Learning Phonological Categories

John Goldsmith, Aris Xanthos

Language, Volume 85, Number 1, March 2009, pp. 4-38 (Article)

Published by Linguistic Society of America
DOI: *https://doi.org/10.1353/lan.0.0100*

LEARNING PHONOLOGICAL CATEGORIES

JOHN GOLDSMITH
University of Chicago

ARIS XANTHOS
University of Lausanne

This article describes in detail several explicit computational methods for approaching such questions in phonology as the vowel/consonant distinction, the nature of vowel harmony systems, and syllable structure, appealing solely to distributional information. Beginning with the vowel/consonant distinction, we consider a method for its discovery by the Russian linguist Boris Sukhotin, and compare it to two newer methods of more general interest, both computational and theoretical, today. The first is based on spectral decomposition of matrices, allowing for dimensionality reduction in a finely controlled way, and the second is based on finding parameters for maximum likelihood in a hidden Markov model. While all three methods work for discovering the fairly robust vowel/consonant distinction, we extend the newer ones to the discovery of vowel harmony, and in the case of the probabilistic model, to the discovery of some aspects of syllable structure.*

Keywords: categorization, machine learning, hidden Markov models, spectral graph theory, vowel harmony, phonology

1. INTRODUCTION. The study of phonological systems has two primary goals: a statement of the generalizations regarding permissible segment sequences and structures, and an analysis of the productive alternations that account for the variant forms of a morpheme occasioned by the phonological content of the larger utterance in which it is found: in short, phonology studies phonotactics and alternations. From a historical point of view, pregenerative American phonology focused on questions of phonotactics, lacking the tools to treat alternations in depth, and generative phonology (and postgenerative phonology) has focused on alternations, lacking the tools to deal with a detailed study of phonotactics.[1]

In this article, we approach the general problem of inference (or acquisition) of phonotactics, and consider the usefulness of three algorithmic styles of analysis to three questions about the overall phonotactics and the phonological categories that phonotactics presuppose (consonants, vowels, etc.). These questions are: (i) Given a sample of data (transcribed symbolically) from a language, can we infer which segments are vowels and which are consonants? (ii) Can we infer on the basis of such data whether the language in question possesses a system of vowel harmony, and if so, what the patterns of vowel harmony are in the language? (iii) Can we draw inferences about the organization of segments into syllabic structure?

We have chosen these closely related questions because they seem to us to be unavoidable questions for phonology: while not every framework will demand a purely distributional method of answering questions, it is more than likely that these questions will be meaningful within any given phonological framework. And if the first question seems very simple, the fact of the matter is that if we demand a fully explicit and formal algorithm to identify vowels and consonants, it turns out (as we have learned) not to be all that easy. Be that as it may, the task of discovering vowel harmony and

* This work was supported in part by a grant from the Swiss National Science Foundation to the second author during his stay at the University of Chicago. We are grateful to a number of our colleagues for discussion of these topics, including François Bavaud, Yu Hu, Remi Jolivet, and Jason Riggle.

[1] A frank and probing survey of linguists working today would no doubt reveal a wide range of opinion regarding the relative importance of these two poles. We believe both are important, but focus on the area of phonotactics in this article.

syllable structure automatically would doubtless strike any working phonologist as a nontrivial task—a highly nontrivial task.

It is not our goal here to engage in an ideological battle, but it would serve no purpose to ignore the simple fact that the approach that we have taken, and described, here stands in stark contrast with much generative work on phonology. The goal is NOT first and foremost to develop a cognitive model of how humans use language; it is, rather, to build a (scientific) model of language, as we know it through our observations of it; and part of the scientific character of the work is the formal development of explicit methods of analyzing data. Perhaps the best way to put it is that we wish to pour our scientific creativity into developing methods for linguistic analysis, rather than into the development of an analysis of any one particular set of data.

We explore three quite different, fully automatic algorithms that address one or more of these questions. We look at the first for purely historical reasons—because it was one of the first algorithms proposed to solve a phonological problem. The other two approaches we explore are based on methods that are both powerful and promising, and are in wide use in the machine-learning community. One is based on eigenvector decomposition and is closely related to such methods as principal component analysis and latent semantic indexing, while the other is based on maximum-likelihood calculations and the application of hidden Markov models. The three methods are these:

(i) The first, due to Sukhotin (1962), is one of the earliest algorithms that we are aware of whose goal is to automatically infer which segments are vowels and which are consonants; while we have implemented it computationally, it is simple enough that it can be applied by hand, which was undoubtedly what motivated its discoverer. We apply the method to a number of phonologically different languages in §2 below.

(ii) The second system is based on spectral graph theory, a relatively new mathematical field that has been applied to a wide variety of both theoretical and practical problems; it can be employed to reduce observational data, which can be thought of as residing in a space of a large number of dimensions, to a greatly simplified representation in a small number of dimensions (Chung 1997). In our case, this operation makes the resulting structure accessible to phonologists, when the dimension turns out to be, for example, a sonority dimension along which vowels and consonants are scattered appropriately. We describe the method in detail, in part because of its unfamiliarity to linguists, and in part because it allows one to compute one-dimensional renderings of data easily on the basis of similarity relationships that would otherwise seem to be quite difficult to collapse in such a way; this method is likely to be of interest to linguistics for other purposes as well (as in Belkin & Goldsmith 2002, for example).

(iii) The third system employs hidden Markov models (HMMs) in order to automatically develop a probabilistic model of the data. We show that constraining the system to learn the probabilistic parameters that maximize the probability of the data leads the systems to infer categories of segments that are in some ways remarkably like traditional phonological divisions of sounds into major categories, but the system consistently infers a syllable structure that is in some ways at odds with traditional analysis; the very same model is also capable of discovering the presence of vowel harmony in data from Finnish.

All of the algorithms that we explore and evaluate in this article fall into the class of what would today be called ‘unsupervised language learning’ (or grammar induction); that is, they are designed to be neutral with respect to the language that they analyze (neutral in the sense that they have no prior knowledge of the structure or lexicon specific to any language) and be capable of taking data from any language as input

and producing an analysis (as its output) that gives an accurate description of the language that generated the input data.

The work we report here was originally motivated by our work in unsupervised learning of natural language morphology (Goldsmith 2001, Xanthos 2008) based on distributional and information-theoretic concepts. In order to develop a language-independent algorithm capable of learning Arabic morphology strictly from the data, it was necessary to consider whether such information-theoretic considerations were sufficient to allow the learning device to 'realize' that there was a natural division of segments into the categories that we have since Classic times called 'vowels' and 'consonants'. Thus the importance of the question arose in the first instance in an essentially practical context, but we have pursued the question beyond the original needs of the morphology learner.

Looking ahead, what we demonstrate is that the first method, Sukhotin's, works relatively well, though it does not extend easily to other problems besides the one it was designed to deal with: distinguishing vowels from consonants. In addition, however, we find that its performance is relatively sensitive to the encoding scheme used, and under some conditions it can perform quite poorly. The spectral method of analyzing similarities does a relatively good job of distinguishing vowels and consonants, though it is not perfect; it does quite nicely for the analysis of vowel harmony, but does not extend naturally to the treatment of syllable structure. Maximum-likelihood analysis on a finite-state automaton (i.e. hidden Markov model) works remarkably well in detecting the vowel/consonant distinction and the vowel harmony system of Finnish, and sheds some interesting light on the sonority hierarchy and syllable structure of French and English.

2. PRIOR SCHOLARSHIP. There has been a certain amount of work along these lines, but most of it is not well known at the present time. The first generation includes the pregenerative work, such as that by Eli Fischer-Jørgensen and Fred Householder, that is methodologically aligned with the view, widely held in the 1950s, that one of the primary goals of linguistic theory is to develop rigorous, purely formal methods for arriving at an analysis of a set of data; this work was almost entirely done without access to computers (see Goldsmith & Xanthos 2008, appendix A, for further discussion).

A second generation of work on distributional classification of phonological segments grew out of computational linguistics, from researchers using tools from mathematics and computer science, and thus was done with full awareness of the growth of knowledge of methods for data-driven classification—and also of the real complexity of the problem. That is, even for the simple case of classifying segments into two subgroups (vowels and consonants), there are $2^{n-1} - 1$ ways to do this, which means that even a modest inventory of thirty phonemes can be divided into two categories in more than 500 million ways. Clearly, it will not suffice to have a quantitative method that will EVALUATE the goodness of any given classification; it would take too long to evaluate each possible division. We are back to the fundamental problem of linguistic analysis, which is to find a means of avoiding a search through all conceivable analyses. In hindsight, it is interesting to reread the structuralists' accounts, because they never seemed to be aware of how difficult the problem is, nor of the degree to which their analysis appears (in retrospect) to be guided by their implicit knowledge of the phonetics.

The period since the late 1950s has seen the development of statistical methods for classification and categorization based on iterative aggregation (see, in particular, Ward 1963). These are 'bottom-up' methods par excellence: the algorithm begins by assuming that all of the elements being considered form distinct classes, each with one member.

At each iteration, the pair of classes that are MOST similar (by some criterion) are collapsed into a single class, and this continues until only one class, containing all the elements, remains. In general, then, such methods do not DETERMINE how many classes are present in the data; but given a measure of similarity and a decision as to how many categories one 'wants', so to speak, such methods may succeed well in finding useful categorizations. In the case we are interested in, it is natural to define 'similarity' on the basis of similar distribution. Powers (1997) reviews and compares quantitatively an impressive number of approaches to this problem based on work done in the 1990s (see notably Powers 1991, Finch 1993, Schifferdecker 1994). He considers in detail the effect of different assumptions about how to measure similarity (or dissimilarity) between two contexts (contexts are typically represented as vectors in a space of dimension $2(n + 1)$, where n is the number of phonemes in the language, and in which each dimension represents the number of occurrences of a phoneme or boundary, to the left or to the right). Powers also considers the impact of different assumptions about how to convert the similarity between two context vectors, on the one hand, into a measure of similarity between two disjoint SETS of elements (in this case, of phonemes), on the other. Perhaps the most significant problem encountered in these bottom-up approaches is that although typically one of the categories discovered by such systems does indeed include the set of all vowels, it is not always the case that in the penultimate iteration of the algorithm—the point at which there are exactly two categories—one of the classes is the vowels, and the other the consonants.

Ellison explored the usefulness of MINIMUM DESCRIPTION LENGTH analysis (henceforth, MDL analysis) for the problem of distinguishing classes of phonological segments (see Ellison 1991, 1994, and Rissanen 1989 for the general framework). One of the goals of MDL analysis is to use information-theoretic concepts in order to determine the correct granularity appropriate for analyzing a collection of data. In its simplest form, MDL analysis calls our attention to the fact that the two extremes of categorization—putting every element into a singleton category, and putting every element in the same category—are both of little or no value; the first overfits the data, and the second underfits. MDL offers a way to measure the complexity of a set of categories and the success with which such a set of categories models the observed data, and it offers an OBJECTIVE FUNCTION (that is, a function whose value we attempt to optimize) combining these two expressions, which should be minimized in order to find the best analysis of the data. In order to achieve this, it is necessary to establish a method that extracts the regularities in the data in a lossless way, in such a way that we can measure the information in the data that is NOT in the regularities, and a method to MEASURE quantitatively both the model that extracts the regularities and the size of the data after the regularities have been extracted. In more concrete terms, then, Ellison's MDL-style analysis consists of three components: the specification of a set of models with these properties, evaluation metrics of the sort just mentioned, and a search algorithm for FINDING the analysis for a given corpus that optimizes the MDL evaluation metrics. Ellison employs simulated annealing, a statistical process according to which the search algorithm hops about in a fashion that is almost completely random at the beginning, but that increasingly hops only when the change leads to an immediate improvement in the value of the evaluation metric, eventually stopping because no change can be found that favors an increase in the evaluation metric (meaning that an optimum—and hopefully a global optimum—for parametric values has been found). Ellison reports excellent results for his method.

The present work seeks to address the challenges of unsupervised learning of phonology in a relatively theory-neutral way, in part to see just how few assumptions can be

made without impeding our ability to infer structural patterns from the linguistic data. We see our work as part of a larger project of understanding linguistic analysis from a Bayesian perspective: crudely put, seeing whether linguistic theory can be construed as a particular form of statistical learning without abandoning any of the established results concerning linguistic structure in the description of particular languages—and if that is possible, how is that reconceptualization to be accomplished. A number of researchers have been developing perspectives along these lines, sometimes unknown to one other, over the last fifteen years, in publications such as Ellison 1991, 2001, Powers 1997, Goldsmith 2001, Goldsmith & O'Brien 2006, Goldwater 2006, Dowman 2008, as well as others cited therein.[2]

3. VOWELS AND CONSONANTS. In this section, we describe and evaluate three approaches to the problem of identifying the two classes, that of vowels and that of consonants, in a distributional way: an approach described by Boris Sukhotin (1962), a method based on spectral decomposition of matrices encoding segment transition information, and a maximum-likelihood method that employs hidden Markov models, or HMMs. This order of presentation corresponds to increasing ability to correctly model the data.

Needless to say, we recognize that there are phonetic differences between what Pike (1943) called vocoids and contoids, but as he was at pains to point out, the phonetic distinction is related to and yet independent of the linguist's notion of vowel/consonant. The question of the relationship of the phonetic character of vocoids and the category of vowels is much like the question of determining what is a verb and what a noun in a crosslinguistic way: most of the time, and for most languages, the answer is perfectly obvious, but that does not mean that to understand the problem deeply and to have an answer that works even for the difficult cases is easy. Semantics provides a way to identify what is a noun and what a verb 95 percent of the time, but the linguist cares more about the more difficult cases where the traditional distributional considerations come to the fore in making the decision. So too in phonology: phonetics suffices 95 percent of the time, but linguists have known since William Dwight Whitney (see n. 17) that phonetics is not enough, and that phonological distribution is both critical and criterial.

3.1. SUKHOTIN'S ALGORITHM. To the best of our knowledge, Sukhotin was the first to propose a truly algorithmic and language-independent solution to the problem of identifying vowels and consonants on the basis of a symbolic transcription (Sukhotin 1962, 1973).[3] His method is also conceptually and computationally much simpler than the other approaches investigated here and provides a good opportunity to introduce a few basic notations. It relies on two fundamental assumptions: first, that the most frequent symbol in a transcription is always a vowel, and second, that vowels and

[2] Regrettably, we were not familiar with the work by Powers and Ellison before the work described here was undertaken, and we offer the reader a broader than usual review of the previous literature in part because so much of it is rarely cited today. See also Peperkamp et al. 2006 for a closely related perspective, and citations involving the use of statistical models in the psycholinguistic acquisition literature, where Saffran et al. 1996 has had a major impact. There is a growing community of researchers who have recently approached phonological problems with computational tools and concepts, and many of them share the basic perspectives of this article; this community includes Sharon Goldwater (Goldwater 2006), Bruce Tesar (Tesar 1998), Bruce Hayes, Colin Wilson (Hayes & Wilson 2008), and Jason Riggle; one of us (JG) has pursued in detail with Jason Riggle the treatment of vowel harmony that is sketched in this article (Goldsmith & Riggle 2007).

[3] We thank Remi Jolivet for drawing our attention to this work. We have also profited from the analysis of Sukhotin's algorithm given by Guy (1991).

consonants tend to alternate more often than not. Starting from the first assumption, Sukhotin's algorithm attempts to divide the phonemes of a language into two classes that satisfy the second assumption.

Consider a language with an inventory of n phonemes $P := \{p_1, \ldots, p_n\}$, and suppose we have a sample from this language (a sample from P^*) called C. We define the function *Count*(.) as specifying the number of times its argument is found in the relevant corpus C; thus *Count*(ba) specifies the number of times the sequence of phonemes /ba/ occurs in the corpus. We may construct a table where each row and each column corresponds to a phoneme, and each cell stores the number of times that the corresponding phonemes occurred next to one another (irrespective of their order). More specifically, we build a square MATRIX R, of dimensions $(n \times n)$, where the cell at the intersection of the i-th row and the j-th column is defined as $r_{ij} := Count(p_ip_j) + Count(p_jp_i)$. R is thus a symmetric matrix; that is, the i-th row is identical to the i-th column, or equivalently $r_{ij} = r_{ji}$. The elements on the main diagonal should be equal to twice the number of times that each phoneme occurs next to itself, but Sukhotin's convention is to ignore these values by setting them to zero ($r_{ii} := 0$).

For instance, given the sample corpus described in Appendix A, we find an inventory of five phonemes $P = \{b, n, s, a, i\}$, so $n = 5$. Using the frequencies of sequences of two phonemes reported in Table A1,[4] we may calculate the components of R as indicated: $r_{11} = 0$ by convention, $r_{12} = Count(bn) + Count(nb) = 0, \ldots, r_{15} = Count(bi) + Count(ib) = 3$, and so on. We obtain the (5×5) matrix given in 1.

(1)
$$R = \begin{array}{c} \\ b \\ n \\ s \\ a \\ i \end{array} \begin{array}{c} \begin{array}{ccccc} b & n & s & a & i \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 4 & 3 \\ 0 & 0 & 2 & 7 & 3 \\ 0 & 2 & 0 & 2 & 2 \\ 4 & 7 & 2 & 0 & 0 \\ 3 & 3 & 2 & 0 & 0 \end{pmatrix} \end{array}$$

Sukhotin's algorithm begins by labeling all phonemes as consonants. Then it enters an iterative phase: during each cycle, it uses the information contained in R to assign to each tentative consonant a score that represents the likelihood that it actually IS a vowel; the single most likely candidate at that point is labeled as a vowel, and then removed from any further calculations, and in effect from the matrix. This process is repeated until no more consonants are likely to change category, and those that are left are the consonants. The algorithm can then return the entire list of phonemes, with each one labeled as vowel or consonant.

At the core of this approach lies the score $v(p_i)$ that is iteratively assigned to each phoneme p_i. Based on the assumption that consonants and vowels are classes that tend to alternate, a candidate for vowelhood is expected to occur more frequently next to a consonant than next to a vowel; thus, the DIFFERENCE between its frequency next to a consonant and its frequency next to a vowel should be positive: the larger, the better. This difference is precisely the score $v(p_i)$ assigned by Sukhotin's algorithm.

When we apply Sukhotin's algorithm to natural language corpora, we find that its accuracy is highly dependent on the particular set of data being processed. We have run experiments on three large lists of words in English, French, and Finnish.[5] The

[4] Sequences involving a word boundary are not used in this case.

[5] These corpora can be downloaded from http://hum.uchicago.edu/~jagoldsm/Papers/GoldsmithXanthos/GoldsmithXanthos.htm, along with relevant software.

English and French corpora were phonetic transcriptions,[6] whereas the Finnish corpus was orthographically transcribed (written Finnish is notoriously close to a phonetic transcription). Basic facts about these corpora are summarized in Table 1.[7]

	# WORDS		# PHONES	
CORPUS	TYPES	TOKENS	TYPES	TOKENS
English	38,466	58,156	54	386,421
French	15,864	21,768	36	147,146
Finnish	32,156	44,040	27	466,134

TABLE 1. Basic facts about the corpora.

Table 2 shows the classification of vowels and consonants performed by the algorithm on each corpus. For French and Finnish, the results are good though not perfect. In the French corpus, the most frequent phoneme turns out to be /ʁ/, so it is misclassified as a vowel in the first place. This does not affect the classification of the remaining phonemes, however, all of which are correctly labeled. In Finnish, all consonants and vowels are correctly identified, with the exception of the rare symbol <q>, whose role is not entirely unlike that which it has in English: it occurs primarily in borrowings, and is pronounced [k] or [kv]. A closer look at the contexts where it occurs confirms that, with regard to the criterion underlying this approach, this symbol clearly behaves more like a vowel than a consonant: it follows a consonant in fifteen out of eighteen occurrences in noninitial position; similarly, it is followed by a consonant in eleven out of sixteen occurrences in nonfinal position (this consonant is systematically <v>). Notice also that the items listed in Table 2 are arranged by decreasing order of typicality: the most vowel-like symbols are at top of the vowels column, and the less vowel-like symbols are at top of the consonants column;[8] thus, the misclassification of <q> in Finnish may also be viewed as a problem of threshold—it should have been the most vowel-like consonant, rather than the other way around.

The classification obtained for English was quite bad when we used the transcriptions for vowels that were present in the file. In particular, half of the phonemes labeled as vowels (10/21) are actually consonants, and the proportion of real vowels misclassified as consonants is even higher (20/33). It appears, however, that the primary reason for the poorness of the results lies in the particular method used to represent stress level: there is no connection made between (for example) the vowel /æ/ and the vowel /'æ/; despite the fact that they are qualitatively the same vowel, they are treated by the system as two unrelated segments, and this leads to a representational scheme in which there are many vowels with a much lower frequency. When we remove the stress level from the vowels, we get very different results, results that are much better. In particular, the only divergence from a phonetic classification is that /ɹ/ is misclassified as a vowel.

[6] They are given here in mostly standard IPA. Note that in the English transcription, there is a distinction between stressed and unstressed vowels, the former of which are marked by a prefixed ' symbol. In the French transcription, there is no distinction between /a/ and /ɑ/, and /h/ denotes the *h*-aspiré, which is treated as a phoneme in this case; *h*-aspiré words are phonetically vowel-initial, but behave with respect to phrase-level phonological rules as if they began with a consonant; most such words descended from Germanic *h*-initial borrowings.

[7] Note that the experiments reported here and throughout the article make no attempt to evaluate how the results of a method vary across a range of samples within a single language; more details about this important issue can be found in Xanthos 2008:75–89.

[8] This is where our implementation of the algorithm differs from Sukhotin's: we keep ORDERING phonemes after the zero threshold, so that we can also evaluate their typicality as consonants.

ENGLISH		FRENCH		FINNISH	
CONSONANTS	VOWELS	CONSONANTS	VOWELS	CONSONANTS	VOWELS
t	ʌ	t	ʁ	t	i
k	ɹ	l	a	s	a
d	ɪ	s	i	n	e
p	s	n	e	l	u
z	l	k	ə	k	o
b	n	m	o	m	ä
ˈɛ	ɝ	d	ɑ̃	r	y
ˈɪ	m	p	ɛ	v	ö
g	w	b	y	p	ɋ
i	h	v	ɔ̃	h	
v	ˈeɪ	z	ɛ̃	j	
ʃ	j	f	ɔ	d	
ˈɔ	ˈɝ	ʒ	u	b	
ˈæ	ʊ	g	ø	g	
ˈa	ɑɪ	j	œ	f	
ŋ	eɪ	ʃ	œ̃	c	
ˈi	tʃ	ɲ		w	
ˈɑɪ	f	w		x	
ˈu	ð	h			
dʒ	ɑʊ	ɥ			
ˈʌ	oɪ				
ɑ					
ˈoʊ					
ɛ					
ˈɑʊ					
u					
ɔ					
æ					
θ					
ˈʊ					
ˈoɪ					
ʊ					
ʒ					

TABLE 2. Results of Sukhotin's algorithm on three natural language corpora.

Similarly to /ʁ/ in French, /ɹ/ is one of the most frequent consonants in this corpus; /n/ and /t/ are more frequent, but once the two first vowels (/ʌ/ and /ɪ/) have been identified, and their cooccurrences next to other phonemes have been subtracted, the phoneme with the highest score is /ɹ/.

On the whole, these results suggest that Sukhotin's algorithm has two main weaknesses, both of which are related to the overall frequency of phonemes. On the one hand, the classification of low-frequency phonemes tends to be unreliable, because of the insufficient diversity of their contexts (though it shares this weakness to some extent with any data-driven method). On the other hand, the algorithm suffers from the fact that its first decisions are based on no or little more information than the overall frequency of phonemes; this means that there is a risk for high-frequency consonants to be misclassified as vowels. In the case of our English corpus, the systematic splitting of each vowel into a stressed and an unstressed phoneme seems to create a situation where both flaws are exacerbated, hence the generalized collapse of the results.

3.2. SPECTRAL CLUSTERING. Spectral clustering is a relatively recent application of well-known principles of matrix algebra to the particular matrices that are used to describe graphs. In this section, we show how it applies to the phonological task of

identifying vowels and consonants. We first review the basics of graph theory, and then address the specific issue of graph partitioning, that is, dividing the nodes of a graph up into natural groupings—where the 'naturalness' emerges in each case directly out of the strengths of graph weights, which indicate similarity, in a sense that we make explicit below. With this by way of background, we show how this method can be used to successfully infer major phonological categories in the three corpora we described above.

GRAPH THEORY. The term GRAPH is a technical term, and it is defined as a set V of NODES (also called VERTICES), and a set E of EDGES that are said to JOIN or CONNECT pairs of nodes (see e.g. Biggs 1993, Chung 1997). In the graphs that we consider, the edges do not have an inherent direction; they simply join nodes, and so we say that the graphs are UNDIRECTED. However, the edges of our graphs are WEIGHTED, which means that they are associated with a real number; such a weight must be nonnegative. Intuitively, the weight of an edge specifies the strength of the connection between two nodes; a zero weight corresponds to the complete absence of connection. Figure 1 gives an example of such a graph. It has $n = 5$ nodes $V = \{\text{b, n, s, a, i}\}$ with weighted edges.[9]

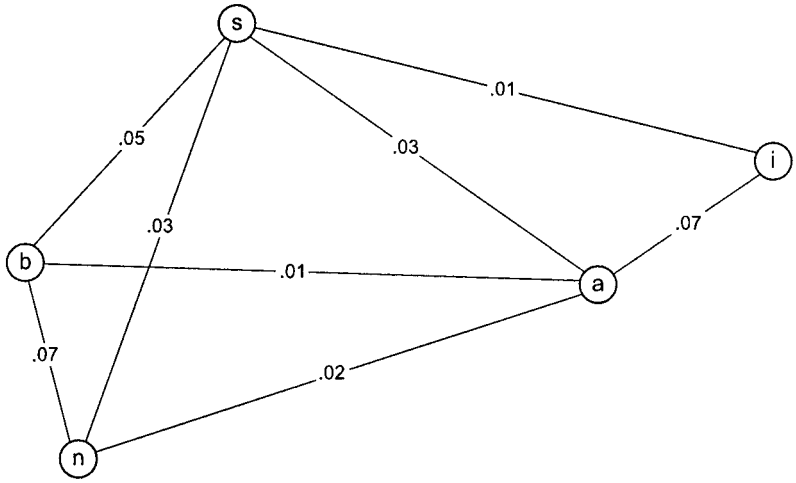

FIGURE 1. A sample weighted undirected graph.

Graphs are commonly represented by a matrix, called an ADJACENCY matrix. If the graph G has n nodes, then its adjacency matrix is an $(n \times n)$ symmetric matrix A where each row and each column corresponds to a node, and the cell a_{ij} at the intersection of the i-th row and j-th column stores the weight of the edge connecting nodes i and j (with 0 if they are not connected). The adjacency matrix of the graph represented in Fig. 1 is given in 2.[10]

[9] Notice that, in this figure, nodes that are strongly connected are less distant than those that have a weaker or no connection; this convention intuitively supports the interpretation of weights as measures of SIMILARITY.

[10] The elements on the main diagonal represent LOOPS, that is, edges connecting a node to itself; for the sake of readability, these were not represented in Fig. 1.

(2)
$$A = \begin{array}{c} \\ \text{b} \\ \text{n} \\ \text{s} \\ \text{a} \\ \text{i} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{b} & \text{n} & \text{s} & \text{a} & \text{i} \end{array} \\ \begin{pmatrix} .09 & .07 & .05 & .01 & 0 \\ .07 & .11 & .03 & .02 & 0 \\ .05 & .03 & .06 & .03 & .01 \\ .01 & .02 & .03 & .13 & .07 \\ 0 & 0 & .01 & .07 & .05 \end{pmatrix} \end{array}$$

The sum of the weights of the edges connecting any given node i to all of its neighbors is called its DEGREE, and we can see that the sum of the i-th row of A is equal to the degree of node i. Using the 'dot notation' according to which placing a dot in the place of a variable is to be construed as a summation over all values, we may define the degree of node i as $d_i := a_{i\bullet}$. This value is a measure of the overall connectivity of i. If we think of the weights on the edges of the graph as characterizing degree of similarity, then the degree of a node represents its total solidarity with the group as a whole. In our example, the degrees of /b/, /n/, /s/, /a/, and /i/ are $d_1 = .09 + .07 + .05 + .01 = .22$, $d_2 = .23$, $d_3 = .18$, $d_4 = .26$, and $d_5 = .13$ respectively. The VOLUME of a graph is a measure of its total connectivity. It is defined as the sum of the degrees of its nodes, or equivalently as the sum of ALL cells of A: $vol(G) := d_{\bullet} = a_{\bullet\bullet}$. In our example, it is equal to 1.02.

GRAPH PARTITIONING. We build a graph below in which each node corresponds to a phoneme, and the weights of the connections between the nodes represent distributional similarity. We would like to employ methods and techniques from graph theory that enable us to automatically find optimal ways to divide the set of nodes of a graph into two or more subgroups on the basis of the weights of the edges. Our goal is to find phonological categories among the phonemes in this way. In order to simplify our discussion, we assume henceforth that all of our graphs are CONNECTED, which means that in effect, there are no islands in our graphs: it is always possible to find a path from any node in a graph to any other node, following edges of the graph.

Partitioning a graph G consists in dividing its nodes into two disjoint subsets S and T. We have assumed that our graph is connected, and therefore partitioning it involves cutting at least one edge. Since the weights of G's edges represent the similarities between the nodes, and since we ultimately are looking for a way of partitioning the nodes of our graph into reasonable groupings, it follows that a natural criterion for choosing among the ways of dividing n nodes into two groups (and there are $2^{n-1} - 1$ different ways!) is to preserve the largest possible amount of connectivity. Intuitively, we can imagine creating a partition by drawing a line on the page in such a way that all of the nodes in S are on one side of the line, and all of the nodes in T are on the other side. Viewed in this way, it is clear that our goal must be to find a line that cuts through as small a number of edges as possible, and the edges that it does cut should have as small a weight as possible. Formally, this means defining the sets S and T in a way that minimizes the resulting CUT, that is, the sum of the weights of edges connecting nodes BETWEEN the two groups, as in 3.

(3) $$cut(S,T) := \sum_{i \in S} \sum_{j \in T} a_{ij}$$

For the graph represented in Fig. 1, this criterion leads to the partition $S = \{\text{b, n, s}\}$, $T = \{\text{a, i}\}$, whose cut is minimal and equal to $.01 + .02 + .03 + .01 = .07$ (see Table 3).

S	T	$cut(S, T)$	$\phi(S, T)$
{b, n, s, a}	{i}	.08	.62
{b, n, s, i}	{a}	.13	.5
{b, n, a, i}	{s}	.12	.67
{b, s, a, i}	{n}	.12	.52
{n, s, a, i}	{b}	.13	.59
{b, n, s}	{a, i}	**.07**	**.18**
{b, n, a}	{s, i}	.18	.58
{b, s, a}	{n, i}	.2	.51
{n, s, a}	{b, i}	.21	.6
{b, n, i}	{s, a}	.19	.43
{b, s, i}	{n, a}	.21	.43
{n, s, i}	{b, a}	.24	.5
{b, a, i}	{n, s}	.18	.44
{n, a, i}	{b, s}	.15	.38
{b, n}	{s, a, i}	.11	.24

TABLE 3. Cut and conductance for each partition of the graph plotted in Fig. 1.

Now, it may happen that using this criterion for 'best cut' yields undesirable results. For example, it might be the case in a graph with 100 nodes that one node i was connected to only one other node in the graph, and that the 'best' cut simply snipped node i off from the rest of the graph, when in reality we were more interested in finding a more balanced division of the nodes into two groups. For this reason, it is useful to refine the criterion for 'best cut' by adding the constraint that S and T should be balanced in terms of the total weights of their nodes. Among several ways of doing this, the experiments described below rely on the CONDUCTANCE measure $\phi(S, T)$ proposed in Kannan et al. 2000 (see Goldsmith & Xanthos 2008, appendix C, for more details on this). In our example, this revised criterion leads to the same partition S = {b, n, s}, T = {a, i}, with minimal conductance $\phi(S, T)$ = .18 (see Table 3).

At this point, what we have is a method for evaluating the relative 'quality' of any proposed partitioning of a graph, but no method for quickly finding the best one. Indeed, the number of partitions to evaluate grows exponentially as the number n of nodes in the graph gets larger. Solutions to problems of this sort that involve exhaustive search are generally unacceptable for obvious reasons—they take too long—and this is what motivates the spectral approach to graph partitioning, whose fundamental idea is based on the natural equivalence of undirected graphs, on the one hand, and symmetric matrices, on the other. A symmetric matrix can always be decomposed in a very special way, one that expresses it as a 'rotation' (a rotation being a mapping that leaves a vector's size unchanged), followed by a set of pure expansions or contractions along a specific set of vectors that are all orthogonal to one another, followed by another rotation; this is known as the spectral theorem for symmetric matrices. It is the intermediate set of expansions/contractions that plays the major role here (the directions of pure contraction or expansion are called EIGENVECTORS), because the ranking of these eigenvectors by their degree of contraction also tells us their importance in the reconstruction of the entire graph. It is a well-known result (Chung 1997) that the vector that best summarizes the cut of the graph into two sections is the SECOND EIGENVECTOR (or FIEDLER VECTOR) of the graph. The main property of this vector is to assign a single number to each node in the graph, in such a way that nodes with a strong connection between them, that is, similar phonemes in this case, are assigned similar numbers; in

effect, this allows us to represent phoneme similarity on a single dimension, as shown in Figure 2.[11]

FIGURE 2. Second eigenvector of the graph represented in Fig. 1.

This process obviously involves a loss of information, but it is guaranteed to yield the best possible reproduction of the overall pattern of similarity defined by the edges of the graph—under the constraint that each node must be represented by a single real number. Thus, although the spectral description in Fig. 2 is only an approximate representation of the graph in Fig. 1, it highlights the similarity between nodes /b/ and /n/ on the one hand, and /a/ and /i/ on the other hand, as well as the more central situation of /s/ (though it is clearly closer to the first pair), and it does it in a purely quantitative way, making it unnecessary for a human being to look at the graph and make decisions about what should be close to what.

Spectral clustering relies on these results to narrow drastically the range of partitions to be evaluated. Since the second eigenvector of a graph summarizes the largest possible amount of the graphs's connectivity, it provides a reasonable basis for filtering out irrelevant partitions—without actually calculating their conductance. Thus, a strategy that is commonly adopted is to evaluate only those partitions that result from grouping nodes according to their position on the second eigenvector. In our example, this amounts to four partitions (see Fig. 2): (i) $S = \{b\}$, $T = \{n, s, a, i\}$; (ii) $S = \{b, n\}$, $T = \{s, a, i\}$; (iii) $S = \{b, n, s\}$, $T = \{a, i\}$; and (iv) $S = \{b, n, s, a\}$, $T = \{i\}$. We have seen previously that partition (iii) has minimal conductance; the important point here is that it was indeed 'preselected' by the spectral approach, contrary to the vast majority of less optimal partitions (eleven out of fifteen, in this artificially small case). This illustrates the efficiency of spectral clustering as a way of quickly searching the space of possible partitions of a graph.

APPLICATION TO THE DISCOVERY OF VOWELS AND CONSONANTS. Weighted graphs are well suited for representing a system of discrete units—in our case, phonemes—with connections of variable strength between them. Undirected graphs add the further constraint that the connections be symmetric; similarity is a typical example of a symmetric relation that can be embodied by a connection in such a graph. When spectral clustering is applied to a graph that encodes some form of similarity between phonemes, it results in a partitioning where similar phonemes are grouped together and the size of groups is as balanced as possible. We observe then that the use of a similarity based on the distribution of phonemes leads to a categorization that corresponds well with the distinction between vowels and consonants.

Any real application of this method requires the notion of DISTRIBUTIONAL SIMILARITY to be made precise. In particular, it is necessary to give a full specification of how the

[11] To be precise, the vector represented in Fig. 2 and used for the spectral clustering is the Fiedler vector of the graph after dividing the value associated with each phoneme by the square root of its stationary probability; see Goldsmith & Xanthos 2008, appendix C, and Xanthos 2008:60–65 for more details.

corpus should be processed in order to assign to each pair of phonemes (or equivalently, to each edge of the graph) a numeric value quantifying the similarity between the distribution of these phonemes. We offer an intuitive presentation of the idea here; see Appendix B here and Goldsmith & Xanthos 2008, appendix D, for further details.

In general, we say that two phonemes are DISTRIBUTIONALLY similar if they occur in similar CONTEXTS. The context of an occurrence of a phoneme can be defined as the previous phoneme (as in the experiments reported below), the two previous phonemes, the previous and next phonemes, and so on. A given corpus can then be used to evaluate the number of occurrences of each phoneme in each context—a number that is typically zero for many phoneme-context combinations. Thus, each phoneme may be characterized by a list of numbers corresponding to its frequency in each context, and the distributional similarity between two phonemes can be assessed by comparing the lists of frequencies associated with them. Given a table with the frequency of each phoneme in each context, it is relatively easy to apply a mathematical manipulation in order to derive the adjacency matrix of a weighted undirected graph, where the weight of an edge corresponds to the distributional similarity of the pair of phonemes connected by this edge.

ENGLISH		FRENCH		FINNISH	
CLUSTER 1	CLUSTER 2	CLUSTER 1	CLUSTER 2	CLUSTER 1	CLUSTER 2
u	ŋ	ə	ɲ	ä	x
ˈʊ	n	ɔ̃	z	a	n
ʊ	ʒ	œ	n	o	h
ˈu	ð	ɔ	f	e	r
ˈɝ	v	i	b	u	v
oɪ	k	ø	v	q	c
ˈoɪ	g	w	d	y	l
ˈɑɪ	dʒ	y	g	i	w
ˈi	ʃ	e	k	ö	m
i	m	ɥ	p		f
ɝ	l	ɛ	ʃ		s
ˈɔ	θ	u	ʒ		d
ˈɪ	f	o	s		j
ˈeɪ	b	a	m		p
ɑɪ	tʃ	ã	h		b
ˈɛ	s	œ̃	l		k
ˈɑ	p	j	t		g
ˈɑʊ	d	ɛ̃	ʁ		t
ˈoʊ	h				
ʌ	t				
ˈʌ					
eɪ					
ˈæ					
ɔ					
oʊ					
ɪ					
ɑ					
j					
æ					
ɛ					
ɑʊ					
w					
ɹ					
z					

TABLE 4. Results of the spectral method on three natural language corpora.

We have applied this procedure to build a phonotactic graph for each of the three corpora used in the previous section. Table 4 shows the partitioning of phonemes resulting from the application of spectral clustering to these three graphs.[12] The classification of English phonemes is not perfect, but it is much better than what Sukhotin's algorithm would predict. In particular, the splitting of vowels into a stressed and unstressed version does not seem to bear on the results.[13] The only errors are the misclassifications of four consonants as vowels: /j/, /w/, /ɹ/, and /z/. Classifying glides with vowels seems to be a consistent behavior of the spectral method, since it also occurs for French (more on this below). Although /ɹ/ and /z/ appear relatively frequently after a consonant, the same holds for other consonants as well, and it is not clear why the method would specifically misclassify these two phonemes with vowels. Since they stand right next to the boundary between vowels and consonants, one hypothesis is that their misclassification stems from the denominator of the conductance (see Goldsmith & Xanthos 2008, appendix C) rather than its numerator: in other words, that they help balance the volumes of the groups more than they contribute to their distributional homogeneity.

The results for French are quite similar, as the glides (/j/, /w/, and /ɥ/) are also misclassified as vowels. The reason for this seems to be that we have chosen to define a phoneme's context as the previous phoneme in a word, and for glides this phoneme is much more likely to be a consonant than a vowel (in both languages). In fact, if we define the context of a phoneme as the two phonemes that surround it, we find that glides are correctly classified as consonants, and so are English /ɹ/ and /z/.[14]

The results for Finnish are exactly identical to those of Sukhotin's algorithm, that is, the symbol <q> is misclassified as a vowel (see §3.1). This behavior recurs when the context is defined as the surrounding phonemes or the following one; in the latter case, <n> is further misclassified as a vowel—the least vowel-like one.

Overall, it seems that the spectral approach performs considerably better than Sukhotin's algorithm. The spectral approach's tendency to label glides as vowels can be fixed by modifying the definition of context to take into account the following phoneme as well, which is also the case in Sukhotin's algorithm. Insofar as the spectral method's classification of English phonemes is considerably better than that of Sukhotin's algorithm, it seems more robust with regard to variations in the encoding scheme being used. On the whole, we consider this a significant step toward an unsupervised solution to the problem of learning major phonological categories.

3.3. MAXIMUM LIKELIHOOD: HIDDEN MARKOV MODELS. The third method that we have explored poses the problem of phone categorization in terms of a natural optimization problem: suppose we construct a finite-state device with a small number of states (two states, in most of the cases that we examine). Each state is in principle capable of generating all of the phonemes of the language. In fact, each state has its own probability distribution for generating each of the symbols of the language, and each state has a probability distribution for transitioning to itself or any of the other states (typically,

[12] In this table, the ordering of phonemes reflects their ordering on the (normalized) Fiedler vector, that is, the phonemes at the top of each column are those that are located at each extreme of the vector.

[13] Interestingly, the stressed and unstressed versions of several vowels (/ʊ/, /oɪ/, /i/, /ʌ/) are actually located next to one another on the Fiedler vector.

[14] Note that when a phoneme's context is defined as the phoneme that follows it, glides are correctly classified, as are English /ɹ/ and /z/, but other divergences occur in English: /s/, /ŋ/, and /n/ are misclassified as vowels, and /i/ as a consonant.

there is only one other state). We desire to find the assignment of probability distributions for these two functions (emission distribution and transition distribution) for each state in such a way that the probability of the corpus—which is to say, of the data sample—is maximized (see Figures 3 and 4).

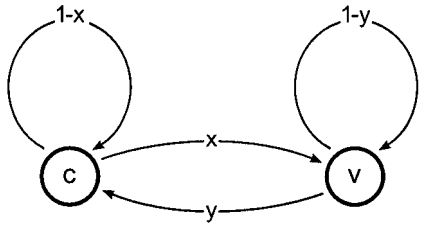

FIGURE 3. A simple two-state hidden Markov model.

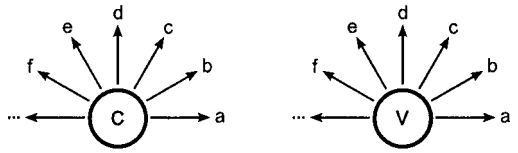

FIGURE 4. The states in the HMM.

As this task is described, it corresponds directly to a well-known task in machine learning, training a hidden Markov model (henceforth, HMM), and there is a well-known algorithm that can rapidly find the parameters for these distributions, and it does this in such a way that the data is assigned the highest probability. (Actually, the algorithm is sure to find a local maximum, and not guaranteed to find a global maximum; this difference does not seem to play a role in the cases we are looking at.) We employed this Baum-Welch algorithm (a special case of expectation maximization) in order to find the appropriate distributions on the basis of the training data that we have described for each language. (For technical discussion of HMMs, we refer the reader to Charniak 1993 and Jelinek 1997.) The intuition that lies behind this is that if there is local structure to the sequence of symbols that the HMM is being trained on, then it will find a way to distribute the sounds differentially to the two states, and to train the transition probabilities between the two states as well. If there is a tendency in the data to alternate between sounds of two different sets, then the system will assign those sounds to different sets, and assign a higher probability to the transitions between distinct states than that which it assigns to the 'transitions' that allow the system to remain in the same state. If, by contrast, the data has different characteristics—if, for example, the data shows stretches of several segments from one subgroup, followed by stretches of segments from another group—then the system will assign higher probabilities in one of the states to the one subgroup, and higher probabilities in the other state to the other subgroup, and at the same time, it will assign relatively low transition probabilities to links between states 1 and 2 in either direction. We observe below that each of those descriptions will be borne out in actual linguistic cases: the former in the case of vowels and consonants, and the latter in the case of vowel harmony.

OBSERVING RESULTS FOR ENGLISH. The HMM takes about 2,000 iterations through the English data we used (on the order of 50,000 words in each case) in order to arrive

at a steady state, but it arrives at a state not far from that steady state within about fifty iterations. At that point, we can observe three aspects of the results: the emission probabilities, the transition probabilities, and the common convergence despite random initial assumptions.

PHONE	LOG RATIO	PHONE	LOG RATIO
ð	−999	u	2.22
ŋ	−999	ɝ	2.30
w	−999	i	2.31
n	−999	aʊ	2.32
l	−999	aɪ	2.83
h	−999	oʊ	3.93
ʃ	−999	eɪ	4.99
ɹ	−999	ˈaɪ	5.11
m	−999	ˈoɪ	5.81
v	−999	ˈi	7.39
ʒ	−999	ˈoʊ	12.7
dʒ	−999	ˈaʊ	275
b	−999	ˈeɪ	262
j	−999	oɪ	263
f	−999	ˈu	999
g	−829	ʌ	999
k	−576	ɛ	999
tʃ	−361	æ	999
θ	−5.19	ˈɝ	999
p	−4.37	ɑ	999
d	−3.95	ɪ	999
s	−2.75	ˈæ	999
t	−2.20	ɔ	999
z	−1.37	ˈɛ	999
		ˈɑ	999
		ˈɔ	999
		ˈɪ	999
		ˈʌ	999
		ˈʊ	999
		ʊ	999

TABLE 5. Phones and the log ratios of their emissions, comparing the two states of the HMM for English.

First, and most importantly, we can observe the relative log probabilities of the EMISSION of each phoneme across the two states, that is, for each phoneme p, $\log \frac{pr_{state_1}(p)}{pr_{state_2}(p)}$. This is given in Table 5, where a positive value indicates a phoneme that the network prefers to generate in state 1, while a negative value indicates a phoneme that the network prefers to generate in state 2. We use '999' to represent a ratio greater than or equal to 999 (typically because the denominator in the expression is zero, or close to it), and similarly for '−999'. The segments are naturally divided into two groups, based on whether the ratio is positive or negative. This informs us of the categorization that the system has learned for the two sets of segments. As we see, the method is thus 100 percent successful. The ENTROPY of the emissions of a state is the average base-2 logarithm of the reciprocal of the emission probabilities, and it is the usual way of looking globally at a set of probabilities; when the entropy decreases, more of the probability is being focused on a smaller subset of the candidates. We can see this 'focusing' explicitly in the top graph of Figure 6, where the fall in the entropies

shows that both states are learning to specialize and divide their labor, so to speak, between them, with one state specializing in consonants and the other in vowels.

Second, we can inspect the TRANSITION probabilities for the two states. We can do this in a few ways. (i) We can consider the final steady-state values of the four state transition probabilities, as shown in Figure 5. (ii) We can plot the evolution of these four transitions on a graph, where the *x*-axis represents 'time', or the iterations in the learning regime, as in Fig. 6. We present graphically the evolution of the transition probabilities over the course of the first forty iterations during the learning phase.[15]

FIGURE 5. English: two-state finite-state automaton.

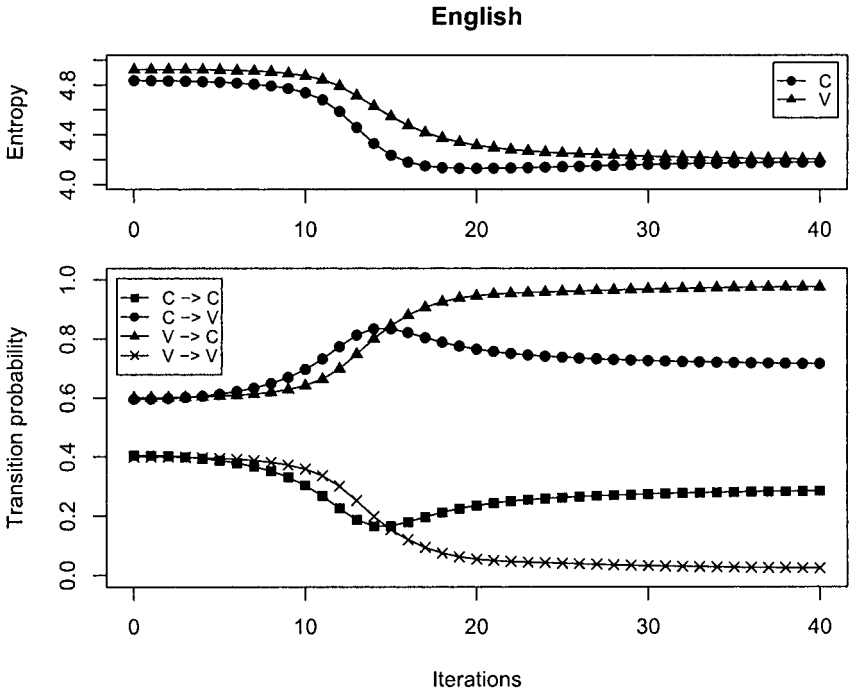

FIGURE 6. English transitions.

[15] A moment's study of the data displayed in Table 5 leads one to the question of WHY there seems to be a span of vowels (/u/, /ɝ/, /i/, /aʊ/, /aɪ/, /oʊ/, /eɪ/, /ˈaɪ/, /ˈoɪ/, /ˈi/) and of consonants (/θ/, /p/, /d/, /s/, /t/, /z/) whose log ratio is surprisingly close to zero. There appear to be two separate answers to this question. The data that we have used, a CMU wordlist widely available at http://www.speech.cs.cmu.edu/cgi-bin/cmudict, includes a number of words in which two vowels appear adjacent to each other: for example, *overarching* = /ˈo͡ʊvɝˈɑɹt͡ʃ ɪŋ/, *biotic* = /biˈɑtɪk/. This appears to be the reason why a number of unstressed diphthongs have such a small log ratio. The consonants whose log ratio is small are those that tend to appear in clusters with high frequency, and we return to their behavior in the next section, when we look at the way three-state HMMs analyze this data.

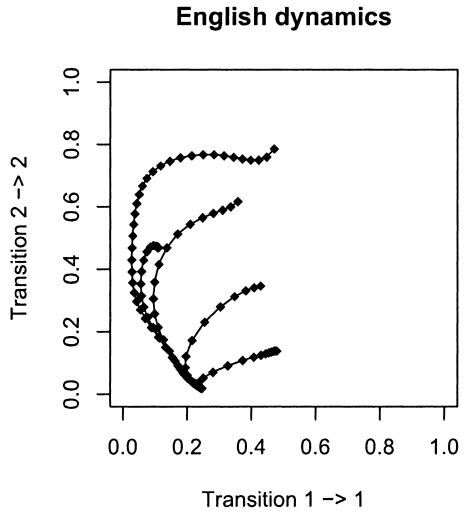

FIGURE 7. Five paths to the learning of English transitions. *x*-axis is prob (state 1 → state 1); *y*-axis is prob (state 2 → state 2). All movement is downward and to the left.

Third, we can observe the evolution of the transition probabilities over the course of several different learning experiments, as in Figure 7. This figure shows the evolution of five learning experiments. Each point resides in a two-dimensional space, with coordinates (x, y); the first coordinate x marks the probability of transition from state 1 to state 1, and y is the probability of the transition from state 2 to state 2. We refer to this space as 'phase space'; its coordinates represent transition probabilities. Starting values for these probabilities were chosen at random from near the center of the square extending from (0,0) to (1,1). As we see, the values expressed during the learning process converge on the same final point in this phase space.[16]

ALTERNATING AND HARMONY SYSTEMS. When the transition from each state to itself is considerably less than 0.5, as is the case here, then the system has learned to preferentially ALTERNATE between the two states (which we may reasonably label 'V' and 'C' once we inspect the identity of the segments being generated by them). It turns out that, when we analyze vowel harmony data in parallel fashion below, the system reaches equilibrium at a point in a different quadrant, one where the probability of the transitions from one state back to itself is close to one; this is a natural characterization of a HARMONY system. See Figure 8 for a graphical representation of these two regions in phase space: the harmony system is the upper right quadrant, and the alternating system is the lower left quadrant.

OBSERVING RESULTS FOR FRENCH. Turning now to a corpus of French, we find essentially the same results; the results after 1,200 iterations are given in Table 6 and Table 7. Figure 9 presents the transition data graphically. Figure 10 illustrates the early and

[16] We see here that when the starting position for the probability of 1 → 1 transition in phase space is further from the final correct position, there is a strong tendency for the learning algorithm to overshoot the correct value along this dimension before correcting the 2 → 2 probability. This tendency deserves closer study.

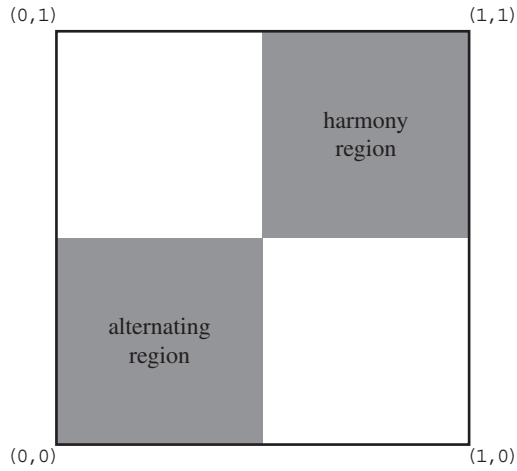

FIGURE 8. Phase space, defined by probability of each state transitioning to itself.

most important part of the learning during a single training, showing both transition probabilities and state emission entropies, as above. See also Figure 11, which shows the passage to learning for three systems starting from four different initial random values. Again, as in English, the end point of the learning is a spot in the alternating region of phase space.

PHONE	LOG RATIO	PHONE	LOG RATIO
s	5.26	ə	−999
t	7.96	ɛ	−999
g	600	ɔ	−999
p	933	u	−999
d	999	i	−999
k	999	ɑ̃	−999
ʒ	999	ɛ̃	−999
m	999	ɔ̃	−999
n	999	ø	−999
l	999	œ	−999
f	999	a	−473
b	999	y	−11.6
r	999	o	−10.5
ɲ	999	œ̃	−5.53
v	999	e	−4.93
ʃ	999		
h	999		
ɥ	999		
w	999		
j	999		
z	999		

TABLE 6. Phones and the log ratios of their emissions, comparing the two states of the HMM for French.

As in English, vowels and consonants are correctly categorized. As above, we use '999' to represent a ratio greater than or equal to 999, and similarly for '−999'. The 'consonant' identified as an /h/ is the *h*-aspiré, which is treated as a phoneme in this data set.

	to state 1	to state 2
from state 1	.23	.77
from state 2	.98	.02

TABLE 7. Transition probabilities, two-state HMM for French.

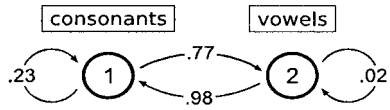

FIGURE 9. French: two-state finite-state automaton.

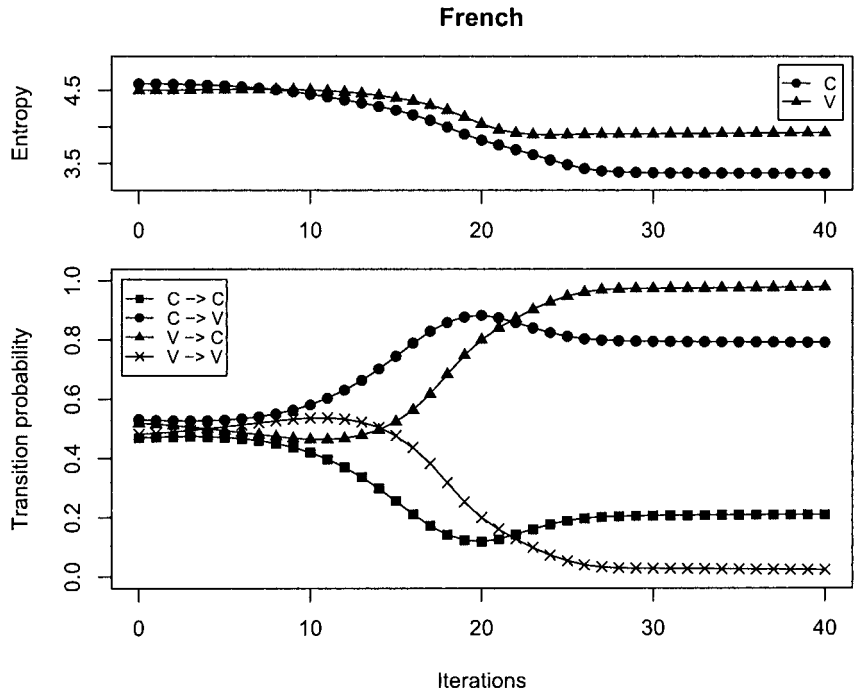

FIGURE 10. French transitions.

The results that are described here, which are similar to the results we have found in all of the data sets we have looked at, suggest that an effective procedure for dividing vowels and consonants into two distinct categories is to train a two-state HMM on a string of symbolic representations of phones, in order to find the parameters that maximize the probability of the data. To turn the same point around, we could say that IF linguists define, at a high level of abstraction, their goal to be the development of a model that maximizes the probability of the data, then if they choose to divide the phonological segments of a spoken language into two sets, there is strong reason to believe that the two sets of segments that EMERGE from this distributional task are the segments that have, since the time of the Greeks, been called VOWELS and CONSONANTS.

It is perhaps not too strong to describe our results so far as the 'discovery' of vowels and consonants—though one might also call them the discovery of a method to discover

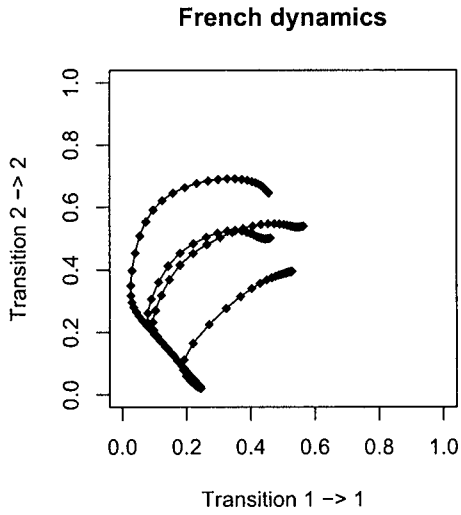

FIGURE 11. Dynamics of learning French V/C. All movement is downward to the left.

vowels and consonants (distinct from, and largely simpler than, that of Ellison, discussed above). These two categories are doubtless the most important and fundamental in all of phonology.[17] What question, or questions, comes next? What other aspects of phonological structure are both basic and robust in a cross-theoretical way? That is, what aspects of phonology would all perspectives on phonology agree upon as the next most significant, after the discovery of the vowel/consonant distinction?

Two possible answers come easily to mind. One is vowel harmony; the other is syllable structure. We turn to each of these two phenomena in the next two sections.

4. LEARNING VOWEL HARMONY. By VOWEL HARMONY we mean the strong tendency of a language to impose a restriction on the choice of vowels inside phonological words in such a way that each word selects vowels from only one of a relatively small number of subsets (typically two) of the vowels of the language. The subsets may overlap in some cases (in which case we speak of 'neutral' vowels); the subsets of vowels are typically, but not always, natural from a phonetic point of view. A common pattern is that the front vowels of a language form one set, and the back vowels another; see van der Hulst & van de Weijer 1995 for an overview of vowel harmony systems.

[17] Whitney, for example, wrote in 1865:

> The question of the mutual relation of vowels and consonants, of what constitutes the essential distinction of either class from the other, is one of primary interest as regards the theory of the alphabet, and does not appear to me ever to have been taken up and discussed in a wholly satisfactory manner . . . Those who study the spoken alphabet have been content . . . to treat the vowels and consonants as two independent bodies, partners in the work of articulate expression, indissolubly married together for the uses of speech, yet distinct individuals, to be classed, arranged, and described separately, and afterward set side by side. (in Silverstein 1971:198)

Whitney proceeds to argue for a cline, stretching from obstruents through liquids to glides and thence to vowels. Bloomfield (1933:130) proposes that phonemes are divided into PRIMARY and SECONDARY (prosodic) phonemes, and primary phonemes are divided into consonants and vowels. Trubetzkoy (1969:92) does likewise, focusing on properties of phonemes rather than on the phonemes themselves (a natural thing, since he was creating structuralism in so doing); these properties divide into vocalic, consonantal, and prosodic properties.

The task of identifying vowel harmony is thus a problem of category discovery. Our question then is this: is there an algorithm that takes as its input a set of phonological data, and returns an answer of 'No!' when the data does not display vowel harmony, and returns a labeling of the vowels into appropriate harmonic subgroups when the data is drawn from a language with vowel harmony? In the next two subsections, we explore the effectiveness of spectral methods and maximum-likelihood/HMM methods in answering this challenge. As noted above, we have used a corpus of 44,040 Finnish words in standard orthography as our training set. The traditional account of Finnish is that there are two neutral vowels, represented by graphemes <i> and <e>, and a vowel harmony system on backness and frontness. The back vowels are <u>, <o>, and <a>, while the front vowels are <ö>, <ä>, and <y>. For this experiment, we have extracted from each word the subsequences consisting of just the vowels; this leaves us with 15,412 distinct vowel sequences in the lexicon, and 101,913 vowel-type occurrences.

4.1. SPECTRAL APPROACH. In §§3.1 and 3.2, we described two methods for classifying the phonemes of a corpus into two categories that correspond well with vowels and consonants. Considering the problem of vowel harmony reveals a fundamental difference between these two methods: Sukhotin's algorithm is able to identify vowels and consonants because it is BY DESIGN a device for detecting alternating patterns, and vowels and consonants constitute a typical instance of this pattern; by contrast, spectral clustering is able to do so because it is a device for grouping similar objects together, and all members of the set of vowels (respectively consonants) are similar with regard to their tendency to alternate with members of the other category. As a consequence, Sukhotin's algorithm is helpless to learn vowel harmony, because members of a harmony category tend NOT to alternate with members of the other, whereas the spectral approach is able to shed light on this phenomenon on the basis of the exact same criterion as before: distributional similarity.

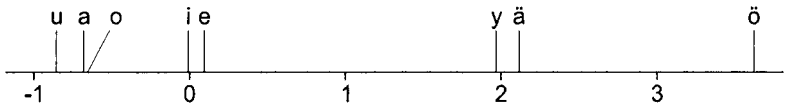

FIGURE 12. Second eigenvector of the graph of Finnish vowels.

Thus we have applied the spectral method introduced in §3.2 without any change to the corpus of Finnish vowels, and it results in a classification where front vowels and neutral vowels form a single group, while back vowels are in a group of their own.[18] As shown in Figure 12, the positions of vowels on the second eigenvector of the graph reveal a more fine-grained structure: neutral vowels <i> and <e> constitute a separate cluster, and the set of front vowels is divided into a cluster comprising <y> and <ä> and another cluster containing only the vowel <ö>.

While the spectral approach is able to capture certain relevant features of a vowel harmony system, it offers no way of handling the fact that in such a system, phonemes

[18] Recall that the clustering algorithm that we use invariably returns two categories. From Fig. 12, it may seem that neutral vowels are more similar to back vowels. But the spectral representation just serves as a filter that discards a large proportion of possible partitionings; ultimately, the crucial criterion is the CONDUCTANCE (see Goldsmith & Xanthos 2008, appendix C, for general discussion) associated with each partitioning, and not the distances induced by the spectral projection.

may in effect belong to more than one group—as the neutral vowels of Finnish do. One way of overriding this limitation would be to apply a FUZZY clustering algorithm (see e.g. Bezdek 1981). The specificity of such algorithms lies in their ability to characterize set membership in probabilistic terms: thus, it is likely that neutral vowels would 'belong' to both groups (back and front vowels) with approximately the same probability, while the vowels composing the core of these groups would 'belong' to one of them with a much higher probability than to the other.

In any event, this example demonstrates the superior generality of the spectral approach over Sukhotin's algorithm, as the former can handle the different patterns of distributional similarity involved in the learning of the vowel/consonant distinction and of a vowel harmony system.

4.2. MAXIMUM-LIKELIHOOD METHODS. We turn now to the task of discovering vowel harmony by maximum-likelihood methods, parallel to the discovery of the vowel/consonant distinction described in §3.3 above. The method is simplicity itself: we train an HMM (one that is identical in its initial form before training to the one used in the earlier analysis) on the sequence of vowels in each word, where what counts as a vowel has already been determined. If the transition parameters for the states map to a point in the 'harmony' part of our phase space—and especially if they map to a point very close to (1,1)—then we can infer that the system has discovered a vowel harmony system. Those vowels that are principally emitted by just one state constitute one of the vowel harmony classes, while the vowels that are principally emitted by just the other state constitute the other vowel harmony class; vowels that are emitted by both states, with roughly equal probabilities, are neutral vowels.

VOWEL	LOG RATIO	VOWEL	LOG RATIO
ö	999	o	−7.66
ä	961	a	−927
y	309	u	−990
e	0.655		
i	0.148		

TABLE 8. Log ratios of emission probabilities for Finnish vowels.

We find that the vowels in our Finnish corpus are quickly and easily distributed along a single dimension, as in Table 8. The vowels seem to fall into four categories: those with a very large positive log ratio (the front vowels, <ö>, <ä>, and <y>), those with a very large negative log ratio (the back vowels, <a> and <u>), those with a log ratio very close to zero (the two neutral vowels in Finnish, <e> and <i>), and, unexpectedly, a fourth category, <o>, which is a back vowel and yet is surprisingly distant from its congeners <a> and <u>. In any event, the system as it stands gives the right results, in the following sense. The optimal path through the finite-state device for a word with only front vowels (or a mixture of front vowels and neutral vowels) keeps the system in state 1, and in a word with only back vowels (or a mixture of back vowels and neutral vowels) in state 2. The emission and transition results after 1,000 iterations of training are given in Table 8 and Table 9. Table 8 shows the separation of the vowels into two groups, and Table 9 shows that this is a harmony system, by virtue of the fact that the transition from each state to itself is much higher than the transition to the other state; this same point is represented graphically in Figure 13, but note that this last figure is deceptive; the labeling there makes it appear that vowels

have unambiguously been divided into two categories, when in fact the structure is a good deal more articulated, as noted earlier.

	to front Vs	to back Vs
from front Vs	.90	.10
from back Vs	.03	.97

TABLE 9. Transition probabilities, two-state HMM for Finnish vowel harmony.

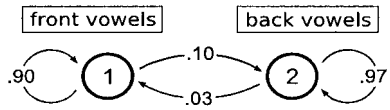

FIGURE 13. Finnish vowel transitions.

As we have presented the use of the HMM so far, its effectiveness might as well have been limited to the ease with which it can be used to find parameters that maximize the probability of the data. There is, however, a second aspect of HMMs that is worth remarking upon. After the appropriate parameters for an HMM have been learned, the typical use to which an HMM is put is this: for each string of data (here, each Finnish word) the HMM will find the UNIQUE PATH through the states that generates the data with the highest probability. Typically, there will be a large number of possible paths through the network that will generate the same string, because each state has a nonzero probability of generating each of the symbols in the alphabet, and each state-to-state transition is greater than zero. But there is a straightforward algorithm that allows us to determine which SINGLE path through the network generates a given string with a higher probability than any other path. Now, this is particularly interesting in the case at hand, because for the two neutral vowels of Finnish, both states generate both vowels with nearly equal probability. But because of that fact, and because the probability of transitioning from one state TO THE OTHER is very low, it follows that a neutral vowel in a front-vowel word will be generated by the front-vowel state, while a neutral vowel in a back-vowel word will be generated by the back-vowel state.

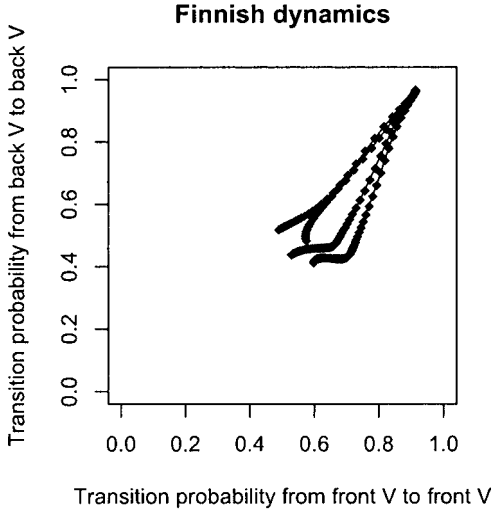

FIGURE 14. Finnish transition evolution. All movement is upward to the right.

In Figure 14, we see a graphic rendering of the evolution of the transition probabilities, that is, the evolution of the system in phase space. As before, the axes on this graph plot the probability of transition from each state back to itself; the *x*-axis marks the probability of a transition from front-vowel state to front-vowel state, and the *y*-axis marks the probability of a transition from back-vowel state to back-vowel state. In each of our training instances, we begin our probabilities with a random value not too far from a uniform distribution, and hence roughly in the middle of this [0,1] square. We see the transition probability values move consistently toward the (1,1) point, and all systems that are in the upper right quadrant are naturally labeled as HARMONY systems: once in a given state, they prefer to remain in that state; see the discussion above contrasting harmony and alternation.

5. LEARNING ASPECTS OF SYLLABLE STRUCTURE.

5.1. SYLLABLE STRUCTURE AS MAXIMUM LIKELIHOOD. The discussion in §3.3 assumed without discussion that we would divide the segments of a language into two categories, vowels and consonants. There is no reason, however, to restrict maximum-likelihood estimation (such as we seek with an HMM) to two categories. We are free to ask a question such as this: if we devise a three-state finite-state automaton, and train it on data from English or French (or any other language) in order to establish its emission and transition probabilities so as to maximize the probability of the training data, what will be generated by each of the three states? The Baum-Welch learning algorithm will assign a function to the third state, one that expresses the next most important statistical dependency in the data, compared to the two-state model—but what would that be? The two-state model is incapable of capturing any sort of dependency between adjacent vowels and between adjacent consonants, but the fact is that in our data (as in most languages), there are far more sequences of adjacent consonants than there are of adjacent vowels. We would therefore think it likely that the learning algorithm would use the new state in order to divide the work of generating consonant sequences across two different states, trying to find a way to predict which consonants occur first in a cluster, and which appear second in a cluster.

On the basis of this reasoning, we expected that when presented with data from French, the system would divide the work of generating consonant sequences into two states, one of which generated coda consonants and one of which generated onset consonants. What we found, however, was slightly different. Although the system passed through a state that was roughly of that sort, it would eventually find a different organization of the data, in which one of the states was playing the same role as the consonant state in the two-state models of §3.3, while the other was essentially responsible for generating the last element of an onset cluster. In this section, we describe that result and suggest some areas for future research.

	to state 1 (V)	to state 2 (C)	to state 3 (cluster)
from state 1 (V)	.01	.94	.05
from state 2 (C)	.71	.08	.21
from state 3 (cluster)	1	0	0

TABLE 10. Transition probabilities, three-state HMM for French.

In Figure 15, we see a representation of a typical instance of learning the transition probabilities. The final equilibrium state for the transition probabilities is what is seen at the end, and it is displayed in Table 10. We can easily see that there is a brief initial learning period leading to a tentative hypothesis of the parameters, reached at about iteration 50, followed by a period of near quiescence up to iteration 200, followed by

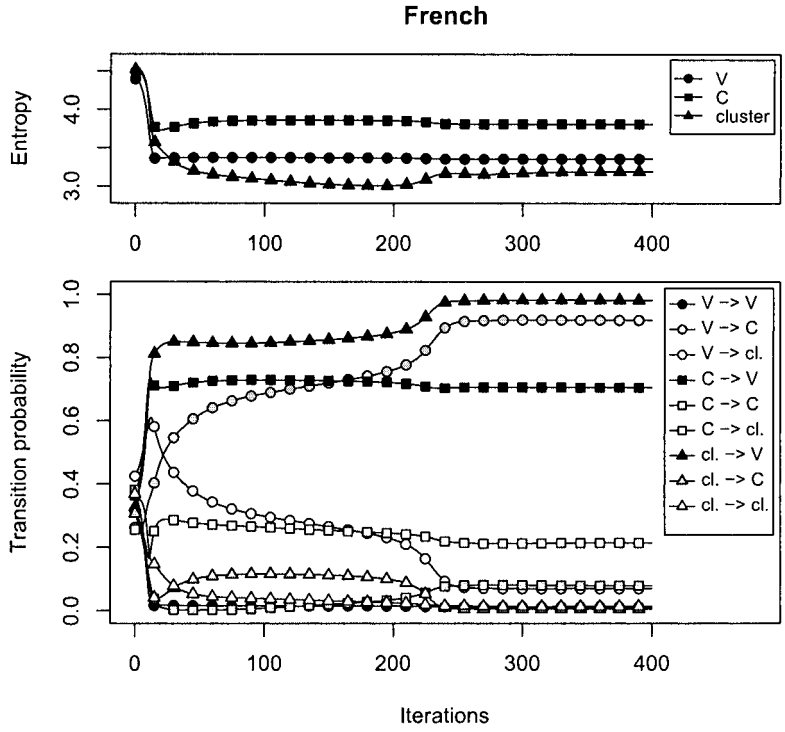

FIGURE 15. French three-state learning dynamics for transition and entropy.

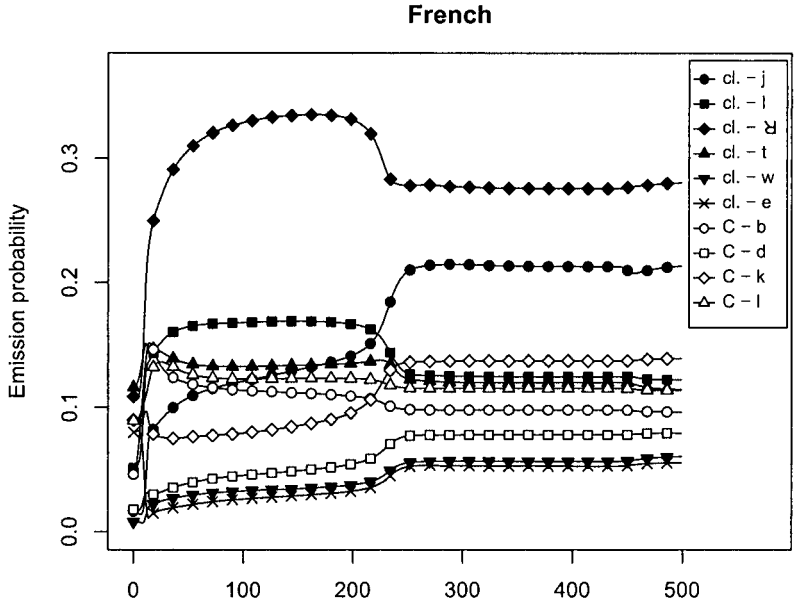

FIGURE 16. Crucial emission changes during French three-state learning dynamics.

a rapid shift to the final equilibrium state by iteration 250. The situation between iterations 50 and 200 represents a hypothesis in which both consonant states can transition to the other, but neither transitions much to itself. But this is abandoned by the discovery of a better structure after iteration 250, in which the two consonant states take on quite different characters. One of them (state 3 in this case) becomes used LESS; it is used primarily to generate the last member of an onset cluster, and it ALWAYS transitions to the vowel state. For mnemonic purposes, we refer to this as the 'cluster state', and the other state as the 'consonant state'. We find this pattern consistently, and we believe that a deeper understanding of this is called for. If we think of the state-transition probabilities as specifying a point in a six-dimensional space (a hypercube), then we may describe this change as one that brings the system to one of the edges of the hypercube (the edge corresponding to transitions out of the cluster state having values (0,1,0)), which in some sense is suggestive of a categorical, rather than a gradient, analysis.[19] When we look more closely at what emission probabilities change along with the transition-probabilities shift during the rapid change from iteration 200 to 250, it turns out that it is only a small number of parameters that are modified; these are shown in Figure 16. The maximum-likelihood parameter values for transition and emission probabilities are given in Table 10 and Table 11. We omit segments whose emission probabilities fall below 0.01. See Figure 17 for a partial graphical summary.

FROM STATE 1	PROB	FROM STATE 2	PROB	FROM STATE 3	PROB
a	.19	ʁ	.14	ʁ	.28
e	.18	s	.11	j	.21
i	.17	t	.10	l	.12
o	.10	k	.10	t	.11
ɛ	.06	l	.08	w	.06
ɑ̃	.06	p	.07	e	.06
y	.05	m	.06	m	.03
ə	.04	d	.06	ɥ	.02
ɔ̃	.04	n	.06	s	.01
u	.03	b	.05	ɛ̃	.01
ɔ	.03	f	.04	n	.01
ɛ̃	.03	g	.03	y	.01
		v	.03	k	.01
		z	.03		
		ʒ	.02		
		ʃ	.02		

TABLE 11. Emission probabilities, three-state HMM for French.

A detailed examination of the final parameters of the model (Tables 10 and 11) reveals the difference in behavior of the consonant state and cluster state (states 2 and 3). Consider the case of single, intervocalic consonants (recall that state 1 is the vowel state). In this context, transitioning via the consonant state is about thirteen times more probable than via the cluster state.

$$(4)\quad Pr(1 \rightarrow 2) \cdot Pr(2 \rightarrow 1) = 0.94 \cdot 0.71 = 0.67$$
$$Pr(1 \rightarrow 3) \cdot Pr(3 \rightarrow 1) = 0.05 \cdot 1 \simeq 0.67/13.3$$

[19] The transitions from each state are determined by two degrees of freedom, so to speak, because the probabilities of the three transitions must add up to 1.0; since there are three states, that means that there are six parameters, and hence a specification of the transition probabilities can be thought of as specifying a point in a part of a six-dimensional space.

Thus, all other things being equal, in order for an intervocalic consonant to be emitted by the cluster state rather than by the consonant state, it should have an emission probability by the former more than 13.3 times greater than by the latter, a constraint that only glides (and all glides) satisfy; most of the time, an intervocalic consonant will be emitted by the consonant state. Virtually any consonant occurring before another consonant should be emitted by the consonant state, whatever precedes it, because the probability of transitioning from the cluster state to something other than the vowel state is close to zero.

(5) $Pr(3 \rightarrow 2) = Pr(3 \rightarrow 3) = 0$

Therefore, in principle, a postconsonantal consonant emitted by the cluster state could occur only in prevocalic position. In postconsonantal and prevocalic position, consonants are emitted by the cluster state unless their emission probability by the consonant state is more than 3.5 times greater than by the cluster state.

(6) $Pr(2 \rightarrow 2) \cdot Pr(2 \rightarrow 1) = 0.08 \cdot 0.71 = 0.06$
$Pr(2 \rightarrow 3) \cdot Pr(3 \rightarrow 1) = 0.21 \cdot 1 \simeq 0.06 \cdot 3.5$

The consonants that will rather be emitted by the cluster state in this context are the glides and liquids, along with /t/ and /m/. To sum up, aside from an occasional intervocalic glide, the model still favors transitioning from the vowel to the consonant state and vice versa, possibly looping within the latter for a while. Sometimes, before it transitions back to the vowel state, it takes a detour via the cluster state to emit a prevocalic glide, liquid, /t/, or /m/. Overall, the base-2 log probability of French material is improved by 1.97 percent by virtue of moving from a two-state model to a three-state model.[20]

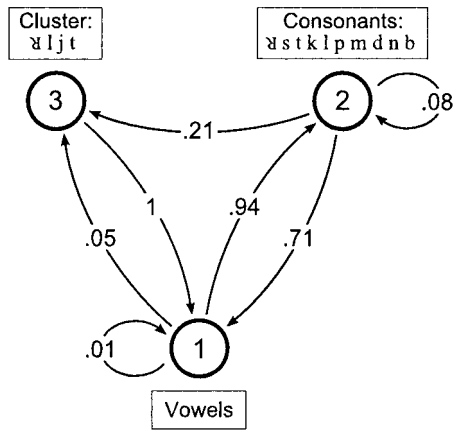

FIGURE 17. Three states for generating French strings.

This model generates sequences like /abʁa/ and those like /aʁba/ in different ways (the logic of the situation is parallel to that discussed in the vowel harmony case). /ʁ/ and /b/ can both be generated by both the consonant and the cluster states, but the transition probabilities between these two states are quite different, and the relevant

[20] On the set of 21,574 French words, the inverse log probability was 611,376 according to the two-state model, and 599,293 according to the three-state model. The comparable values for the 58,156 English words were 1,926,121 and 1,872,089, corresponding to an improvement of 2.75 percent. This means that for both corpora, the three-state model fits the data better than the two-state model—at the cost of an increase in complexity of the model, as measured by a greater number of parameters.

Emit:	while in state:	prob	transition	prob	
a	1	.19	1 → 2	.94	
b	2	.05	2 → 2	.08	probability: 1.35×10^{-5}
ʁ	2	.14	2 → 1	.71	
a	1	.19			
Emit:	while in state:	prob	transition	prob	
a	1	.19	1 → 2	.94	
b	2	.05	2 → 3	.21	probability: 9.98×10^{-5}
ʁ	3	.28	3 → 1	1	
a	1	.19			
Emit:	while in state	prob	transition	prob	
a	1	.19	1 → 2	.94	
ʁ	2	.14	2 → 2	.08	probability: 1.35×10^{-5}
b	2	.05	2 → 1	.71	
a	1	.19			
Emit:	while in state:	prob	transition	prob	
a	1	.19	1 → 2	.94	
ʁ	2	.14	2 → 3	.21	probability: 1.03×10^{-8}
b	3	.001	3 → 1	1	
a	1	.19			

TABLE 12. Structural differences between /abʁa/ and /aʁba/.

calculations are given explicitly in Table 12. The path through the HMM that produces the sequence /abʁa/ with maximum probability is the one that emits those symbols by following the sequence of states 1 2 3 1, while the path that produces the sequence /aʁba/ with maximum probability involves the sequence of states 1 2 2 1. While in theory there are 3^4 possible state sequences, that is, paths, to generate any sequence of four symbols, in practice we can ignore any sequence that does not generate the vowels from state 1, and we can ignore any path that involves a sequence 3 → 1 or 3 → 2, since those transition probabilities are close to zero. We have chosen this example to illustrate the point we noted above, that state 3 is effectively dedicated to generating the last element of an onset cluster.[21]

Needless to say, a range of further cases should be studied. We would predict, for example, that a language that contains an optional coda but no onset clusters will use its third state to generate coda consonants, and an interesting study would be to look at further languages that, like English and French, have both codas and onset clusters, to see under what conditions the third state is used to account for codas, and under what condition for onset clusters.[22]

[21] It is worth noting that this way of observing the behavior of the model, namely in relation to a small number of selected phonological patterns, is often more revealing of the phonotactics of the language than the direct study of probabilities assigned by the model to unrestricted phoneme sequences. For instance, if we consider all sequences of two consonants that are LOGICALLY possible given the inventory of French phonemes, we find that in initial position, the three-state model assigns the highest probability to /rr/, which is not a legal cluster in French. This is not surprising, as HMMs are based on the assumption that state transitions and symbol emissions are independent; as a result, illegal clusters involving very frequent phonemes are often more probable than legal clusters of rarer phonemes.

[22] For a discussion and comparison of the three major themes in the treatment of syllabification, see Goldsmith 2009; these three focus, respectively, on segment-transition possibilities, on constituent structure, and on waves of sonority.

6. DISCUSSION. There are two general points that arise out of the work discussed here. The first is that a nontrivial problem (and perhaps more than one) can be solved without recourse to a rich set of prior assumptions of the sort that would be good candidates for inclusion in universal grammar. Indeed, not only do we not need to have recourse to a rich universal grammar, but also the principle that we have employed ('maximize the probability of the data', or maximize the likelihood) is not far from a basic principle of rationality. Experts may argue over how that principle should be made precise, and the details do matter; but there is no call for an explanation that relies on genetic endowment or Darwinian evolution.

The second general point is that this article has focused on questions of METHOD of empirical analysis. Like any discussion of method, the proof, or test, of the method here lies entirely in the results that flow from the method and their value to us as linguists. But it has long been a shibboleth in theoretical linguistics that a focus on methods of data analysis is misplaced effort: in this view, the path from observations to hypothesis is the sociology of the scientific laboratory, and of no interest as such to science or scientists; all that matters is providing evidence in support of a hypothesis, regardless of how the hypothesis is found.

In our view, those who embrace this view have gone too far. The position undoubtedly has its origins in the proposals of the logical empiricists (notably those of Reichenbach 1938) to distinguish the CONTEXT OF DISCOVERY from the CONTEXT OF JUSTIFICATION: how a scientist comes up with an idea is a good story for a biography, but it is not the stuff of which science is made. While this is doubtless true, the point can be overmade, and it can lead to a perspective in which scientists feel they may pick and choose the data that serves their hypothesis best.[23] We have argued BY DOING that well-conceptualized decisions about method may lead to surprising conclusions that shed considerable light on the nature of language.

Of course, if we have focused on method, it is method at an abstract level. In the treatment of graph-theoretic approaches to phonological analysis, we have emphasized the conceptual content of the approach, and the particular numerical algorithms used to calculate eigenvectors (to take one example) are of no particular interest, once we understand how they work. In the maximum-likelihood models, we employed hidden Markov models in order to compute the appropriate values of the parameters, but the HMMs themselves are of no particular interest, once we understand the conditions under which we can use the standard learning algorithms to optimize a function (which in our case we choose to be the probability of the data, given certain structural constraints).

[23] A clear example of going too far in such a direction, in our opinion, is offered by Chomsky (2000), who presents a case in favor of a style that he refers to as Galilean. Of course, any two people can look at what Galileo did and draw radically different lessons from his successes, but Chomsky suggests that '[w]hat was striking about Galileo and was considered very offensive at that time, was that he dismissed a lot of data; he was willing to say ''Look, if the data refute the theory, the data are probably wrong.'' And the data that he threw out were not minor.' We read the Galilean record quite differently. Galileo's scientific style had three components to it: first, a deep and thorough skepticism about the established beliefs of the time; second, a belief that REALLY LOOKING at nature—as it is, not as we would like it to be—is essential; and third, a belief that the language in which the principles of nature are written is mathematical in character. These are the Galilean principles that we have attempted to emulate. There is no scientific style that permits one to ignore data; there is only the acknowledgment that one's job is not yet finished. Those are two very different things.

In terminology suggested in Chomsky 1986 and widely adopted since, the analysis proposed here is essentially one of E-LANGUAGE, rather than I-LANGUAGE. While no two writers use these terms in exactly the same way, there is rough agreement that the study of I-language is the study of a capability or a faculty of individual humans who are speakers of a language, while the study of E-language is the analysis of linguistic data that is collected from some naturalistic source (that is, the data in question was not designed and prepared for this experiment, but is rather sampled in some appropriate way from a natural source). We have no objection at all to the study of I-language (indeed, we have been known to actively engage in it, and urge others to do so), but believe that researchers who study E-language are at an advantage with regard to achieving proper scientific standards of linguistic rigor vis-à-vis linguists who study I-language, and this advantage is only growing as improvements in computational and statistical methods become available. Our purpose here has been to demonstrate this proposition in several case studies.

APPENDIX A: SAMPLE CORPUS

Throughout the article, we use the following list of words for our examples: /ban/, /banana/, /bib/, /binis/, /nab/, /saab/, /sans/, and /sins/. These combinations of phonemes were selected with the intent of keeping the phonemic inventory small. Any resemblance with existing words is merely a coincidence.

Table A1 below gives the number of occurrences of each phoneme and each sequence of two phonemes in this corpus. Sequences including a word-initial boundary (denoted by the symbol #) are also listed, since they are used for the spectral clustering of consonants and vowels (see Appendix B).

PHONEME	COUNT	SEQUENCE	COUNT
b	7	aa	1
n	7	ab	2
s	6	an	4
a	8	ba	2
i	4	bi	2
		ib	1
		in	2
		is	1
		na	3
		ni	1
		ns	2
		sa	2
		si	1
		#b	4
		#n	1
		#s	3

TABLE A1. Number of occurrences of phonemes and sequences of phonemes.

APPENDIX B: BUILDING A PHONOTACTIC GRAPH

In this appendix, we introduce a method for constructing a graph in which each node corresponds to a phoneme and the weight of each edge is a measure of the DISTRIBUTIONAL SIMILARITY between two phonemes. The data that we use are frequencies of phonemes in CONTEXTS. For the sake of simplicity, we assume that the context of a phoneme is its left neighbor within a word (including the word boundary symbol #, in the case of the first phoneme of a word), but the model is flexible with regard to what counts as a context. With this definition, the number of occurrences of a phoneme j in a context k in a corpus is equal to the number of occurrences of these two symbols in that order: $Count(kj)$. Thus, on the basis of a corpus with n different phonemes and m different contexts, we may construct a matrix F with n rows and m columns, and store the number of occurrences of phoneme j in context k in the cell at the intersection of the j-th row and k-th

column: $f_{jk} := Count(kj)$. For example, based on the sample corpus given in Appendix A, we construct the following (5×6) matrix F.[24]

$$F = \begin{pmatrix} & \#\_ & b\_ & n\_ & s\_ & a\_ & i\_ \\ b & 4 & 0 & 0 & 0 & 2 & 1 \\ n & 1 & 0 & 0 & 0 & 4 & 2 \\ s & 3 & 0 & 2 & 0 & 0 & 1 \\ a & 0 & 2 & 3 & 2 & 1 & 0 \\ i & 0 & 2 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Our goal is to use F to build the adjacency matrix A of a weighted undirected graph. We do so by means of a two-step method. First, we construct a square matrix W with n rows and n columns, such that the value at the intersection of the i-th row and the j-th column represents the PROBABILITY for phoneme j to occur in the same context as phoneme i (i.e. in a context where phoneme i can also occur). Then, we apply a simple operation to W in order to turn these probabilities into a measure of distributional similarity between phonemes, thus effectively building the desired adjacency matrix A.

In order to construct the matrix W, we first construct two ($n \times m$) matrices, H and V, which result from the horizontal and vertical normalization of F. Thus, H is obtained by dividing the values in each row of F by the sum of these values.

$$H = \begin{pmatrix} & \#\_ & b\_ & n\_ & s\_ & a\_ & i\_ \\ b & 4/7 & 0 & 0 & 0 & 2/7 & 1/7 \\ n & 1/7 & 0 & 0 & 0 & 4/7 & 2/7 \\ s & 3/6 & 0 & 2/6 & 0 & 0 & 1/6 \\ a & 0 & 2/8 & 3/8 & 2/8 & 1/8 & 0 \\ i & 0 & 2/4 & 1/4 & 1/4 & 0 & 0 \end{pmatrix}$$

Similarly, V is obtained by dividing each column of F by its sum.

$$V = \begin{pmatrix} & \#\_ & b\_ & n\_ & s\_ & a\_ & i\_ \\ b & 4/8 & 0 & 0 & 0 & 2/7 & 1/4 \\ n & 1/8 & 0 & 0 & 0 & 4/7 & 2/4 \\ s & 3/8 & 0 & 2/6 & 0 & 0 & 1/4 \\ a & 0 & 2/4 & 3/6 & 2/3 & 1/7 & 0 \\ i & 0 & 2/4 & 1/6 & 1/3 & 0 & 0 \end{pmatrix}$$

Both H and V are transition matrices, from phonemes to contexts and from contexts to phonemes respectively. It is important to notice that, in this framework, the term 'transition' is NOT used to refer to the succession of phonemes in the speech stream, but to a process that is not directly observed in the data, and consists of the selection of a context given a phoneme or the other way round.

The ($n \times n$) matrix W obtains as the result of the matrix product of H and V^T, where V^T denotes the transpose of V, that is, the matrix whose (ordered) columns are the (ordered) rows of V.

$$W = \begin{pmatrix} & b & n & s & a & i \\ b & .4 & .31 & .25 & .04 & 0 \\ n & .31 & .49 & .13 & .08 & 0 \\ s & .29 & .15 & .34 & .17 & .06 \\ a & .04 & .07 & .13 & .5 & .27 \\ i & 0 & 0 & .08 & .54 & .38 \end{pmatrix}$$

This matrix is also a transition matrix, this time from phonemes to phonemes: each cell of W gives the probability of transitioning from one phoneme to another via all possible contexts. It is maximal when the two phonemes have the same distribution (in the mathematical sense) and minimal when they never occur in the same context, that is, when their distributions are complementary.

In spite of this correlation with distributional similarity, however, W is not an actual measure of it, insofar as it is not symmetric. But it has certain properties that entail a natural way of turning it into a symmetric

[24] By convention, we use the underscore symbol _ to distinguish references to (isolated) contexts from references to phonemes.

matrix. Define the STATIONARY probability of phoneme i as the ratio of the total count of i (i.e. the sum of the i-th row of F) to the total count of phonemes in the corpus (i.e. the sum of all the cells of F): $\pi_i := \frac{f_{i\bullet}}{f_{\bullet\bullet}}$. It can be shown (see e.g. Chung 1997) that W is specifically associated with the graph described by the adjacency matrix A, which is defined as as follows.

$$a_{ij} := \pi_i \cdot w_{ij}$$

In other words, A can be easily calculated by multiplying each row i of W by the corresponding stationary probability π_i. In our example, we find that the values of π_i are .22, .22, .19, .25, and .13; multiplying the rows of W by these values results in the following matrix A.

$$A = \begin{pmatrix} & \text{b} & \text{n} & \text{s} & \text{a} & \text{i} \\ \text{b} & .09 & .07 & .05 & .01 & 0 \\ \text{n} & .07 & .11 & .03 & .02 & 0 \\ \text{s} & .05 & .03 & .06 & .03 & .01 \\ \text{a} & .01 & .02 & .03 & .13 & .07 \\ \text{i} & 0 & 0 & .01 & .07 & .05 \end{pmatrix}$$

This is actually the adjacency matrix that we used as an example in §3.2 and represented in Fig. 1. As desired, each row and column of A corresponds to a phoneme, and the weight a_{ij} of the connection between phonemes i and j is a measure of their distributional similarity.[25] Phonemes with similar distributions are strongly connected, whereas phonemes with dissimilar distributions are weakly or not connected. As we have seen in §3.2, the application of spectral clustering to the adjacency matrix that was just constructed results in a partitioning of phonemes into classes that correspond well with vowels and consonants.[26]

REFERENCES

BAVAUD, FRANÇOIS, and ARIS XANTHOS. 2005. Markov associativities. *Journal of Quantitative Linguistics* 12.123–37.

BELKIN, MIKHAIL, and JOHN A. GOLDSMITH. 2002. Using eigenvectors of the bigram graph to infer morpheme identity. *Proceedings of the sixth workshop of the ACL Special Interest Group in Computational Phonology*, ed. by Michael Maxwell, 41–47. East Stroudsburg, PA: Association for Computational Linguistics.

BEZDEK, JAMES C. 1981. *Pattern recognition with fuzzy objective function algorithms*. New York: Plenum.

BIGGS, NORMAN. 1993. *Algebraic graph theory*. 2nd edn. Cambridge: Cambridge University Press.

BLOOMFIELD, LEONARD. 1933. *Language*. New York: H. Holt and Company.

CHARNIAK, EUGENE. 1993. *Statistical language learning*. Cambridge, MA: MIT Press.

CHOMSKY, NOAM. 1986. *Knowledge of language*. New York: Praeger.

CHOMSKY, NOAM. 2000. An interview on minimalism. Online: http://www.ling.ed.ac.uk/~s0450647/docs/interview_Chomsky.pdf.

CHUNG, FAN R. K. 1997. *Spectral graph theory*. Providence, RI: American Mathematical Society.

DOWMAN, MIKE. 2008. Minimum description length as a solution to the problem of generalization in syntactic theory. Tokyo: University of Tokyo, MS. Online: http://www.ling.ed.ac.uk/~mdowman/mdl-and-generalization.pdf.

ELLISON, T. MARK. 1991. The iterative learning of phonological constraints. Crawley: University of Western Australia, MS.

ELLISON, T. MARK. 1994. *The machine learning of phonological structure*. Crawley: University of Western Australia dissertation.

[25] Notice that, in general, the elements on the main diagonal of A are NOT constant: $a_{11} \neq a_{22} \ldots \neq a_{nn}$. Indeed, under this scheme, the similarity of a phoneme with itself depends on its similarity with all other phonemes.

[26] For mathematical reasons that are beyond the scope of this article, the actual matrix that undergoes the spectral decomposition discussed in §3.2 is a NORMALIZED version of A, defined as $C := \Pi^{-1/2} A \Pi^{-1/2}$, where Π stands for the matrix containing the stationary probabilities of phonemes on the main diagonal and zeros everywhere else (see e.g. Bavaud & Xanthos 2005 for details on this).

ELLISON, T. MARK. 2001. Induction and inherent similarity. *Similarity and categorization*, ed. by Ulrike Hahn and Martin C. Ramscar, 29–49. Oxford: Oxford University Press.
FINCH, STEVEN. 1993. *Finding structure in language*. Edinburgh: University of Edinburgh dissertation.
GOLDSMITH, JOHN A. 2001. The unsupervised learning of natural language morphology. *Computational Linguistics* 27.153–98.
GOLDSMITH, JOHN A. 2009. The syllable. *The handbook of phonological theory*, vol. 2, ed. by John Goldsmith, Jason Riggle, and Alan Yu. Oxford: Blackwell, to appear.
GOLDSMITH, JOHN A., and JEREMY O'BRIEN. 2006. Learning inflectional classes. *Language Learning and Development* 2.219–50.
GOLDSMITH, JOHN A., and JASON RIGGLE. 2007. Information theoretic approaches to phonology: The case of Finnish vowel harmony. Chicago: University of Chicago, MS. Online: http://hum.uchicago.edu/~jagoldsm//Papers/boltzmann.pdf.
GOLDSMITH, JOHN A., and ARIS XANTHOS. 2008. Three models for learning phonological categories. Technical report 2008-8. Chicago: Department of Computer Science, University of Chicago.
GOLDWATER, SHARON. 2006. *Nonparametric Bayesian models of lexical acquisition*. Providence, RI: Brown University dissertation.
GUY, JACQUES. 1991. Vowel identification: An old (but good) algorithm. *Cryptologia* 15.258–62.
HAYES, BRUCE, and COLIN WILSON. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry* 39.379–440.
JELINEK, FREDERICK. 1997. *Statistical methods for speech recognition*. Cambridge, MA: MIT Press.
KANNAN, RAVI; SANTOSH VEMPALA; and ADRIAN VETTA. 2000. On clusterings: Good, bad, and spectral. *Proceedings of the 41st Annual Symposium on the Foundation of Computer Science*, 367–80. Washington, DC: IEEE Computer Society.
PEPERKAMP, SHARON; ROZENN LE CALVEZ; JEAN-PIERRE NADAL; and EMMANUEL DUPOUX. 2006. The acquisition of allophonic rules: Statistical learning with linguistic constraints. *Cognition* 101.B31–B41.
PIKE, KENNETH. 1943. *Phonetics*. Ann Arbor: University of Michigan Press.
POWERS, DAVID M. W. 1991. How far can self-organization go? Results in unsupervised language learning. *Proceedings of AAAI Spring Symposium on Machine Learning of Natural Language and Ontology*, ed. by David M. W. Powers and Larry Reeker, 131–37.
POWERS, DAVID M. W. 1997. Unsupervised learning of linguistic structure: An empirical evaluation. *International Journal of Corpus Linguistics* 2.91–132.
REICHENBACH, HANS. 1938. *Experience and prediction*. Chicago: University of Chicago Press.
RISSANEN, JORMA. 1989. *Stochastic complexity in statistical inquiry*. Singapore: World Scientific.
SAFFRAN, JENNY R.; RICHARD N. ASLIN; and ELISSA L. NEWPORT. 1996. Statistical learning by 8-month-old infants. *Science* 274.1926–28.
SCHIFFERDECKER, G. 1994. Finding structure in language. Karlsruhe: University of Karlsruhe master's thesis.
SILVERSTEIN, MICHAEL (ed.) 1971. *Whitney on language: Selected writings of William Dwight Whitney*. Cambridge, MA: MIT Press.
SUKHOTIN, BORIS V. 1962. Eksperimental'noe vydelenie klassov bukv s pomoščju EVM. *Problemy strukturnoj lingvistiki* 234.189–206.
SUKHOTIN, BORIS V. 1973. Méthode de déchiffrage, outil de recherche en linguistique. *T. A. Informations* 2.1–43.
TESAR, BRUCE. 1998. An iterative strategy for language learning. *Lingua* 104.131–45.
TRUBETZKOY, NICOLAI S. 1969. *Principles of phonology*. Berkeley: University of California Press.
VAN DER HULST, HARRY, and JEROEN VAN DE WEIJER. 1995. Vowel harmony. *The handbook of phonological theory*, ed. by John A. Goldsmith, 495–534. Oxford: Blackwell.
WARD, JOE H. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association* 58.236–44.

XANTHOS, ARIS. 2008. *Apprentissage automatique de la morphologie: Le cas des structures racine-schème*. (Sciences pour la communication 88.) Berne: Peter Lang.

Goldsmith
University of Chicago
Departments of Linguistics and Computer Science
1010 East 59th St.
Chicago, IL 60637
[goldsmith@uchicago.edu]

Xanthos
University of Lausanne
Department of Computer Science and Mathematical Methods
Anthropole
CH-1015 Lausanne, Switzerland
[aris.xanthos@unil.ch]

[Received 8 January 2007;
revision invited 29 July 2007;
revision received 10 February 2008;
accepted 9 August 2008]