly labels data from previously seen samples. Moreover, the number of samples m that are required to guarantee such a hypothesis for any concept from a class with VC dimension d is:

$$m = \max\left(\frac{4}{\varepsilon}\log_2\frac{2}{\delta},\ \frac{8d}{\varepsilon}\log_2\frac{13}{\varepsilon}\right) \implies m \leq \frac{4}{\varepsilon}\left(2d\log_2\frac{13}{\varepsilon}+\log_2\frac{2}{\delta}\right). \tag{12}$$

In other words, even in the worst case, with the most adversarial probability distribution on the sample space, and with the worst consistent learning algorithm, a sample of size linear in the VC dimension of the class guarantees (PAC) learnability. Thus with k constraints, any consistent HG learning algorithm needs to observe the outcomes of only linearly many (in k) binary tableaux to have probability $1-\delta$ of finding a hypothesis grammar (vector of constraint weights) that misclassifies the language the sample represents with less than ε probability.

In addition to this worst-case upper bound on sample size, Blumer et al. (1989) show that the VC dimension d of a concept class establishes a minimal sample size, below which no learning algorithm can generate, with confidence δ, a hypothesis better than chance ($\varepsilon < \frac{1}{2}$):

$$m \geq \max\left(\frac{1-\varepsilon}{\varepsilon}\ln\frac{1}{\delta},\ d(1-2(\varepsilon(1-\delta)+\delta))\right). \tag{13}$$

That the VC dimension of HG is finite is thus a positive result for the learnability of HG in general, and that it is linear in the dimensionality of the model (i.e. the number of constraints) is a positive result for the worst case behavior of any HG learning algorithm. Unlike the VC dimension of OT, which was guaranteed to be finite simply by the fact

[11] The lower and upper bound proofs are simpler without the restriction to the positive orthant. For the lower bound, let $\mathcal{A}$ be all planes perpendicular to the coordinate axes; the sectors are the 2^k orthants. For the upper bound, assume there is a homogeneous arrangement $\mathcal{A} = \{P_1,\ldots,P_{k+1}\}$ of $k+1$ hyperplanes in k dimensions, with $\vec{x}_i$ the normal vector of hyperplane P_i. One of the $\vec{x}_i$ must be linearly dependent on the others; without loss of generality say $\vec{x}_1 = c_2\vec{x}_2 + \cdots + c_{k+1}\vec{x}_{k+1}$. Then considering the orientation $(-1,\text{sign}(c_2),\ldots,\text{sign}(c_{k+1}))$ leads to a contradiction as in Proof 4.

[12] For example, the VC dimension of the class of languages generated by rankings of a set of k constraints in OT could be at most $\log_2 k!$, which is on the order of $k\ log_2\ k$. Yet, as Riggle (2009) shows, the VC dimension is actually $k-1$.

that there are only $k!$ grammars for k constraints, the VC dimension of HG had no *a priori* guarantee of finiteness. Even though the infinite range of grammars obtained by considering all vectors of k constraint weights yields an infinite typology of languages (depending on the constraints; see Smolensky and Legendre, 2006; Pater et al., 2007), the VC dimensions of the infinite (HG) and finite (OT) models are exactly the same. This means that although HG and OT models may differ greatly in the typological predictions they make, they are the same in terms of the worst-case behavior of learning algorithms for them. Further, the worst case is linear, and thus not at all bad. At least from the perspective of worst-case sample complexity, then, OT and HG are equally reasonable models of human grammar.

Yet OT and HG have significant differences, both in grammars generated and typological predictions. If not in worst-case learnability, where should we look for these differences to manifest themselves? One possibility is to examine OT and HG not just in the worst case but also in the average case, for example using the related notion of *VC entropy* (see Vapnik, 1998, ch. 4; Bousquet et al., 2004). Another is to examine the typologies generated by OT and HG in concrete cases to understand what grammars are generated by HG but not OT, and how they affect learnability (see ongoing work by Pater et al., 2007; Pater, 2009; Bane and Riggle, to appear).

A further point of divergence between OT and HG is in the worst-case mistake bounds for online learning (see Blum (1998) for a survey of on-line learning). For OT, ranking algorithms such as Tesar and Smolensky's (2000) Recursive Constraint Demotion algorithm or Riggle's (2008) r-volume learner are guaranteed to make no more than on the order of k^2 mistakes and $k \log k$ mistakes respectively when learning rankings of k constraints. For HG on the other hand, there is no upper bound on the number of mistakes that a learner can make because the mistake bound is a function of particular properties of the set of training data that are totally independent of the number of constraints. This fact about HG follows directly from the fact that learning weightings is a linear classification problem on separable data (see Vapnik, 1998, 377).

The robustness and the simplicity of the connection between the VC dimension and learnability have led some researchers such as Niyogi (2006, 941) to speculate that finite VC dimension may be a necessary property for the class of human grammars. However, most learnability problems in linguistics—such as learning Minimalist grammars (Chomsky, 1995), Tree Adjoining grammars (Joshi et al., 1975), or Head-driven Phrase Structure grammars (Pollard and Sag, 1994)—have infinite VC dimension. In fact, any formalism that is at least as powerful as regular grammars also has infinite VC dimension (Nowak et al., 2002). By this same token, the problem of learning OT/HG grammars over arbitrary sets of regular constraints has infinite VC dimension. These facts lead Stabler (2009) to suggest that one of the central problems in learnability and natural language is finding the right characterization of the way that the "actual" problem of grammar learning differs from the most general characterizations of the problem. This is precisely what we have done here. Though the problem of learning constraint-based grammars over arbitrary constraints has infinite VC dimension, if the constraints are a known and finite set and if the training data consist of violation vectors for optimal and suboptimal candidates, then the problem of learning weightings/rankings is not hard: even the worst case requires a set of training samples whose size is only linear in the number of constraints. This positive result provides a baseline against which it will be possible to evaluate learning from noisy data, data with (partially) hidden structures, or learning where some of the constraints are unknown.

Appendix: Point representation

Consider a set of n binary tableaux over k constraints. Each tableau can be represented by a hyperplane as in §4, however, we can give an alternative "point representation" where each binary tableau is represented as the k-dimensional normal vector of the hyperplane, which is the difference of the violation vectors corresponding to the two candidates, or *difference vector*.

input	c_1	c_2	c_3	c_4	
candidate a	1	1	2	0	Implication: $a \succ b$ iff $(-1,0,-1,1) \in c$
candidate b	0	1	1	1	Implication: $b \succ a$ iff $(1,0,1,-1) \in c$

Figure 11: Point representation using difference vectors.

The concept class $\mathcal{C}$ defined by HG grammars over k constraints in this representation is the set of homogeneous half-spaces corresponding to weight vectors $\vec{w} \in \mathbb{R}^k_+$. A candidate is selected by a $c \in C$ just in case the difference vectors obtained by subtracting its violations from its competitors violations lie in the half-space c. In this representation, the proof that the VC dimension is $k-1$ can be greatly simplified.

Lower bound. Consider the set of $k-1$ points $\vec{a}_1 = (1,-1,0,\ldots,0)$, $\vec{a}_2 = (0,1,-1,0,\ldots,0)$, …, $\vec{a}_{k-1} = (0,\ldots,1,-1)$. By a similar argument as in Thm. 4.1, this set is shatterable by $\mathcal{C}$.

Upper bound. Suppose $S = \{\vec{a}_1,\ldots,\vec{a}_k\}$ is a set of k points shatterable by $\mathcal{C}$. Shatterability implies these points are linearly independent (Burges, 1998, 160); they thus form a basis for $\mathbb{R}^k$. In particular, the point $\vec{1} = (1,\ldots,1) \in \mathbb{R}^k$ can be written as a linear combination of the $\vec{a}_i$:

$$\vec{1} = \sum_{i=1}^{k} c_i \vec{a}_i \quad \text{(for some } c_i \in \mathbb{R}\text{)}$$

By the assumption that S is shatterable, there exists a weight vector $\vec{w}^* \in \mathbb{R}^k_+$ such that for each $\vec{a}_i \in S$, $\vec{w}^* \cdot \vec{a}_i > 0$ if $c_i < 0$ and $\vec{w}^* \cdot \vec{a}_i < 0$ if $c_i > 0$. Then

$$\vec{w}^* \cdot \vec{1} = \sum_{i=1}^{k} c_i \vec{w}^* \cdot \vec{a}_i < 0.$$

This contradicts the assumption that $\vec{w}^* \in \mathbb{R}^k_+$ because one of $\vec{w}^*$'s components must be negative and thus no set of k points is shatterable by homogeneous hyperplanes with normal vectors in $\mathbb{R}^k_+$. □

In this representation, shatterability is closely tied to linear separability. A shatterable set of binary tableaux corresponds to a set of difference vectors in $\mathbb{R}^k$, each with at least one positive and one negative component (either candidate can win under some weighting), such that for each partition of the set into two parts (a choice of winner for each tableau), the two parts are linearly separable by a hyperplane with normal vector in $\mathbb{R}^k_+$.

Acknowledgments

For useful discussion of this work and insightful comments on earlier drafts, we are grateful to Giorgio Magri, Partha Niyogi, Joe Pater, Chris Potts, and two anonymous *Lingua* reviewers.

References

Albright, A., 2007. Gradient phonological acceptability as a grammatical effect, Ms, MIT.

Albright, A., 2008. From clusters to words: grammatical models of nonce word acceptability. Handout from 82nd Annual Meeting of the Linguistic Society of America, Chicago (January).

Albright, A., Magri, G., Michaels, J., 2008. Modeling doubly marked lags with a split additive model. In: Chan, H., Jacob, H., Kapia, E. (Eds.), Proceedings of the 32nd annual Boston University Conference on Language Development.

Bane, M., Riggle, J., to appear. Evaluating strict domination: The typological consequences of weighted constraints. In: Proceedings of the 45th Annual Meeting of the Chicago Linguistic Society.

Blum, A., 1998. On-line algorithms in machine learning. In: Fiat, A., Woeginger, G. (Eds.), Online algorithms: The state of the art. Springer, Berlin.

Blumer, A., Ehrenfeucht, A., Haussler, D., Warmuth, M. K., 1989. Learnability and the Vapnik-Chervonenkis dimension. J. ACM 36 (4), 929–965.

Boersma, P., Pater, J., 2008. Convergence properties of a gradual learning algorithm for harmonic grammar. Ms, University of Amsterdam and University of Massachusetts at Amherst. ROA-970.

Bousquet, O., Boucheron, S., Lugosi, G., 2004. Introduction to Statistical Learning Theory. Lecture Notes in Computer Science 31, 169–207.

Burges, C., 1998. A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery 2 (2), 121–167.

Chomsky, N., 1995. The Minimalist Program. MIT Press, Cambridge, MA.

Farris-Trimble, A., 2008a. Cumulative faithfulness effects in phonology. Ph.D. thesis, Indiana University, Bloomington.

Farris-Trimble, A., 2008b. Cumulative faithfulness effects: Opaque or transparent? In: Farris-Trimble, A., Dinnsen, D. (Eds.), IUWPL6: Phonological Opacity Effects in Optimality Theory. IULC Publications, Bloomington, Indiana, pp. 119–45.

Farris-Trimble, A., in press. Nothing is better than being unfaithful in multiple ways. In: Proceedings of the 45th Annual Meeting of the Chicago Linguistic Society. Chicago Linguistic Society, Chicago.

Goldsmith, J., 1990. Autosegmental and Metrical Phonology. Blackwell, Oxford.

Goldsmith, J., 1991. Phonology as an intelligent system. In: Napoli, D. J., Kegl, J. (Eds.), Bridges Between Psychology and Linguistics: A Swarthmore Festschift for Lila Gleitman. Lawrence Erlbaum, Mahwah NJ, pp. 247–267.
Goldsmith, J., 1993a. Harmonic phonology. In: Goldsmith (1993b), pp. 221–269.
Goldsmith, J., 1993b. The Last Phonological Rule: Reflections on Constraints and Derivations. University of Chicago Press, Chicago.
Goldwater, S., Johnson, M., 2003. Learning OT rankings using a maximum entropy model. In: Proceedings of the Workshop on Variation within Optimality Theory. Stockholm University.
Hayes, B., Wilson, C., 2008. A maximum entropy model of phonotactics and phonotactic learning. Linguistic Inquiry 39, 379–440.
Itô, J., Mester, A., 2003. Japanese Morphophonemics: Markedness and Word Structure. MIT Press.
Jäger, G., Rosenbach, A., 2006. The winner takes it all—almost: Cumulativity in grammatical variation. Linguistics 44, 937–71.
Jesney, K., Tessier, A.-M., 2008. Gradual learning and faithfulness: Consequences of ranked vs. weighted constraints. In: Abdurrahman, M., Schardl, A., Walkow, M. (Eds.), Proceedings of the 38th Meeting of the North East Linguistics Society (NELS 38). GLSA, Amherst, MA.
Joshi, A., Levy, L., Takahashi, M., 1975. Tree adjunct grammars. Journal of Computer Systems Science 10 (1).
Kager, R., Shatzman, K., 2007. Phonological constraints in speech processing, paper presented at the Workshop on Experimental Approaches to Optimality Theory, University of Michigan, May 18–20.
Kearns, M., Vazirani, U., 1994. An Introduction to Computational Learning Theory. MIT Press, Cambridge MA.
Legendre, G., Miyata, Y., Smolensky, P., 1990. Harmonic grammar – a formal multi-level connectionist theory of linguistic well-formedness: Theoretical foundations. Proceedings of the Twelfth Annual Conference of the Cognitive Science Society, 388–395.
Legendre, G., Sorace, A., Smolensky, P., 2006. The Optimality Theory–Harmonic Grammar Connection. In: Smolensky and Legendre (2006), pp. 339–402.
Niyogi, P., April 2006. The Computational Nature of Language Learning and Evolution. MIT Press, Cambridge, MA.
Nowak, M. A., Komarova, N. L., Niyogi, P., June 2002. Computational and evolutionary aspects of language. Nature 417, 611–617.
Pater, J., 2009. Weighted constraints in generative linguistics. Cognitive Science to appear, ROA-982.
Pater, J., Bhatt, R., Potts, C., 2007. Linguistic optimization. Ms, University of Massachusetts at Amherst. ROA-924.
Pollard, C., Sag, I. A., 1994. Head-Driven Phrase Structure Grammar. University of Chicago Press, Chicago, IL.
Potts, C., Pater, J., Bhatt, R., Becker, M., 2008. Harmonic grammar with linear programming: From linear systems to linguistic typology. Ms, University of Massachusetts at Amherst, ROA-984.
Prince, A., 2002a. Anything goes. In: Honma, T., Okazaki, M., Tabata, T., Tanaka, S. (Eds.), A New Century of Phonology and Phonological Theory. Kaitakusha, Tokyo, pp. 66–90.
Prince, A., 2002b. Entailed ranking arguments. Ms, Rutgers University, ROA-500.
Prince, A., 2007. Let the decimal system do it for you: A very simple utility function for OT. ROA 934 1207.
Prince, A., Smolensky, P., 1993/2004. Optimality theory: Constraint interaction in generative grammar. ROA-537.
Riggle, J., 2008. Counting rankings, Ms, University of Chicago. Draft available at `http://hum.uchicago.edu/~jriggle/`.
Riggle, J., 2009. The complexity of ranking hypotheses in optimality theory. Computational Linguistics 35, 47–59.
Samek-Lodovici, V., Prince, A., 1999. Optima, Ms, Rutgers University. ROA-363.
Smolensky, P., 1995. On the structure of the Constraint Component CON of UG. Handout to talk presented at University of California at Los Angeles, April 7, 1995. ROA-86.
Smolensky, P., Legendre, G., 2006. The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar. MIT Press.
Stabler, E., 2009. Computational models of language universals: Expressiveness, learnability and consequences. In: Christiansen, M., Collins, C., Edelman, S. (Eds.), Language Universals. Oxford University Press, pp. 200–223.
Tesar, B., Smolensky, P., 2000. Learnability in Optimality Theory. MIT Press.
Valiant, L. G., 1984. A theory of the learnable. Communications of the ACM, 1134–1142.
Vapnik, V., 1998. Statistical Learning Theory. Wiley.
Vapnik, V. N., Chervonenkis, A., 1971. On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability and its Applications 16, 264–280.
Zhang, J., 2007. Constraint weighting and constraint domination: a formal comparison. Phonology 24 (03), 433–459.

ROA = Rutgers Optimality Archive (`http://roa.rutgers.edu`)