The Blackwell Companion to Phonology

Edited by Marc van Oostendorp, Colin J. Ewen, Elizabeth Hume, and Keren Rice

Volume I
General Issues and Segmental Phonology

WILEY-BLACKWELL
A John Wiley & Sons, Ltd., Publication

3 Learnability

JEFFREY HEINZ
JASON RIGGLE

1 Introduction

How can anything (such as a child or any other computing device) automatically acquire (any aspect of) a phonological grammar on the basis of its experience?

This is the fundamental (and unresolved) question of phonological learnability, and it is essentially independent of whether the grammar is taken to be the best description of the cognitive state of the language faculty, or whether the grammar is taken to be the best description of a language pattern, independent of its cognitive status. The problem of learning a general pattern from a finite set of observations has roots in the philosophical problem of inductive logic (Popper 1959; Sober 2008). Though language learning is simply a specific instance of this problem, it has played a perennially central role in the discussion. In fact, the modern formal study of learnability was inspired by the problem of language acquisition and much of the learnability literature is couched in formal language theory, whose early period was also influenced by the founding of generative linguistics. There have been many developments in formal learning theory and related disciplines, such as grammatical inference, computational learning theory, and machine learning. For the purposes of this chapter we will refer to all of these areas with the term *learning theory*.

All characterizations of learning – whether the domain is syntax, phonology, or gardening, and whether the models are connectionist, Bayesian, or symbolic – are subject to the results of learning theory. Even if they are not intended as such, answers to the question posed at the outset of this chapter constitute hypotheses about the broad characteristics of the computations that humans perform in learning the phonology of their language(s). Our goal in this chapter is to motivate the applicability of learning theory to the problem of learning phonological grammars. In this pursuit, we discuss but a fraction of the many grammatical formalisms and models of phonological learning that have been proposed (space does not permit a comprehensive survey, so we apologize in advance to those whose work is omitted in our brief discussion of the literature). Our main points are that learning theory: (i) makes clear *what* it is that is being learned; (ii) reveals little conceptual difference in the problems of learning gradient *vs.* categorical distinctions; (iii) makes a theory of universal grammar inevitable; (iv) can make clear which properties of phonological patterns are important for learnability; (v) emphasizes understanding the *general behavior* of learning models.

§2 reviews the foundations of formal language theory and its relevance to phonology. §3 covers the main contributions of a few of the theoretical learning frameworks to our understanding of the problem of learning phonological grammars. §4 examines the role of structure in generalization. §5 reviews several phonological learning models from the perspective of learning theory.

2 Formal language theory and phonology

How do we represent the patterns (i.e. languages) that a learner might attempt to learn? (See also CHAPTER 101: THE INTERPRETATION OF PHONOLOGICAL PATTERNS IN FIRST LANGUAGE ACQUISITION for a different perspective.) In formal language theory, languages are characterized as sets, relations, or, equivalently, functions (Harrison 1978; Hopcroft *et al.* 1979). This abstraction focuses attention on the patterns themselves rather than the particular grammars that describe the patterns. Though grammars typically reflect the generalizations that we are interested in, there can be many different ways to describe the same language.

2.1 *Phonotactic patterns*

For a concrete example, consider the set of all and only the words that obey a given phonotactic pattern. In this case, the phonotactic pattern makes a binary distinction between well-formed and ill-formed words (gradient distinctions are discussed later). For example, suppose that (1) designates all and only those words which obey the constraint that obstruents in codas do not disagree in voice.

(1) {fɪst, dæft; rabd, . . .}

Already the connection between the foundations of generative grammar and formal language theory are apparent. If the "three dots" in (1) are meant to include only actual English words, then clearly the set is finite. Generative phonologists reject such finite "list" representations, because the evidence is overwhelming that phonological competence goes beyond the finitely many words a speaker actually knows. In other words, the "three dots" are meant to include every conceivable word which includes many things that are not words of English, such as those in (2).

(2) {plɪst, θæft, wabd, . . . , peifɪst, . . .}

The fact that these sets can be infinite is what necessitates a generative grammar – that is, some finite device capable of generating all and only those *logically possible* words which obey the phonotactic pattern.

2.2 Alternations

(Morpho)phonological alternations can be represented as *relations* (i.e. sets consisting of all and only the pairs of words that obey the alternation). The example in (3) represents word-final obstruent devoicing found in languages like Dutch (CHAPTER 69: FINAL DEVOICING AND FINAL LARYNGEAL NEUTRALIZATION). Again, *all* logically possible pairs that obey the alternation are included (which, in the case of Dutch, includes pairs like (ag, ak) even though Dutch lacks /g/).

(3) {(ab, ap), (ad, at), (ag, ak), . . . , (bab, bap), (bad, bat), (bag, bak), . . .}

The pairs can be taken to mean that underlying /ab/ is realized as [ap], and so on. When an underlying form *u* is paired with a surface form *s*, we write $\langle u, s\rangle$. Again, much depends on how the "three dots" are interpreted. If they are interpreted as strictly as possible, then the alternation could be generated by the *SPE* rule:

(4) $\begin{bmatrix}+\text{voice}\\ -\text{son}\end{bmatrix} \rightarrow [-\text{voice}]\ /\ \_\_\ \#$

In introductory phonology, it is often pointed out that the feature [+voice] in the target of the rule is unnecessary. In the interests of having shorter rules and rules which apply maximally without being falsified, the feature [+voice] is omitted in favor of the following.

(5) $[-\text{son}] \rightarrow [-\text{voice}]\ /\ \_\_\ \#$

This would mean that the "three dots" are intended to include pairs like those in (6).

(6) {(ap, ap), (at, at), (ak, ak), . . .}

Thus the rule in (5) applies even to hypothetical form like /ap/, mapping it to [ap]. Koskenniemi (1983) takes this one step further, and considers the "three dots" to include pairs like those in (7).

(7) {(a, a), (as, as), (af, af), (ar, ar), (an, an), . . .}

In other words, all hypothetical underlying forms are included in the left-hand side of some pair. The corresponding *SPE* rule could be said to apply to all possible underlying forms, though in most cases its application is vacuous. The application only results in a change when the final consonant is a voiced obstruent. Mainstream phonology never adopted this perspective, for two reasons: it made the standard *SPE* rules more difficult to write (and sometimes more complex according to the *SPE* simplicity metric), and there were some discouraging complexity results (Barton 1986; Barton *et al.* 1987). But Koskenniemi observes a conceptual shift when thinking of the patterns in this way: rules can be thought of as constraints on alternations.

Note that all phonological knowledge can be deduced from the alternation pattern. In the case of phonotactic knowledge, this is straightforward: the set of forms in the right-hand side (the "surface forms") of the alternation pattern constitutes the infinite set of all forms that obey the surface phonotactic constraints of the language. Similarly, the set of forms in the left-hand side (the "underlying forms") is an infinite set which constitutes all forms that obey the Morpheme Structure Constraints (MSCs) of the language (CHAPTER 86: MORPHEME STRUCTURE CONSTRAINTS). If the notion of the rich base is adopted, there are no morpheme structure constraints, and the left-hand side becomes all logically possible underlying forms. Finally, the alternation pattern itself determines the contrasts. Generally, any two segments that are mapped to the same surface segment in all contexts (i.e. are neutralized) are not contrastive. Because knowledge of MSCs, phonotactics, and contrasts can all be deduced from alternations, the problem of devising learners for phonological alternations is one of the most important frontiers of phonological learnability.

2.3 *Gradience*

The above discussion only makes binary distinctions of well-formedness. Recently, however, many phonologists have argued for the importance of gradient distinctions (Coleman and Pierrehumbert 1997; Zuraw 2000; Albright and Hayes 2003; Coetzee 2008; Hayes and Wilson 2008). Gradient distinctions have been used to model the confidence of speakers in the face of lexical exceptions, variation in the productions of individual speakers, and variation across speakers in experimental settings.

The scope of formal language theory is not limited to binary distinctions. The sets and relations above can be thought of as functions whose domain is all logically possible words, or pairs of words, and whose co-domain is simply 0 and 1, for "ill-formed" and "well-formed," respectively; i.e. as *indicator functions*. Phonological patterns can also be thought of as functions whose co-domain is real-valued. Moreover, if these values sum to one, then the function is a probability distribution.[1]

From the perspective of formal language theory and learning theories, the differences between indicator functions and distributions are not particularly significant. Consider the Chomsky Hierarchy:

(8) finite $\subset$ regular $\subset$ context-free $\subset$ context-sensitive $\subset$ recursively enumerable

This inclusion hierarchy classifies patterns (e.g. sets of (pairs of) forms) in terms of the complexity of the kinds of formal devices (e.g. grammars) needed to generate them (see, e.g. Harrison 1978; Hopcroft *et al.* 1979; Thomas 1997). A remarkable range of ways to characterize complexity all converge on the distinctions in (8), which is why the hierarchy is considered to be so illuminating.

Crucially, the place of a function in the Chomsky Hierarchy is entirely independent of whether its co-domain is Boolean or real-valued. For learning theory – and the central problem of generalization – the co-domain matters little. Vapnik (1998: 8) writes:

[1] Formally, let Σ^* be the set of all logically possible words given a finite alphabet Σ. A pattern L is an indicator function if $L : \Sigma^* \rightarrow \{0, 1\}$. It is real-valued if $L : \Sigma^* \rightarrow \mathcal{R}$ and it is a probability distribution iff $\Sigma_{w \in \Sigma^*} L(w) = 1$. If Δ is another alphabet, then $L : \Sigma^* \times \Delta^* \rightarrow \{0, 1\}$ is a Boolean alternation and $L : \Sigma^* \times \Delta^* \rightarrow \mathcal{R}$ is a real-valued one.

> Generalizing the results obtained for estimating indicator functions (pattern recognition) to the problem of estimating real-valued functions (regressions, density functions, etc.) was a purely technical achievement. To obtain these generalizations, no additional concepts needed to be introduced.

Thus the choice of binary *vs.* gradient distinctions should depend simply on *what* one is trying to model.

2.4 Properties of phonological patterns

What are the properties of phonological patterns? When we consider alternations, it is the case that they can be described with any grammar capable of describing regular relations (Johnson 1972; Kaplan and Kay 1981, 1994; Karttunen 1993, 1998; Eisner 1997a, 1997b; Riggle 2004).[2] It is known that all regular relations have regular domains and co-domains, so it follows that all phonotactic patterns are regular as well. This is a striking hypothesis in light of the fact that some syntactic patterns appear to belong to higher levels of the Chomsky Hierarchy (Chomsky 1956; Joshi 1985; Shieber 1985; Kobele 2006).

Though limiting phonology to regular patterns is a significant restriction, it is not nearly restrictive enough. For instance, consider a hypothetical stress pattern consisting of all forms with an even number of stressed syllables. This pattern is regular, but it is wildly unlike those observed in natural language (see e.g. Hayes 1995; Gordon 2002; CHAPTER 41: THE REPRESENTATION OF WORD STRESS). Furthermore, though assuming that phonology is regular provides significant structure to the hypothesis space, there are many learning frameworks where this is still too little structure to guarantee learnability.

Learning theorists are interested in the properties that make patterns learnable. Linguistic properties are just now beginning to be investigated foŗ their contributions to learnability. In the case of phonological patterns, it seems likely that the relevant properties will be subregular; that is, properties that carve out some proper subclass of the regular languages. Rogers and Pullum (2007) draw attention to the Subregular Hierarchy (McNaughton and Papert 1971), which classifies regular patterns according to the properties of different kinds of grammars capable of generating them. Additional recent work which attempts to relate phonological patterns to their place in the Subregular Hierarchy include Edlefsen *et al.* (2008), Graf (2010), and Heinz (2010).

3 Learning theory

3.1 Goals

There are many good resources on formal learning theory for phonologists. Nowak *et al.* (2002) provides an excellent, short introduction. Niyogi (2006) and de la Higuera (2010) provide detailed, accessible treatments, and Anthony and

[2] The notable exception to this is reduplication (CHAPTER 100: REDUPLICATION; CHAPTER 119: REDUPLICATION IN SANSKRIT), which is arguably a morphological process (Inkelas and Zoll 2005). For regular (finite-state) approaches to reduplication, see Roark and Sproat (2007).

Biggs (1992), Kearns and Vazirani (1994), and Jain *et al.* (1999) provide technical introductions. Here we summarize some of the main ideas and common results.

Learning theory characterizes learning, and the necessary and sufficient conditions required for learning strategies to be successful, or to exhibit some other particular behavior. This focus on characterizing a learner's behavior helps us understand precisely *why* a particular learning strategy succeeds in some cases, and helps us to characterize the class of cases where it may fail.

Learning theory defines learners as functions which map experience to grammars. The experience of the learner is necessarily finite, but the target languages typically are not. Any learning procedure can be characterized in this way, including learners that are connectionist (e.g. Rumelhart and McClelland 1986) or Bayesian (Griffiths *et al.* 2008), and learners based on maximum entropy (e.g. Goldwater and Johnson 2003), as well as those embedded within generative models such as Recursive Constraint Demotion (Tesar 1995; Tesar and Smolensky 1998) and minimal generalization (Albright and Hayes 2002; Albright 2009). Results of formal learning theory apply to all of these particular proposals and many others.

By characterizing learning algorithms as functions, it is possible to focus on the functional behavior of a learning strategy rather than its procedural description. This allows one to identify relevant properties of the mapping – like the linguistic typology predicted by a function's range – that are independent of the algorithm's implementation. Moreover, these properties are often crucial in understanding precisely what kinds of patterns learners are guaranteed to learn, and where they can fail.

Learning functions can also be characterized in terms of their computational complexity. Some learning procedures may require unreasonable resources and time. The exact meaning of "unreasonable" is studied in a number of works, including Pitt (1989) and de la Higuera (1997).

3.2 *Learning frameworks*

In §3.2.1–§3.3 we survey three learning frameworks: Identification in the Limit from Positive Data (Gold 1967), Probably Approximately Correct learning (Valiant 1984), and the Mistake Bounds model (Littlestone 1988). Other frameworks are discussed in §3.3.1, and the major results of learning theory are given in §3.4. Across the frameworks, precisely the same conclusion explicates the necessity of (some form of) Universal Grammar: namely, without a structured, restricted hypothesis space, feasible learning is impossible.

3.2.1 *Identification in the Limit from Positive Data*

In the Identification in the Limit from Positive Data (ILPD) framework, there are no limits on the learner's computational resources or time, and the input is assumed to consist of an infinitely long noise-free text that contains at least one instance of every form in the target pattern. Learners are partial functions, which map initial finite portions of these texts to grammars. A learner is said to *converge* to a grammar G if and only if at some finite point every future hypothesis is G. The learner is said to identify a language (or class of languages) in the limit just in case the learner converges to a grammar that generates the target language for *any* text from the target language (for any member of the class of languages).

Though the learner's input is generously assumed to include every potentially useful finite collection of forms from the target, the criterion for success is very strict: for *any* logically possible text for the language, the learner must find a grammar that generates the target pattern without a single deviation. This scenario focuses the learning problem squarely on generalization. Given as much *finite* experience as desired, can any learning device, no matter how powerful, exactly learn some language pattern, which may be *infinite* in size? How can the learning device cover the gap between its finite experience and the infinite set which represents the capacity of a normal speaker?

3.2.2 *Probably Approximately Correct learning*

Another model that has received a great deal of attention in learning theory is the Probably Approximately Correct (PAC) framework, first introduced by Valiant (1984) and subsequently developed by Kearns *et al.* (1987), Angluin (1988a), and Blumer *et al.* (1989), among others.[3] This model offers a probabilistic perspective on efficiently learning a class of languages in terms of the probability of attaining a hypothesis that has a low likelihood of making errors modulo the number of training samples observed.

The input to the learners is determined by drawing elements from the instance space X of data points, according to a probability distribution Π. Instead of exact identification, the quality of a learner's hypothesis h is evaluated in terms of the probability that h disagrees with the target language l for any $x \in X$, randomly drawn according to distribution Π. The ingenious aspect of the PAC model is that it does not matter what the distribution is, only that the same distribution Π is used for training and for evaluation.

The *error* of a hypothesis, denoted *error(h)*, is the sum of the probability that Π assigns to data points where h disagrees with the target. Analysis of learnability in the PAC model centers on the following question:

(9) For a given level of error ε, if a learner is presented with m samples drawn from X at random according to Π, what is our confidence δ that *error(h)* of the learner's hypothesis h is less than ε?

For a language class $\mathcal{L}$ and any given learning strategy, the *sample complexity* is the number of samples m needed to ensure that, for any $l \in \mathcal{L}$ and *any* distribution Π, the likelihood is at least δ that a learner will generate a hypothesis whose error is at most ε. This leads to the following definition of PAC-learnability:

(10) $\mathcal{L}$ is PAC-learnable iff the sample complexity of $\mathcal{L}$ is a polynomial function of ε and δ.

The PAC-learning framework differs from IDLP in two important respects. In one sense, the PAC model is more stringent because the required training data and computation must be *feasible* (i.e. polynomial). But, in another sense, the PAC model

[3] See Haussler *et al.* (1992) and Haussler (1995) for overviews of work in learnability theory and insights into the deep connections between the PAC, Bayesian, and mistake-bound perspectives.

is less stringent than the Gold model, in that it loosens the definition of success from exact identification of a language to *approximate* identification that is likely to be correct *most* of the time.

3.3 *Mistake bounds*

Littlestone (1988) observes that, in many cases of interest, learnability can be characterized by the fact that the number of mistaken classifications – and subsequent corrections – is bounded. In this online framework, a learning algorithm $\mathcal{A}$ must classify each form it observes according to its current hypothesis h, which may be updated after the correct classification is revealed. The *mistake bound* for $\mathcal{A}$ on language class $\mathcal{L}$, denoted $M_{\mathcal{A}}(\mathcal{L})$, is the number of mistaken classifications that $\mathcal{A}$ might make when facing a diabolical adversary who knows $\mathcal{A}$'s strategy and has boundless computing resources to choose the hardest language in $\mathcal{L}$, and the least helpful presentation of examples. The *optimal mistake bound* for $\mathcal{L}$, denoted $Opt(\mathcal{L})$, is the smallest $M_{\mathcal{A}}(\mathcal{L})$ for any possible $\mathcal{A}$.

Littlestone (1988) shows that if $Opt(\mathcal{L})$ is finite, then it is the case that the class $\mathcal{L}$ is both identifiable in the limit and PAC-learnable. The converse, however, does not hold; neither PAC nor Gold learnability guarantees a finite mistake bound. In the former case there might be an infinite sequence of imperfect hypotheses that all have error less than ε, and in the latter case one might be able to guarantee that the number of mistakes will be finite without being able to give a specific bound.

3.3.1 *Other frameworks*

There are other learning frameworks. Some enrich the learner's input in particular ways, which gives the learner more information and generally leads to stronger positive results. For example, Gold (1967) also considers the case of learning from positive and negative data. In this scenario, the entire class of recursively enumerable languages is learnable in principle, though no learner is efficiently computable for even the regular languages (Gold 1978). Gold also shows that restricting texts to those with certain useful kinds of structure (for example, by only allowing texts whose structure is describable with primitive recursive functions; see also Rogers 1967) can also guarantee the learnability of the recursively enumerable languages. This means that knowing crucial properties of the presentations of the data can, like negative evidence, make a huge contribution to pattern learning. However, it is highly doubtful that the natural language data children observe have either of these properties (note that occasional overt corrections do not necessarily constitute negative evidence).

Similarly, Horning (1969) shows that, when learning stochastic languages (distribution learning), if it is the case that learners are required to succeed only on texts generable by the target distribution then it follows that probabilistic context-free languages can be learned (see also Osherson *et al.* 1986). Angluin (1988b) supersedes Horning's result, and shows that the entire class of recursively enumerable distributions is in fact learnable in this sense. Like Gold's result above, these results suggest that knowing properties of the presentations of the input data dramatically increases what is learnable in principle. Crucially, however, the learners in these proofs are not remotely feasible, so these results do not inform human language learning.

3.4 Main results

In the ILPD, PAC, and MB frameworks surveyed above, there are significant negative results: none of the major classes in the Chomsky Hierarchy is learnable. In the case of identification in the limit from positive data, no class which is a proper superset of the finite languages is learnable. In the PAC and MB models, not even the finite class of languages is learnable. In other words, there is no learner, not even in principle, that can PAC-learn or identify in the limit from positive data all regular, context-free, or context-sensitive language patterns.

There are many ways to interpret this result (see, for example, Pinker 1979). Gold mentions restricting the problem so that not *all* regular (or context-sensitive) patterns are permitted in natural language. This possibility is promising for three reasons. First, the field of grammatical inference has identified many classes of languages that are ILDP and PAC-learnable (Angluin 1982; Muggleton 1990; Clark and Eyraud 2007; Heinz 2008; de la Higuera 2010). Many of these classes contain infinitely many patterns, and some include context-free, even context-sensitive patterns. In virtually every case, the successes occur because the language classes *are non-arbitrary* in important ways: the hypothesis space is structured. Secondly, this possibility makes sense from the studies of distribution learning above: while recursively enumerable distributions are learnable in principle they are not feasibly learnable in practice. The efficiency issues can be overcome by restricting the class of distributions to be learned (if doing so adds sufficient structure to the hypothesis space). Finally, this possibility also matches well with language typologists' repeated observations that the extensive variation that exists in natural languages appears to be limited, though stating exact universals is difficult (Greenberg 1963, 1978; Mairal and Gil 2006; Stabler 2009).

The results surveyed above lead to the following conclusion: structure matters. In particular, if the collection of language patterns to be learned has the right kind of structure – the right kind of properties – then learning is possible. The most interesting learners will use the structure or properties in the language class to license the right generalizations from their finite experience to an infinite pattern. Conversely, these results show that there is essentially no hope of learning in cases where the range of possible patterns is too unstructured.

4 The role of structure in generalization

The *structure* of the hypothesis space is what allows for generalization. In this section, we discuss very general structural properties important to learnability. We begin with a discussion of finite hypothesis spaces, then turn to structure related to what has been called the *subset problem*, and conclude with a general metric of structure known as the Vapnik–Chervonenkis Dimension.

4.1 Finiteness as a kind of structure

Many linguistic theories, such as Principles and Parameters and Optimality Theory, only allow finitely much variation in the typology, thereby providing a finite collection of languages. This property of hypothesis spaces is a sufficient property for success in many frameworks, including PAC and IDLP. A common brute-force strategy for any finite hypothesis space is to essentially match all grammars with the learning data, and choose the one that is the most consistent.

Although it is a sufficient property for learning, finiteness is hardly an interesting property. To see why, recall the earlier discussion of quantity insensitive (QI) stress patterns, and let us artificially place an upper bound on word length so that we only have finitely many patterns to consider. For ease of illustration we set the bound at four). If we restrict ourselves to just one string of each length, then there are 2^{10} = 1,024 logically possible patterns, of which eight are shown in (11).

(11) *Some logically possible stress patterns over 1–4 syllables*

	natural QI stress systems	*unnatural systems*
Initial	1 10 100 1000	0 01 100 0101
Final	1 01 001 0001	1 10 101 0110
Edges	1 11 101 1001	1 00 000 1010
Binary-initial	1 10 101 1010	1 01 101 1100

The artificial bound limits the class in a very significant but uninteresting way, because almost all of these 1,024 patterns belong in the "unnatural" column. The properties that determine which patterns belong to the "natural" column are going to be precisely those same linguistic properties that hold regardless of whether the class is finite or infinite. It is of far greater interest how those properties – and not finiteness – structure the hypothesis space.

Finiteness is hardly a necessary property for learnability – many infinite language classes are efficiently learnable because they have structure that learners can utilize (Jain *et al.* 1999; de la Higuera 2010). On the other hand, brute-force learners that simply traverse an enumeration of all hypotheses are not generally feasible (since finite classes can still be very large). Even for the finite case, the interesting learners are those that make use of structure (see e.g. Recursive Constraint Demotion; Tesar and Smolensky 2000).

4.2 *Tell-tale sets and the subset problem*

Angluin (1980) provides one benchmark for necessary and sufficient structure in a hypothesis space. If every language pattern L in the hypothesis space contains a finite set S, such that no other language pattern L' in the hypothesis space is simultaneously a superset of S and proper subset of L (see Figure 3.1), then this hypothesis space is sufficiently structured to be identified in the limit from positive data. The finite set S is called a *tell-tale set*, and we call the above property of hypothesis spaces the *tell-tale property*.

The tell-tale property is sufficient for learning, because a learner that guesses L after exposure to its tell-tale set is guaranteed to have hypothesized the smallest language in the class consistent with the sample. Characterizing the tell-tale sets of a hypothesis space – and more generally, characterizing the finite experience a learner needs to generalize correctly to the language patterns in a hypothesis space – is one of the important lessons of learning theory. It adds to the functional characterization of the learner. This is because once the tell-tale sets are characterized, when given a learner and a language pattern L from the learner's

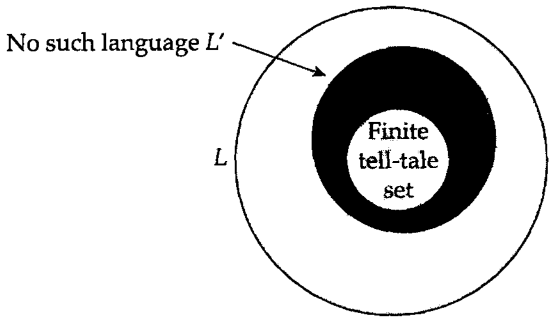

Figure 3.1 The tell-tale property

hypothesis space, one knows whether the learner will succeed given some finite sample S (by checking whether S is a tell-tale set).

Identifying properties of the tell-tale sets is important to phonologists for two reasons. First, it provides an additional way to evaluate learning proposals, since the tell-tale sets can be compared with the actual linguistic forms available to children. Secondly, knowledge of the properties of tell-tale sets allows one to understand how the learner generalizes, and may provide insight into stages of the learning process.

4.3 *The Vapnik–Chervonenkis dimension*

One particularly simple and robust metric of structure for concept classes is the combinatorial measure of complexity known as the Vapnik–Chervonenkis (VC) dimension (Vapnik 1998; Vapnik and Chervonenkis 1971). For a given concept class $\mathcal{L}$, the VC dimension (VCD) of $\mathcal{L}$ is the cardinality of the largest set of data S such that there is at least one language in $\mathcal{L}$ for each of the $2^{|S|}$ possible ways of labeling the data points in S as "ungrammatical" or "grammatical." If S has this property it is said to be *shattered* by $\mathcal{L}$.[4] If sets of arbitrary size are shatterable then the VCD is said to be infinite.

For an illustration, suppose that we represent coda clusters as points in $\mathcal{R}^2$ (the x–y plane) where the x-axis encodes the sonority of the second consonant and the y-axis the sonority of the first. Suppose further that $\mathcal{L}$ is the (infinite) set of languages corresponding to "half-spaces" defined by straight lines that split $\mathcal{R}^2$ into two regions, one for licit clusters and the other for illicit clusters. Figure 3.2 provides a rough example that situates the clusters *sn, pl, pt* in $\mathcal{R}^2$. In this example, a grammar that includes all three clusters can be obtained by drawing a line off to one side so that the illicit (shaded) area does not include the points. Grammars that include any two of the points can be obtained by drawing a line between the point to be excluded and the other two, and shading the side with the excluded point. These four possibilities make up the top row of Figure 3.2. The other four possibilities are illustrated in the bottom row of Figure 3.2; these are obtained by inverting the grammars in the top row. Since there is a grammar (i.e. a half-space)

[4] Formally, sample $S = \{x_1, \ldots, x_n\} \subseteq \mathcal{X}_n$ is shatterable if $\forall(v_1, \ldots, v_n) \in \{0, 1\}^n$, $\exists l \in \mathcal{L}$ such that $\forall i\ c(xi) = v_i$. The VC dimension of $\mathcal{L}$ is the cardinality of the largest shatterable sample: $vcd(L) = max\{|S| : S \text{ is shatterable}\}$.

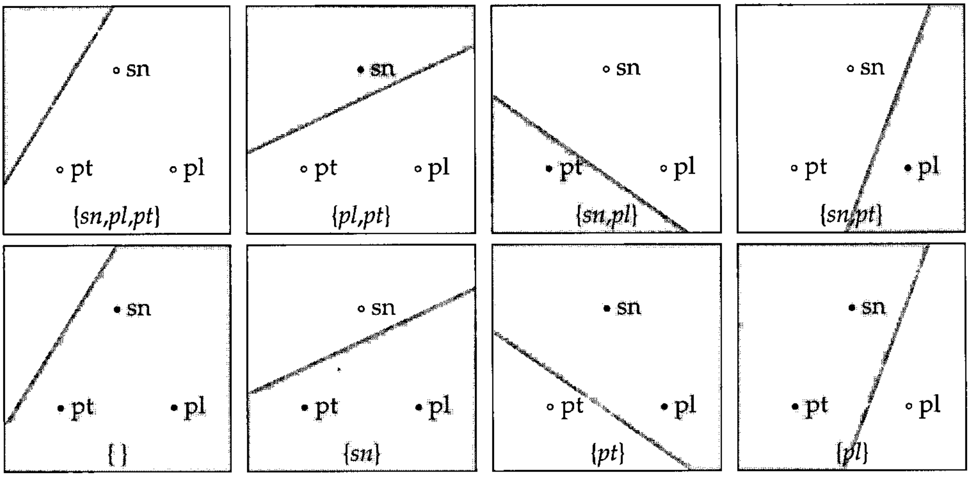

Figure 3.2 A set of three points that is shatterable by half-spaces in $\mathcal{R}^2$

for each of the eight logically possible grammatical/ungrammatical distinctions over these three points, we say that this set of points is shattered by the class $\mathcal{L}$.

Because there is a shatterable set of three points, the VC dimension of $\mathcal{L}$ is at least three. This does not entail that *every* set of three points is shatterable. For instance, any set of three collinear points cannot be shattered, because one of the points lies directly between the other two and thus cannot be separated by a half-space. This same logic explains why no set of four points in $\mathcal{R}^2$ is shatterable; either one point is inside a triangle whose corners are the other points, or the four points are the corners of a convex polygon. In the former case, no hyperplane can include the interior point while excluding all the points at the corners and, in the latter case, no hyperplane can include two opposing corners while excluding both points at the other corners. The fact that there are shatterable sets of three points, but no shatterable sets of four, places the VC dimension of half-spaces in $\mathcal{R}^2$ at three (for $\mathcal{R}^n$ it is n+1). What this means in terms of learnability is that, for any dataset with more than three points, there *must* be points whose grammaticality is interdependent.

To understand the role this structure plays in learning, consider a class $\mathcal{L}$ whose VC dimension is d and a learner with m data points. As long as $m \leq d$, it is possible that the labeling of every point is totally independent of the others. But, as soon as $m > d$, some generalization/prediction is always possible, because there are fewer than 2^m distinct ways to label the data points as grammatical or ungrammatical according to languages in $\mathcal{L}$ (otherwise the VCD would be higher). Furthermore, it turns out that when $m > d$, the number of possible labelings is a *polynomial* function of m. In essence, there is a sort of "phase transition" from exponentially many labelings when $m \leq d$ to only polynomially many when $m > d$, which makes the complexity of the hypothesis space a polynomial function of m when the VCD is finite.[5]

[5] By complexity, we mean information-theoretic complexity in the sense that $\mathcal{L}$ makes it possible to describe any labeling of $m > d$ data points with fewer than $\log_2 m$ bits. See Kearns and Vazirani (1994) for more discussion and for proofs.

A finite VCD is both necessary and sufficient for PAC-learnability (Blumer *et al.* 1989), and thus not even the class of finite languages (which has infinite VCD) is learnable in the PAC framework. It follows that none of the major classes of the Chomsky Hierarchy are PAC-learnable, which again suggests that the right characterization of the class of patterns in phonology is some class that cross-cuts the Chomsky Hierarchy.

Finally, we should note that substantive linguistic restrictions will shrink the VC dimension below the upper bound that follows from more general structural properties of the class. For example, in the grammars of Figure 3.2, phonetic factors such as ease of articulation or perception might rule out the possibility of languages that admit [pt] clusters while excluding [sl] and [sn] clusters. When additional properties such as implicational universals over sonority sequencing restrict a concept class, the VCD can quantify the structure that such factors bring to the learning problem.

4.4 *Summary*

The three kinds of structure surveyed here – finiteness, the tell-tale property, and the VC dimension – provide a foundation for phonologists to investigate the contribution phonological properties make to learning. Phonologists widely agree that there is intricate structure in phonological patterns. How this phonological structure relates to the structures that are relevant to learnability is a promising new research area.

5 Phonological learners

5.1 *Learning rule-based alternations from pairs*

Johnson (1984) presents an algorithm that takes as input a set of $\langle u, s\rangle$ pairs and returns segment substitution rules and their orderings that are logically consistent with the data. The class $\mathcal{L}$ of all languages (sets of $\langle u, s\rangle$ pairs) that are representable by ordered sequences of substitution rules is superfinite, and thus we know that this strategy cannot identify $\mathcal{L}$ in the limit from positive data. Johnson notes that this set of induced rules and orderings can be reduced via evaluation metrics and other heuristics grounded in language universals. The need for the latter shows that, while logical properties of phonological rules can restrict the hypothesis space, additional structure in linguistic systems must play a role in choosing among hypotheses.

Gildea and Jurafsky (1996) present an algorithm that takes as input $\langle u, s\rangle$ pairs from a dataset with some alternation, and returns a rule, which unlike Johnson's system can include deletion and epenthesis. Their work begins with a result from Oncina *et al.* (1993), who present an algorithm dubbed OSTIA, which identifies in the limit from positive data a subclass of regular relations describable by subsequential finite state transducers. Since the flapping rule of English can be represented this way, Gildea and Jurafsky ask whether OSTIA will acquire the

[6] Since the CMU dictionary does not include allophonic information, Gildea and Jurafsky modified the dictionary to replace [t] and [d] with [ɾ] in every instance where the rule would apply.

flapping rule from an appropriately modified version of the *Carnegie Mellon University Pronouncing Dictionary* (CMU 1993).[6] Essentially, they ask whether the CMU contains a tell-tale set (for OSTIA). Because OSTIA fails to learn the flapping rule from the CMU dictionary, the answer is no, probably because a tell-tale set would need to include non-English forms like *ttt*.[7]

Gildea and Jurafsky then augment OSTIA with three phonologically motivated principles. These are Faithfulness: underlying–surface pairs tend to be similar; Community: similar segments tend to behave similarly; and Context: phonological rules can access variables in their context. This modified OSTIA algorithm gets much closer to acquiring a rule that represents the English flapping alternation. Gildea and Jurafsky conclude that these biases aid learning, and argue for a research program for evaluating the contributions of such biases. We agree wholeheartedly; domain-appropriate biases that add structure to or otherwise reduce the hypothesis space are likely to aid learning by also reducing the size of tell-tale sets. However, it is critical to ask exactly how and why this occurs, and most crucially what class of rules are learnable with the biases in place. To our knowledge, neither of these interesting questions has been addressed.

Albright and Hayes (2003) also aim to learn alternations expressed by phonological rules. Their algorithm takes as input $\langle u, s\rangle$ pairs and returns a set of rewrite rules with confidence scores. A central idea in their rule construction procedure is a strategy called minimal generalization. The idea is that if two sounds are known to undergo some alternation, then one may conclude that all sounds in the smallest natural class containing those two sounds also undergoes the alternation (cf. the Community principle). In addition, the algorithm assigns a confidence score to each rule based on the frequency of the rule's application in the corpus. The confidence score can be used to analyze free variation, or phonologically conditioned allomorphy (as with the irregular English past tense).

Albright and Hayes do not focus on an analytical characterization of the class of languages that their algorithm can learn, but instead compare the behavior of their algorithm to the judgments of native speakers on "wug" tests (Berko 1958; CHAPTER 96: EXPERIMENTAL APPROACHES IN THEORETICAL PHONOLOGY). These comparisons reveal intriguing correlations, but they are somewhat difficult to interpret. On the one hand, a shift in focus from the analysis of properties that define various learnable classes of languages to the behavior of humans is undoubtedly appealing to any who feel that the results of learnability theory are too abstract and remote from real-world learning problems. On the other hand, having observed that an algorithm $\mathcal{A}$ and human subject $\mathcal{H}$ give similar responses for a particular set of test items T after being exposed to a set of training data D, it is not clear what we can conclude about $\mathcal{H}$ or the relationship between $\mathcal{A}$ and $\mathcal{H}$, because they might wildly diverge for some other data T' and D'. The goal of determining which properties of the data critically underlie learnability – or in this case the correlation between $\mathcal{A}$ and $\mathcal{H}$ – is precisely why learning theory focuses mainly on the

[7] It should be emphasized that OSTIA learns a rule that is consistent with the data. It is just that the alternation that this rule describes is not the same infinite set of ⟨underlying form, surface form⟩ pairs that phonologists think the flapping rule ought to describe.

properties of classes of languages or the *general behavior of specific algorithms*, as opposed to the specific behavior of specific algorithms.

5.2 *Learning OT grammars*

Optimality Theory (Prince and Smolensky 1993) is a theory of grammar which characterizes alternations by a strict ranking of constraints which evaluate possible $\langle u, s \rangle$ pairs. A $\langle u, s \rangle$ pair belongs to the alternation just in case it is optimal among the (possibly infinite) range of $\langle u, s' \rangle$ pairs according to the ranked constraints. (CHAPTER 63: MARKEDNESS AND FAITHFULNESS CONSTRAINTS).

For a fixed (universal) set of k constraints there are at most $k!$ languages and thus any set of constraints defines a finite class of languages that is learnable in the limit. Though the members of any finite class of languages can be identified in the limit by enumerating the languages, such an approach is not feasible in practice. An early positive result for OT learning was provided by Tesar and Smolensky's (1993) Recursive Constraint Demotion (RCD) algorithm. Tesar and Smolensky (1996: 26) subsequently showed that the structure that ranked constraints given to the hypothesis space guarantees that RCD will successfully identify languages with a polynomial mistake bound (unlike a brute-force enumeration).

5.3 *The VC dimension of OT and HG*

As mentioned earlier, finitude is itself a very simple kind of structure for concept classes. With regard to the VC dimension, this is reflected by the fact that the VCD of any finite set of grammars is at most $\log_2$ of the cardinality of the set. This follows because it takes at least 2^n concepts to shatter a set of n data points. Hence the VCD of any set of OT grammars over a fixed set of k constraints is at most $\log_2 k!$, because there are only $k!$ possible rankings. By contrast, if we take the same constraints and consider grammars defined by real-valued *weightings* (as in Harmonic Grammar; HG)[8] there are infinitely many possible grammars and thus no a priori bound on the VCD.

This pair of cases proves to be quite illuminating. Though the finitude of $\mathcal{L}$ (or lack thereof) provides some information about its learnability, the characterization is both coarse and incomplete. In the case of OT, the finitude of the concept class bounds the VC dimension at $\log_2$ of $k!$ (which is on the order of $k \log_2 k$). Unsurprisingly, the hypothesis space has more structure than its mere finitude, and this structure bounds OT's VC dimension at k–1 (Riggle 2009). By contrast, one might expect the infinite hypothesis space of HG to have much less structure, but it turns out that learning weightings can be represented as the problem of learning half-spaces in $\mathcal{R}^k$ (as in Figure 3.2), so the VC dimension cannot be greater than k+1 and in fact is k–1 (Bane *et al.* 2010). This parity means not only that both models are efficiently learnable, but that the learning problems are essentially of equal complexity (recalling Vapnik's observation in §2.3).

[8] In addition to HG (Legendre *et al.* 1990; Smolensky and Legendre 2006; Pater 2009), a range of weighted models have been proposed by Goldsmith (1990, 1991, 1993a, 1993b) and a few others.

5.4 PAC learning of rankings and weightings in OT and HG

Both OT and HG have the same VC dimension: k–1 for grammars with k constraints. For a concrete example of what this means in terms of learnability, consider the three hypothetical tableaux in (12).

(12)

input 1	c_1	c_2	c_3	c_4	
Cand *a*		*			implication: $a > b$ iff $w_1 > w_2$ in HG or $c_1 >> c_2$ in OT
Cand *b*	*				implication: $b > a$ iff $w_2 > w_1$ in HG or $c_2 >> c_1$ in OT

input 2	c_1	c_2	c_3	c_4	
Cand *c*			*		implication: $c > d$ iff $w_2 > w_3$ in HG or $c_2 >> c_3$ in OT
Cand *d*		*			implication: $d > c$ iff $w_3 > w_2$ in HG or $c_3 >> c_2$ in OT

input 3	c_1	c_2	c_3	c_4	
Cand *e*				*	implication: $e > f$ iff $w_3 > w_4$ in HG or $c_3 >> c_4$ in OT
Cand *f*			*		implication: $f > e$ iff $w_4 > w_3$ in HG or $c_4 >> c_3$ in OT

In both OT and HG it is possible to formulate sets of k–1 binary tableaux like those in (12), in which each of the exponentially many (i.e. 2^{k-1}) ways to choose a set of winners is possible under some grammar. However, as soon as a learner has seen k or more tableaux – in either model – there are only polynomially many ways to choose a set of winners (i.e. there is no set of four tableaux in which all patterns of winners are possible). The remarkable consequence of this fact is that any learner that meets the simple condition that its hypotheses are always consistent with all previous observations is guaranteed to PAC-learn a ranking/weighting from a set of training data whose size is a linear in the number of constraints.[9]

Given a constraint set and a dataset comprising ⟨winner, loser⟩ pairs, Recursive Constraint Demotion (Tesar and Smolensky 1993, 1998, 2000; Tesar 1995, 1997, 1998a, 1998b) constructs a stratified hierarchy $\mathcal{H}$ (i.e. a weak ordering) that is consistent with the data by constructing strata consisting of constraints for which, in each remaining $\langle w, l\rangle$ pair, w has no more violations than l, and then discarding any pair in which w is optimal according to the $\mathcal{H}$ constructed thus far. This process is reiterated until all $\langle w, l\rangle$ pairs are gone (or until no constraint favors a winner, in which case no ranking is consistent with the data). If, in addition to $\mathcal{H}$, RCD records the ranking conditions that support its correct predictions, then it can generate hypotheses consistent with all previous observations and thereby be guaranteed to PAC-learn rankings from in the order of k random samples (the extra record-keeping is needed to ensure consistency because "accidentally" correct predictions can be undone by subsequent updates to $\mathcal{H}$).

[9] The bound on sample complexity m, according to VC dimension d, is $m \leq [(4/\varepsilon)\ [d \ln (12/\varepsilon) + \ln (2/\delta)]$; see e.g. Blumer *et al.* (1989).

For HG grammars, Potts *et al.* (2010) propose a consistent learner that finds a constraint-weighting $w = \langle w_1, w_2, \ldots, w_k \rangle \in \mathcal{R}^k$ that simultaneously satisfies all the linear inequalities that correspond to a set of ⟨winner, loser⟩ pairs – such as those in (12) – using a technique from linear programming called the *simplex algorithm* (see e.g. Papadimitriou and Steiglitz 1998: chapter 2). Though their learner is intended to operate over batches of $\langle w, l \rangle$ pairs, they could conceivably be recast as an "error-driven" learner, so that, rather than generating a new hypothesis for each new datum based on all prior observations, a new hypothesis would be generated only in the event of an erroneous prediction.

RCD also has an error-driven formulation, and an especially useful property of error-driven learners is that they only need to "remember" data points that they misclassified (often called "supports") in order to faithfully reconstruct correct predictions for all forms in the training sequence. This allows a mistake bound to double as a memory bound on the amount of information that a learner could ever need to store.

Pater (2008) observes that Rosenblatt's (1958) "perceptron" can be straightforwardly applied to HG learning. The perceptron is an error-driven learner that maintains a weighting $w = \langle w_1, w_2, \ldots, w_k \rangle \in \mathcal{R}^k$, with which they make predictions as follows. For candidates a and b, the value $\Delta(a, b) \in \mathcal{Z}^k$ is the result of subtracting b's violations from a's violations (e.g. in (10), $\Delta(a, b) = \langle -1, 1, 0, 0 \rangle$). This point in k-dimensional space is "in" just in case it lies within the half-space described by w (i.e. if the inner product $w \cdot \Delta(a, b)$ is greater than zero; this is a linear-classifier like the ones in Figure 3.2). Upon misclassifying a data point, the hyperplane represented by the weight-vector w is nudged in the direction of $\Delta(a, b)$. Though multiple errors on the same data point are possible (i.e. the update rule is non-corrective), the perceptron is guaranteed to eventually converge to a correct weighting if one exists. In the general case, the perceptron is not a PAC-learner, because the sample complexity can be exponential in k when the probability mass of Π is concentrated on positive and negative data points that are packed too close to the hyperplane that separates them. Moreover, though the perceptron will converge eventually, it is precisely these "hard" probability distributions that lead to many mistakes.

5.5 *Mistake bounds in OT and HG*

Regarding optimal mistake bounds, Littlestone (1988) shows that, while a lower bound on $Opt(\mathcal{L})$ is set by $\mathcal{L}$'s VC dimension, in cases where $\mathcal{L}$ is finite, the upper bound of $Opt(\mathcal{L})$ is $\log_2|\mathcal{L}|$. This follows because the strategy of making predictions that accord with a plurality of the hypotheses consistent with previous observations only errs on data points that half or fewer of the remaining hypotheses correctly classify (else the correct prediction would have been made) and, as such, each error halves the set of viable hypotheses which allows no more than $\log_2|\mathcal{L}|$ errors.

This suggests room for improvement over RCD's quadratic mistake bound of $k(k{-}1)/2$, which follows from the maximum number of stratified hierarchies that RCD can entertain on the way from all k constraints in a single stratum to a total order (Tesar and Smolensky 1996: 26). To implement Littlestone's halving algorithm for OT, Riggle (2008) proposes a recursive function for calculating the fraction of the space of possible rankings that are consistent with a set of optimal candidates, a quantity he calls the r-volume. For just two candidates a and b, if A denotes the constraints for which *a* has fewer violations and *B* those for which *b* has fewer violations, then the fraction of rankings that select *a* is precisely |A| / (|A|+|B|).

(13)

input	c_1	c_2	c_3	c_4	
Cand *a*		*	*	*	the *r*-volume of candidate *a* is ⅓ (i.e. 8 rankings)
Cand *b*	**		*		the *r*-volume of candidate *a* is ⅔ (i.e. 16 rankings)

Unfortunately, though computing *r*-volume for larger sets of candidates can often be done in ways vastly more efficient than exhaustive search, there are "hard" cases where computation will always be intractable.[10] This highlights the core tension between power and efficiency in learning; RCD's mistake bound may be sub-optimal but it is still polynomial and it is obtainable at amazingly low computational cost, whereas the halving algorithm yields a nearly optimal mistake bound (i.e. within a logarithmic factor of k–1), but does so by introducing computation that is intractable in the worst case.

Analysis of mistake bounds illuminates a significant point of divergence between OT and HG. Though the two models have the same VC dimension, the mistake bound of the former is finite, while the mistake bound of the latter is not. This is so because it is possible to construct a sample sequence of arbitrary length in which each new data point causes an error that leads to an ever smaller change in the weighting. Thus, though learners that use strategies such as the perceptron algorithm will eventually converge to a correct constraint weighting for any HG grammar (see Pater 2008), there is no general bound on the rate of convergence (i.e. the number of mistakes along the way) that holds for all possible sets of training data.

Partially due to this fact, much of the work on learning linear classifiers has focused on the way that specific properties of samples affect learnability. For instance, the quantity γ, known as the *margin*, measures the distance (in high dimensional space) between the grammatical and ungrammatical points and the line that separates them. Given γ, one can derive bounds on the number of mistakes and the rate of convergence. In fact, if the margin is large enough, it supplants the dimensionality of the sample space in determining the VC dimension of the learning problem. Thus, with large margins, HG grammars with thousands of constraints might nonetheless have very low mistake bounds and sample complexity, suggesting that searching for so-called large-margin classifiers might provide linguistic insights.

5.6 Learning segmental adjacency patterns

Hayes and Wilson (2008) develop a learner that takes as input a list of words and outputs a maximum entropy grammar consisting of a finite set of weighted constraints that define a probability distribution over forms. The algorithm has several properties of interest. First, the constraints it returns are essentially *n*-grams and thus, in its simplest form, the algorithm can learn adjacency patterns, but not harmony patterns. Secondly, the units in these constraints are feature bundles denoting natural classes. Thirdly, the algorithm is designed to first search for more general constraints (i.e. those with smaller *n* and fewer features). Fourthly, following the

[10] This follows from the fact that pairs of candidates can be used to define partial orders over the constraints and the fact that the problem of counting the linear extensions of partial orders is #*P*-Complete.

principle of maximum entropy, the model weights constraints so that their observed number of violations in the training data matches the expected number.

The authors provide case studies using corpus data suggesting that phonological features play a crucial role in generalization. However, Albright (2009) explores feature-based generalization in Hayes and Wilson's model, as well as one based on minimal generalization, and shows that the specific contribution features make to learning remains unclear (CHAPTER 17: DISTINCTIVE FEATURES). This is an interesting class of models, and the phonological biases with the hypothesis space are in many ways appealing. However, as with the biases in Gildea and Jurafsky (1996), formal analysis of their contribution is needed.

5.7 *Learning Harmony Patterns*

Hayes and Wilson (2008) show that when representations are enriched by allowing segments with certain features to project onto tiers (where segments without such features are not projected) (see CHAPTER 14: AUTOSEGMENTS; CHAPTER 105: TIER SEGREGATION), if the algorithm is allowed to search for *n*-gram-like constraints on these additional levels of representation then it is possible to learn long-distance phonotactic constraints (i.e. harmony; see CHAPTER 91: VOWEL HARMONY: OPAQUE AND TRANSPARENT VOWELS). Hayes and Wilson (2008: 32) conclude that "in controlled comparative simulations, [tiers] makes phonotactic learning possible where it would not otherwise be so." It is, however, critical to bear in mind that this result tells us something about a particular *algorithm*, and not something about the linguistic phenomenon of *harmony* (i.e. a class of languages). Indeed, Heinz (2007, 2010) shows that long-distance phonotactic constraints can be learned without tiers (see below). Furthermore, the tiers that are critical to the success of the algorithm are taken by Hayes and Wilson to be antecedently given, but this does not entail (nor do the authors claim) that they must be antecedently given. Goldsmith and Riggle (forthcoming) offer a strategy for learning long-distance patterns that has many similarities to Hayes and Wilson's approach, but begins with an algorithm from Goldsmith and Xanthos (2009) for "discovering" tiers via unsupervised categorization of the sounds of corpus into vowels and consonants.

Heinz (2007, 2010) shows that phonotactic patterns derived from long-distance agreement patterns (Hansson 2001; Rose and Walker 2004) can be learned without tiers, using the notion of a discontiguous subsequence of length two. This idea is similar to bigram learning where learners keep track of contiguous subsequences of length two. Heinz provides proofs and formal analysis of classes of patterns this algorithm is able to identify in the limit. Unfortunately, the absence of analysis of what classes are learnable by the previously discussed phonotactic learners hinders comparisons of the models.

5.8 *Learning stress patterns*

Stress patterns can be thought of as word-well-formedness conditions, and hence a kind of phonotactic pattern. Since stress typologies are diverse and well established, learning stress patterns has become a popular and challenging proving ground for learning algorithms (CHAPTER 39: STRESS: PHONOTACTIC AND PHONETIC EVIDENCE; CHAPTER 41: THE REPRESENTATION OF WORD STRESS; CHAPTER 44: THE IAMBIC–TROCHAIC LAW; CHAPTER 57: QUANTITY-SENSITIVITY).

Dresher and Kaye (1990) propose a learning model in the Principles and Parameters framework for learning stress patterns. In this framework, a grammar is a vector of parameters. The learner takes as input a list of words, and for each word, sets parameters as determined by checking whether the word consists of particular properties, called cues. Gillis *et al.* (1995) implement the model with interesting discussion regarding what constitutes an appropriate cue. They only provide input words up to four syllables in length, and demonstrate that the learner succeeded in learning 75 percent of the patterns. Related work includes Gibson and Wexler's (1994) Triggering Learning Algorithm (see also Frank and Kapur 1996 and Niyogi 2006: chapter 3 for discussion).

Goldsmith (1994) and Gupta and Touretzky (1994) investigate how quantity-insensitive stress patterns can be learned using dynamic networks. Although the models differ in their specifics – Goldsmith employs a different updating procedure than Gupta and Touretzky, who use a standard perceptron – these methods achieve a certain level of success in learning the patterns for which data is presented.

Tesar and Smolensky (2000) discuss twelve OT constraints which yield a typology of quantity-sensitive stress patterns. The OT constraints make reference to feet (CHAPTER 40: THE FOOT), which are not part of the learning input. Consequently, another procedure is necessary to parse the learner's input data, so that it can be processed by RCD (the underlying form is assumed to be a string of the right number of unstressed syllables). This procedure is non-trivial, as there may be different parses (i.e. foot assignments) for a given stress pattern. Tesar (1998a) proposes a procedure called *robust interpretive parsing*. To test their system, Tesar and Smolensky hand-selected a test set consisting of 124 languages containing most of the "familiar metrical phenomena" analyzable with their constraints (Tesar and Smolensky 2000: 68). Note, however, that they acknowledge this set is not necessarily representative of the whole typology generated by their constraints. Using robust interpretive parsing, they report that if the initial state of the learner is monostratal – that is, no a priori ranking – then the learner succeeds on about 60 percent of the languages in the test set. When a particular initial constraint hierarchy is adopted, the learner achieves ~97 percent success. So in this case, robust interpretive parsing (mostly) addresses the problem RCD has with hidden structure (for this particular set of test data).

Heinz (2007, 2009) proposes that all phonotactic patterns are neighborhood-distinct, which is a locality condition defined in automata-theoretic terms. It is shown that all but two of 109 descriptions of the world's stress patterns are neighborhood-distinct and that a particular learner that uses this property can learn 100 of these 109 patterns exactly. Although not every pattern can be learned, the patterns acquired in the "failure" cases differ only slightly from the target patterns. Heinz concludes that this particular notion of locality structures the hypothesis space in a way that makes a significant contribution to phonotactic learning.

6 Conclusions

We have argued that learning theory affirms the role of structure as a solution to the problem of generalization, and that there are ideas and methods within learning theory that allow one to measure this structure and the class of languages which have such structure. These tools offer phonologists a way to characterize the contribution various structural properties of phonological patterns can make to learning.

With the exception of a substantial amount of work on learning in Optimality Theory (and Heinz 2010, on phonotactics), it is striking that most proposed learning algorithms have been evaluated only with case studies. Though such studies are suggestive and can be vital in the development of models, in order to know whether a given case study illustrates general properties of a problem we need analytical results that show *why* the algorithm succeeds, what properties of the training sample are critical to success, and how the algorithm maps experience to grammars.

Finally, we have emphasized what we believe to be the most fruitful direction for future research. Phonologists ought to identify properties of phonological patterns that structure the hypothesis space or reduce its size (cf. Heinz 2009; Tesar, forthcoming). This approach works in tandem with, rather than in lieu of, formal analysis.

REFERENCES

Albright, Adam. 2009. Feature-based generalisation as a source of gradient acceptability. *Phonology* 26. 9–41.

Albright, Adam & Bruce Hayes. 2002. In Mike Maxwell (ed.) *Proceedings of the 6th Meeting of the ACL Special Interest Group on Computational Phonology,* 58–69. Philadelphia: Association for Computational Linguistics.

Albright, Adam & Bruce Hayes. 2003. Rules vs. analogy in English past tenses: A computational/experimental study. *Cognition* 90. 119–161.

Angluin, Dana. 1980. Inductive inference of formal languages from positive data. *Information Control* 45. 117–135.

Angluin, Dana. 1982. Inference of reversible languages. *Journal for the Association of Computing Machinery* 29. 741–765.

Angluin, Dana. 1988a. Queries and concept learning. *Machine Learning* 2. 319–342.

Angluin, Dana. 1988b. Identifying languages from stochastic examples. Unpublished ms., Yale University.

Anthony, Martin & Norman Biggs. 1992. *Computational learning theory.* Cambridge: Cambridge University Press.

Bane, Max, Jason Riggle & Morgan Sonderegger. 2010. The VC dimension of constraint-based grammars. *Lingua* 120. 1194–1208.

Barton, G. Edward. 1986. Computational complexity in two-level morphology. *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics,* 53–59. Available (July 2010) at http://aclweb.org/anthology-new/P/P86.

Barton, G. Edward, Robert C. Berwick & Eric Sven Ristad. 1987. *Computational complexity and natural language.* Cambridge, MA: MIT Press.

Berko, Jean. 1958. The child's learning of English morphology. *Word* 14. 150–177.

Blumer, Anselm, Andrzej Ehrenfeucht, David Haussler & Manfred K. Warmuth. 1989. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM* 36. 929–965.

Chomsky, Noam. 1956. Three models for the description of language. *IRE Transactions on Information Theory.* 113–124.

Clark, Alexander & Rémi Eyraud. 2007. Polynomial identification in the limit of substitutable context-free languages. *Journal of Machine Learning Research* 8. 1725–1745.

CMU. 1993. *The Carnegie Mellon University Pronouncing Dictionary* v0.1. Available (July 2010) at www.speech.cs.cmu.edu/cgi-bin/cmudict.

Coetzee, Andries W. 2008. Grammaticality and ungrammaticality in phonology. *Language* 84. 218–257.

Coleman, John & Janet B. Pierrehumbert. 1997. Stochastic phonological grammars and acceptability. In John Coleman (ed.) *Proceedings of the 3rd Meeting of the ACL Special Interest Group in Computational Phonology (SIGPHON* 3), 49–56. Somerset, NJ: Association for Computational Linguistics.

Dresher, B. Elan & Jonathan Kaye. 1990. A computational learning model for metrical phonology. *Cognition* 34. 137–195.

Edlefsen, Matt, Dylan Leeman, Nathan Meyers, Nathaniel Smith, Molly Visscher & David Wellcome. 2008. Deciding Strictly Local (SL) languages. In *Proceedings of the Midstates Conference for Undergraduate Research in Computer Science and Mathematics* 6, 66–75.

Eisner, Jason. 1997a. Efficient generation in primitive Optimality Theory. In *Proceedings of the 35th Annual Meeting of the ACL and 8th Conference of the European Chapter of the Association for Computational Linguistics*, 313–320. Morristown, NJ: Association for Computational Linguistics.

Eisner, Jason. 1997b. What constraints should OT allow? Handout from paper presented at the 71st Annual Meeting of the Linguistic Society of America, Chicago (ROA-204).

Frank, Robert & Shyam Kapur. 1996. On the use of triggers in parameter setting. *Linguistic Inquiry* 27. 623–660.

Gibson, Edward & Kenneth Wexler. 1994. Triggers. *Linguistic Inquiry* 25. 407–454.

Gildea, Daniel & Daniel Jurafsky. 1996. Learning bias and phonological-rule induction. *Computational Linguistics* 22. 497–530.

Gillis, Steven, Gert Durieux & Walter Daelemans. 1995. A computational model of P&P: Dresher & Kaye (1990) revisited. In Maaike Verrips & Frank Wijnen (eds.) *Approaches to parameter setting*, 135–173. Amsterdam: University of Amsterdam.

Gold, E. M. 1967. Language identification in the limit. *Information and Control* 10. 447–474.

Gold, E. M. 1978. Complexity of automata identification from given data. *Information and Control* 37. 302–320.

Goldsmith, John A. 1990. *Autosegmental and metrical phonology*. Oxford & Cambridge, MA: Blackwell.

Goldsmith, John A. 1991. Phonology as an intelligent system. In Donna Jo Napoli & Judy Kegl (eds.) *Bridges between psychology and linguistics: A Swarthmore Festschrift for Lila Gleitman*, 247–267. Hillsdale, NJ: Lawrence Erlbaum.

Goldsmith, John A. 1993a. Harmonic phonology. In Goldsmith (1993b), 221–269.

Goldsmith, John A. 1993b. *The last phonological rule: Reflections on constraints and derivations*. Chicago: University of Chicago Press.

Goldsmith, John A. 1994. A dynamic computational theory of accent systems. In Jennifer Cole & Charles W. Kisseberth (eds.) *Perspectives in phonology*, 1–28. Stanford: CSLI.

Goldsmith, John A. & Jason Riggle. Forthcoming. Information theoretic approaches to phonological structure: The case of Finnish vowel harmony. *Natural Language and Linguistic Theory*.

Goldsmith, John A. & Aris Xanthos. 2009. Learning phonological categories. *Language* 85. 4–38.

Goldwater, Sharon & Mark Johnson. 2003. Learning OT constraint rankings using a Maximum Entropy model. In Jennifer Spenador, Anders Eriksson & Östen Dahl (eds.) *Proceedings of the Stockholm Workshop on Variation within Optimality Theory*, 111–120. Stockholm: Stockholm University.

Gordon, Matthew. 2002. A factorial typology of quantity-insensitive stress. *Natural Language and Linguistic Theory* 20. 491–552.

Graf, Thomas. 2010. Comparing incomparable frameworks: A model theoretic approach to phonology. *University of Pennsylvania Working Papers in Linguistics* 16(1). Available at http://repository.upenn.edu/pwp/vol16/iss1.

Greenberg, Joseph H. 1963. Some universals of grammar, with particular reference to the order of meaningful elements. In Joseph H. Greenberg (ed.) *Universals of language*, 73–113. Cambridge, MA: MIT Press.

Greenberg, Joseph H. 1978. Initial and final consonant sequences. In Joseph H. Greenberg, Charles A. Ferguson & Edith A. Moravcsik (eds.) *Universals of human language*, vol. 2: *Phonology*, 243–279. Stanford: Stanford University Press.

Griffiths, Thomas L., Charles Kemp & Joshua B. Tenenbaum. 2008. Bayesian models of cognition. In Ron Sun (ed.) *The Cambridge handbook of computational cognitive modeling*, 59–100. Cambridge: Cambridge University Press.

Gupta, Prahlad & David Touretzky. 1994. Connectionist models and linguistic theory: Investigations of stress systems in language. *Cognitive Science* 18. 1–50.

Hansson, Gunnar Ólafur. 2001. Theoretical and typological issues in consonant harmony. Ph.D. dissertation, University of California, Berkeley.

Harrison, Michael A. 1978. *Introduction to formal language theory*. Reading: Addison Wesley.

Haussler, David. 1995. Part 1: Overview of the Probably Approximately Correct (PAC) learning framework. Available (July 2010) at http://citeseer.ist.psu.edu/haussler92part.html.

Haussler, David, Michael Kearns & Robert Schapire. 1992. Bounds on the sample complexity of Bayesian learning using information theory and the VC dimension. Technical Report LRC-91-44.

Hayes, Bruce. 1995. *Metrical stress theory: Principles and case studies*. Chicago: University of Chicago Press.

Hayes, Bruce & Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry* 39. 379–440.

Heinz, Jeffrey. 2007. The inductive learning of phonotactic patterns. Ph.D. dissertation, University of California, Los Angeles.

Heinz, Jeffrey. 2008. Learning left-to-right and right-to-left iterative languages. In Alexander Clark, François Coste & Laurent Miclet (eds.) *Grammatical inference: Algorithms and applications*, 84–97. Berlin: Springer.

Heinz, Jeffrey. 2009. On the role of locality in learning stress patterns. *Phonology* 26. 303–351.

Heinz, Jeffrey. 2010. Learning long distance phonotactics. *Linguistic Inquiry* 41. 623–661.

Higuera, Colin de la. 1997. Characteristic sets for polynomial grammatical inference. *Machine Learning* 27. 125–138.

Higuera, Colin de la. 2010. *Grammatical inference: Learning automata and grammars*. Cambridge: Cambridge University Press.

Hopcroft, John E., Rajeev Motwani & Jeffrey D. Ullman. 1979. *Introduction to automata theory, languages, and computation*. Boston: Addison-Wesley.

Horning, J. J. 1969. A study of grammatical inference. Ph.D. dissertation, Stanford University.

Idsardi, William J. 1998. Tiberian Hebrew spirantization and phonological derivations. *Linguistic Inquiry* 29. 37–73.

Inkelas, Sharon & Cheryl Zoll. 2005. *Reduplication: Doubling in morphology*. Cambridge: Cambridge University Press.

Jain, Sanjay, Daniel Osherson, James S. Royer & Arun Sharma. 1999. *Systems that learn: An introduction to learning theory*. 2nd edn. Cambridge, MA: MIT Press.

Johnson, C. Douglas. 1972. *Formal aspects of phonological description*. The Hague & Paris: Mouton.

Johnson, Mark. 1984. A discovery procedure for certain phonological rules. In *Proceedings of the 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics*, 344–347.

Joshi, Aravind K. 1985. Tree-adjoining grammars: How much context sensitivity is required to provide reasonable structural descriptions? In David R. Dowty, Lauri Karttunen & Arnold M. Zwicky (ed.) *Natural language parsing: Psychological, computational, and theoretical perspectives*, 206–250. Cambridge: Cambridge University Press.

Kaplan, Ronald & Martin Kay. 1981. Phonological rules and finite state transducers. Paper presented at the 55th Annual Meeting of the Linguistics Society of America, New York.

Kaplan, Ronald & Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20. 331–378.

Karttunen, Lauri. 1993. Finite-state constraints. In Goldsmith (1993b), 173–194.

Karttunen, Lauri. 1998. The proper treatment of optimality in computational phonology. In *Proceedings of the International Workshop on Finite State Methods in Natural Language Processing*, 1–12. Ankara: Bilkent University.

Kearns, Michael & Umesh Vazirani. 1994. *An introduction to computational learning theory*. Cambridge, MA: MIT Press.

Kearns, Michael, Ming Li, Leonard Pitt & Leslie G. Valiant. 1987. On the learnability of Boolean formulae. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, 285–295. New York: ACM Press.

Kobele, Gregory. 2006. Generating copies: An investigation into structural identity in language and grammar. Ph.D. dissertation, University of California, Los Angeles.

Koskenniemi, Kimmo. 1983. *Two-level morphology: A general computational model for word-form recognition and production*. Helsinki: Department of General Linguistics, University of Helsinki.

Legendre, Géraldine, Yoshiro Miyata & Paul Smolensky. 1990. Harmonic Grammar: A formal multi-level connectionist theory of linguistic well-formedness – theoretical foundations. In *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, 388–395. Mahwah, NJ: Lawrence Erlbaum.

Littlestone, Nick. 1988. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning* 2. 285–318.

Mairal, Ricardo & Juana Gil (eds.) 2006. *Linguistic universals*. Cambridge: Cambridge University Press.

McNaughton, Robert & Seymour A. Papert. 1971. *Counter-free automata*. Cambridge, MA: MIT Press.

Muggleton, Stephen. 1990. *Inductive acquisition of expert knowledge*. Wokingham: Addison-Wesley.

Niyogi, Partha. 2006. *The computational nature of language learning and evolution*. Cambridge, MA: MIT Press.

Nowak, Martin A., Natalia L. Komarova & Partha Niyogi. 2002. Computational and evolutionary aspects of language. *Nature* 417. 611–617.

Oncina, José, Pedro García & Enrique Vidal. 1993. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15. 448–458.

Osherson, Daniel, Scott Weinstein & Michael Stob. 1986. *Systems that learn*. Cambridge, MA: MIT Press.

Papadimitriou, Christos H. & Kenneth Steiglitz. 1998. *Combinatorial optimization: Algorithms and complexity*. Mineola, NY: Dover Publications.

Pater, Joe. 2008. Gradual learning and convergence. *Linguistic Inquiry* 39. 334–345.

Pater, Joe. 2009. Weighted constraints in generative linguistics. *Cognitive Science* 36. 999–1035.

Pinker, Steven. 1979. Formal models of language learning. *Cognition* 7. 217–283.

Pitt, Leonard. 1989. Inductive inference, DFAs and computational complexity. In *Proceedings of the International Workshop on Analogical and Inductive Inference*, 18–44. Heidelberg: Springer.

Popper, Karl. 1959. *The logic of scientific discovery*. New York: Basic Books.

Potts, Christopher, Joe Pater, Karin Jesney, Rajesh Bhatt & Michael Becker. 2010. Harmonic Grammar with linear programming: From linear systems to linguistic typology. *Phonology* 27. 77–117.

Prince, Alan & Paul Smolensky. 1993. Optimality Theory: Constraint interaction in generative grammar. Unpublished ms., Rutgers University & University of Colorado, Boulder. Published 2004, Malden, MA & Oxford: Blackwell.

Riggle, Jason. 2004. Generation, recognition, and learning in finite state Optimality Theory. Ph.D. dissertation, University of California, Los Angeles.

Riggle, Jason. 2009. The complexity of ranking hypotheses in Optimality Theory. *Computational Linguistics* 35. 47–59.

Roark, Brian & Richard Sproat. 2007. *Computational approaches to morphology and syntax.* Oxford: Oxford University Press.

Rogers, Hartley. 1967. *Theory of recursive functions and effective computability.* New York: McGraw Hill.

Rogers, James & Geoffrey K. Pullum. 2007. Aural pattern recognition experiments and the subregular hierarchy. In Marcus Kracht (ed.) *Proceedings of the 10th Mathematics of Language Conference,* 1–7. Los Angeles: University of California, Los Angeles.

Rose, Sharon & Rachel Walker. 2004. A typology of consonant agreement as correspondence. *Language* 80. 475–531.

Rosenblatt, Frank. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65. 386–408.

Rumelhart, D. E. & J. L. McClelland. 1986. On learning the past tenses of English verbs. In J. L. McClelland, D. E. Rumelhart & the PDP Research Group (eds.) *Parallel distributed processing: Explorations in the microstructure of cognition,* vol. 2, 216–271. Cambridge, MA: MIT Press.

Shieber, Stuart M. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy* 8. 333–343.

Smolensky, Paul & Géraldine Legendre (eds.) 2006. *The harmonic mind: From neural computation to optimality-theoretic grammar.* Cambridge, MA: MIT Press.

Sober, Elliot. 2008. *Evidence and evolution.* Cambridge: Cambridge University Press.

Stabler, Edward P. 2009. Computational models of language universals: Expressiveness, learnability and consequences. In Morten H. Christiansen, Chris Collins & Simon Edelman (eds.) *Language universals,* 200–223. Oxford: Oxford University Press.

Tesar, Bruce. 1995. Computational Optimality Theory. Ph.D. dissertation, University of Colorado, Boulder.

Tesar, Bruce. 1997. Multi-recursive constraint demotion. Unpublished ms., Rutgers University.

Tesar, Bruce. 1998a. An iterative strategy for language learning. *Lingua* 104. 131–145.

Tesar, Bruce. 1998b. Error-driven learning in Optimality Theory via the efficient computation of optimal forms. In Pilar Barbosa, Danny Fox, Paul Hagstrom, Martha McGinnis & David Pesetsky (eds.) *Is the best good enough? Optimality and competition in syntax,* 421–436. Cambridge, MA: MIT Press.

Tesar, Bruce. Forthcoming. Learning phonological grammars for output-driven maps. *Papers from the Annual Meeting of the North East Linguistic Society* 39.

Tesar, Bruce & Paul Smolensky. 1993. The learnability of Optimality Theory: An algorithm and some basic complexity results. Unpublished ms., University of Colorado, Boulder.

Tesar, Bruce & Paul Smolensky. 1996. Learnability in Optimality Theory (long version). Technical Report 96:3, Department of Cognitive Science, Johns Hopkins University.

Tesar, Bruce & Paul Smolensky. 1998. Learnability in Optimality Theory. *Linguistic Inquiry* 29. 229–268.

Tesar, Bruce & Paul Smolensky. 2000. *Learnability in Optimality Theory.* Cambridge, MA: MIT Press.

Thomas, Wolfgang. 1997. *Languages, automata, and logic,* vol. 3. New York: Springer.

Valiant, Leslie G. 1984. A theory of the learnable. *Communications of the ACM* 27. 1134–1142.

Vapnik, Vladimir. 1998. *Statistical learning theory.* New York: Wiley.

Vapnik, Vladimir & Alexey Chervonenkis. 1971. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications* 16. 264–280.

Zuraw, Kie. 2000. Patterned exceptions in phonology. Ph.D. dissertation, University of California, Los Angeles.