Learning phonological categories

John Goldsmith and Aris Xanthos

December 7, 2006

Abstract

This paper 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 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, and offer an evaluation of the results.

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, pre-generative American phonology focused on questions of phonotactics, lacking the tools to treat alternations in depth, and generative phonology (and post-generative phonology) has focused on alternations, lacking the tools to deal with a detailed study of phonotactics.

In this paper, we approach the general problem of inference (or acquisition) of phonotactics, and consider the usefulness of three algorithmic styles of analysis to three questions regarding the overall phonotactics, and the phonological categories that phonotactics presuppose (consonants, vowels, etc.). These questions are: (1) Given a sample of data (transcribed symbolically) from a language,

can we infer which segments are vowels and which are consonants? (2) 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? (3) Can we draw inferences about the organization of segments into syllabic structure?

We have chosen these 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 syllable structure automatically would doutbless strike any working phonologist as a non-trivial task–a highly non-trivial task.

It is not our goal in this paper to engage in an ideological battle, but it would serve no purpose to ignore the simple fact that the approach which 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. The first is for purely historical reasons–because it was one of the first algorithms proposed to solve a phonological problem. The second 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 (HMMs). The three methods are these:

(i) The first, due to Sukhotin (1962), is one of the earliest algorithm 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 Section 2 below.

(ii) The second system is based on spectral graph theory, a relatively new mathematical field which 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. 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 due to the fact that 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 is done in Belkin & Goldsmith 2002, for example).

(iii) The third system employs hidden Markov models (HMMs) in order to develop automatically 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 infers consistently 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 paper 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 which 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) which gives an accurate description of the language which generated the input data.

Looking ahead, what we will find 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 vowel 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. Spectral methods of analyzing similarities do a relatively good job of distinguishing vow-

els 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 models) work remarkably well on detecting the consonant/vowel distinction, and the vowel harmony system of Finnish, and sheds some interesting light on the sonority hierarchy and syllable structure in 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 of this work includes the pre-generative work, such as that by Fischer-Jörgensen and Householder, which 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.[1]

A second generation of work on distribution classification of phonological segments grew out of computational linguistics, by 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 30 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 for 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 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

[1] In Appendix A below, we discuss this material in greater detail.

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 which 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 regarding 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 regarding 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 vowels, and the other consonants.

Ellison explored the usefulness of Minimum Description Length analysis (henceforth, *MDL* analysis) for the problem of distinguishing classes of phonological segments (see Ellison 1992, 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 which is *not* in the regularities, and a method to *measure* quantitatively both the model which 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 which increasingly hops only in favor of changes that increase the evaluation metric, eventually stopping because there is no change which can be found which 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, to see 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 unbeknownst to each other, over the last fifteen years, in publications such as Ellison (1992), Powers (1997), Ellison (2001), Goldsmith (2001), Goldsmith & O'Brien (2006), Goldwater (2006), Dowman (ms.), as well as others cited therein.[2]

[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.

3 Vowels and consonants

In this section, we describe and evaluate three approaches to the problem of identifying the class of vowels and consonants in a distributional way: an approach described by 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. We shall see that this order of presentation corresponds to increasing ability to correctly model the data.

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 in this paper, 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 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 $Count(\texttt{ba})$ 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 $r_{ij} := Count(p_i p_j) + Count(p_j p_i)$. R is thus a symmetric matrix, i.e. 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 ($r_{ii} := 0$).

[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)

For instance, given the sample corpus described in Appendix B (p. 45), we find an inventory of 5 phonemes $P = \{\texttt{b}, \texttt{n}, \texttt{s}, \texttt{a}, \texttt{i}\}$, so $n = 5$. Using the frequencies of sequences of two phonemes[4] reported in table 14 (p. 46), we may calculate the components of R as indicated: $r_{11} = 0$ by convention, $r_{12} = Count(\texttt{bn}) + Count(\texttt{nb}) = 0$, ..., $r_{15} = Count(\texttt{bi}) + Count(\texttt{ib}) = 3$, and so on. We obtain the following (5×5) matrix:

$$R = \begin{pmatrix} 0 & 0 & 0 & 4 & 3 \\ 0 & 0 & 2 & 7 & 3 \\ 0 & 2 & 0 & 2 & 2 \\ 4 & 7 & 2 & 0 & 0 \\ 3 & 3 & 2 & 0 & 0 \end{pmatrix} \tag{1}$$

Sukhotin's algorithm begins by labelling 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 labelled 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 labelled 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 should be positive: the larger, the better. This difference is precisely the score $v(p_i)$ assigned by Sukhotin's algorithm.

During the initialization step, since all phonemes are assumed to be consonants, their frequency next to a vowel is zero, so the difference between their frequency next to a consonant and their frequency next to a vowel is simply equal to their frequency, irrespective of the context. For each phoneme p_i, this turns out to be the sum of the values found on the i-th row of R. Here and throughout, we use 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.

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

Corpus	#words (types)	#phones (types)	#phones (tokens)
English	58,156	54	386,421
French	21,768	36	147,146
Finnish	44,040	27	466,134

Table 1: Basic facts about the corpora

With this notation, we can write: $v(p_i) := r_{i\bullet}$. In our example, we find that $v(\texttt{b}) = 4{+}3 = 7$, $v(\texttt{n}) = 2{+}7{+}3 = 12$, $v(\texttt{s}) = 2{+}2{+}2 = 6$, $v(\texttt{a}) = 4{+}7{+}2 = 13$, and $v(\texttt{i}) = 3 + 3 + 2 = 8$. The phoneme `a`, which has the highest score, is thus labelled as a vowel, which matches the assumption that the most frequent phoneme in a language is a vowel.

The score $v(p_i)$ assigned to the remaining phonemes must be updated to reflect the new composition of the sets of vowels and consonants. For each phoneme p_i (other than `a`—it is effectively out of the game now), this can be done by simply subtracting 2 times the value found at the intersection of the i-th row of R and the column that corresponds to the phoneme that was just labelled as a vowel.[5] In our case, the column corresponding to phoneme `a` is the fourth one; by subtracting its values from the scores of the remaining phonemes, we update the scores as $v(\texttt{b}) = 7 - (2 \cdot 4) = -1$, $v(\texttt{n}) = 12 - (2 \cdot 7) = -2$, $v(\texttt{s}) = 6 - (2 \cdot 2) = 2$, and $v(\texttt{i}) = 8 - 0 = 8$.

The phoneme with the highest score is now `i`. It is labelled as a vowel and the scores of the remaining phonemes are updated accordingly: $v(\texttt{b}) := -1 - (2 \cdot 3) = -7$, $v(\texttt{n}) := -2 - (2 \cdot 3) = -8$, and $v(\texttt{s}) := 2 - (2 \cdot 2) = -2$. Since there are no more positive scores, the algorithms deduces that it has found all the vowels, and it returns the following labelling: the set of vowels is $\{\texttt{a}, \texttt{i}\}$, and the set of consonants is $\{\texttt{b}, \texttt{n}, \texttt{s}\}$.

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. The 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.

Table 2 (p. 10) shows the classification of vowels and consonants performed

[5] The factor 2 stems from the fact that the frequency of the newly discovered vowel must be added to the total frequency of vowels *and* subtracted from the total frequency of consonants.

[6] The former is encoded in ArpaBet format (see Jurafsky & Martin 2000, or any of a number of locations on the internet, for details) and the latter uses standard IPA.

English		French		Finnish	
Consonants	Vowels	Consonants	Vowels	Consonants	Vowels
T	AH0	t	ʁ	t	i
K	R	l	a	s	a
D	IH0	s	i	n	e
P	S	n	e	l	u
Z	L	k	ə	k	o
B	N	m	o	m	ä
EH1	ER0	d	ɑ̃	r	y
IH1	M	p	ɛ	v	ö
G	W	b	y	p	q
IY0	HH	v	ɔ̃	h	
V	EY1	z	ɛ̃	j	
SH	Y	f	ɔ	d	
AO1	ER1	ʒ	u	b	
AE1	OW0	g	ø	g	
AA1	AY0	j	œ	f	
NG	EY0	ʃ	œ̃	c	
IY1	CH	ɲ		w	
AY1	F	w		x	
UW1	DH	h			
JH	AW0	ɳ			
AH1	OY0				
AA0					
OW1					
EH0					
AW1					
UW0					
AO0					
AE0					
TH					
UH1					
OY1					
UH0					
ZH					

Table 2: Results of Sukhotin's algorithm on three natural language corpora

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 that it is misclassified as a vowel in the first place. However this does not affect the classification of the remaining phonemes, all of which are correctly labelled.[7] In Finnish, all consonants and vowels are correctly identified, with the exception of the rare symbol *q*. A closer look to the contexts where

[7] Notice that there is no distinction between /a/ and /ɑ/ in this corpus; /h/ denotes the *h-aspiré*, which is treated as a phoneme in this data set.

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 15 out of 18 occurrences in non-initial position; similarly, it is followed by a consonant in 11 out of 16 occurrences in non-final 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 on top of the *Vowels* column, and the less vowel-like symbols are on 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 the most vowel-like consonant, rather than the other way round.

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 labelled as vowels (10/21) are actually consonants, and the proportion of real vowels misclassified as consonants is even higher (20/33). However, it appears that the primary reason for the poorness of the results lies in the unusual method used to represent stress level. In the original form, vowels are transcribed with a label that consists of their quality followed by their stress level (e.g., AE1). There is no connection made between (for example) the vowel AE0 and the vowel AE1 – 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 which are much better. In particular, the only divergence with regard to a phonetic classification is that R is misclassified as a vowel. Similarly to /ʁ/ in French, R is one of the most frequent consonants in this corpus; N and T are more frequent, but once the two first vowels (AH and IH) have been identified, and their cooccurrences next to other phonemes have been subtracted, the phoneme with the highest score is R.

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 implies that

[8] This is where our implementation of the algorithm differs from Sukhotin's: we keep *ordering* phonemes after the 0 threshold, so that we can also evaluate their typicality as consonants.

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 which will become clear 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.

3.2.1 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 edges, and so we say that the graphs are *undirected*. However, the edges of our graphs are *weighted*, which means that their edges are associated with a real number; such a weight must be non-negative. 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 $V = \{\texttt{b}, \texttt{n}, \texttt{s}, \texttt{a}, \texttt{i}\}$ with weighted edges.[9]

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

[9]Notice that, on 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*.

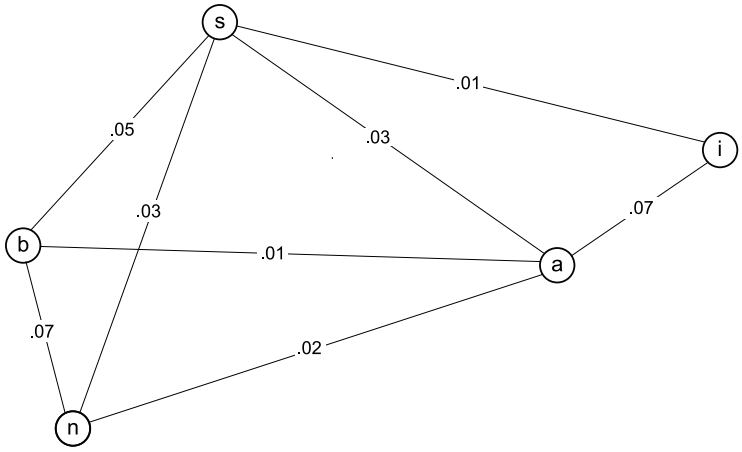

Figure 1: A sample weighted undirected graph

matrix of the graph represented on Figure 1 is:[10]

$$A = \begin{pmatrix} .09 & .07 & .05 & .01 & 0 \\ .07 & .11 & .03 & .02 & 0 \\ .05 & .03 & .06 & .03 & .01 \\ .01 & .02 & .03 & .13 & .07 \\ 0 & 0 & .01 & .07 & .05 \end{pmatrix} \tag{2}$$

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. We will employ the "dot-notation," defined above, and so we may define the degree of node i, which is expressed d_i, as $a_{i\bullet}$. 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.

[10] The elements on the main diagonal represent *loops*, i.e. edges connecting a node to itself; for the sake of readability, these were not represented on Figure 1.

3.2.2 Graph partitioning

We will build a graph below (see section 3.2.3 and Appendix D) 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 which will 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 will 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*, i.e. the sum of the weights of edges connecting nodes *between* the two groups:

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

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

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

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 Figure 1.

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 by Kannan *et al.* (2000) (see Appendix C for more details on this). In our example, this revised criterion leads to the same partition $S = \{\mathtt{b}, \mathtt{n}, \mathtt{s}\}, T = \{\mathtt{a}, \mathtt{i}\}$, with minimal conductance $\phi(S,T) = .07/\min(.63, .39) = .18$ (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. The spectral theorem is a fundamental result in linear algebra whose details are beyond the scope of this paper; in the context of graph partitioning, it basically enables us to summarize the information contained in an $(n \times n)$ adjacency matrix into a single vector of n real numbers, called the *second eigenvector* (or *Fiedler vector*) of the graph. In effect, this vector assigns a single number to each node in the graph, so that they can be represented as

Figure 2: Second eigenvector of the graph represented in Figure 1.

points on a single dimension (see Figure 2).[11]

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 Figure 2 is only an approximate representation of the graph in Figure 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 4 partitions (see Figure 2): i) $S = \{\mathtt{b}\}, T = \{\mathtt{n}, \mathtt{s}, \mathtt{a}, \mathtt{i}\}$, ii) $S = \{\mathtt{b}, \mathtt{n}\}, T = \{\mathtt{s}, \mathtt{a}, \mathtt{i}\}$, iii) $S = \{\mathtt{b}, \mathtt{n}, \mathtt{s}\}, T = \{\mathtt{a}, \mathtt{i}\}$, and iv) $S = \{\mathtt{b}, \mathtt{n}, \mathtt{s}, \mathtt{a}\}, T = \{\mathtt{i}\}$. We have seen previously that partition iii) has minimal conductance; the important point here is that is was indeed "pre-selected" by the spectral approach, contrary to the vast majority of less optimal partitions (11 out of 15, in this case). This illustrates the efficiency of spectral clustering as a way of quickly searching the space of possible partitions of a graph.

[11] To be precise, the vector represented in Figure 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 Appendix D (and in particular note 27, p.51) for more details.

3.2.3 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. As we will see, 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 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. Such a specification is given in Appendix D, so we will remain at a more intuitive level of explanation here. 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 will typically be 0 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 such as the one described in Appendix D 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.

We have applied this procedure to build a phonotactic graph for each of the three corpora used in the previous section. Table 4 (p. 19) 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

[12] In this table, the ordering of phonemes reflects their ordering on the (normalized) Fiedler

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 that four consonants are misclassified as vowels: Y, W, R, and Z. Classifiying glides with vowels seems to be a consistent behaviour of the spectral method, as it also occurs for French (more on this below). Although R and Z are relatively frequent 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 Appendix C) rather than its numerator: in other words, that they help balancing 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). The results for Finnish are exactly identical to those of Sukhotin's algorithm, i.e. the symbol q is misclassified as a vowel (see section 3.1).

Overall, it seems that the spectral approach performs considerably better than Sukhotin's algorithm. The spectral approach's tendency to label glides as vowels could probably be fixed by modifying the definition of context to take into account the following phoneme as well, which is the case in Sukhotin's algorithm. Insofar as the spectral method's classification of English phonemes is incomparably 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 (HMMs)

3.3.1 Introduction

The third method which we have explored poses the problem of phone categorization in terms of a natural optimization problem: suppose we construct

vector, i.e. 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 (UH, OY, IY, AH) are actually located next to one another on the Fiedler vector.

English		French		Finnish	
Cluster 1	Cluster 2	Cluster 1	Cluster 2	Cluster 1	Cluster 2
UW0	NG	ə	ñ	ä	x
UH1	N	ɔ̃	z	a	n
UH0	ZH	œ	n	o	h
UW1	DH	ɔ	f	e	r
ER1	V	i	b	u	v
OY0	K	ø	v	q	c
OY1	G	w	d	y	l
AY1	JH	y	g	i	w
IY1	SH	e	k	ö	m
IY0	M	ɥ	p		f
ER0	L	ɛ	ʃ		s
AO1	TH	u	ʒ		d
IH1	F	o	s		j
EY1	B	a	m		p
AY0	CH	ã	h		b
EH1	S	œ̃	l		k
AA1	P	j	t		g
AW1	D	ɛ̃	ʁ		t
OW1	HH				
AH0	T				
AH1					
EY0					
AE1					
AO0					
OW0					
IH0					
AA0					
Y					
AE0					
EH0					
AW0					
W					
R					
Z					

Table 4: Results of the spectral method on three natural language corpora

a finite state device with a small number of states (2 states, in most of the cases that we will 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, there is only 1 other state). We desire to find the assignment of probability distributions for these two functions (emission distribution and

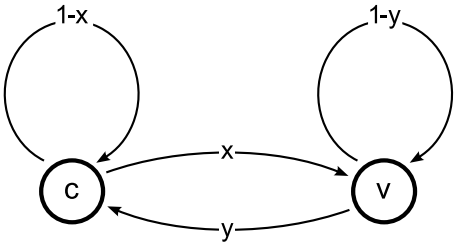

Figure 3: A simple 2-state hidden Markov model

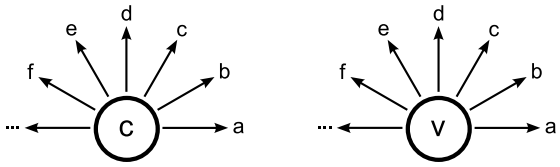

Figure 4: The states in the HMM

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.

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, on the other hand, 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. As we will see, 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.

3.3.2 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 50 iterations.[14] At that point, we can observe three aspects of the results: the emission probabilities, the transitions probabilities, and the common convergence despite random initial assumptions.

(1) 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 the phonemes of English are identified by their ArpaBet representation, as above. A positive value indicates a phoneme which the network prefers to generate in State 1, while a negative value indicates a phoneme which 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 0, or close to it), and similarly for "-999". The segments are naturally divided into two groups, depending as this 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 log 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.

[14] The relevant files can be found at `http://hum.uchicago.edu/~jagoldsm/Papers/2006LearningPhonoCategories/English2StatesPhonemes/`.

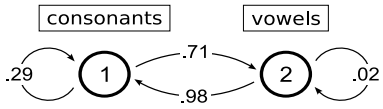

Figure 5: English: 2-state FSA

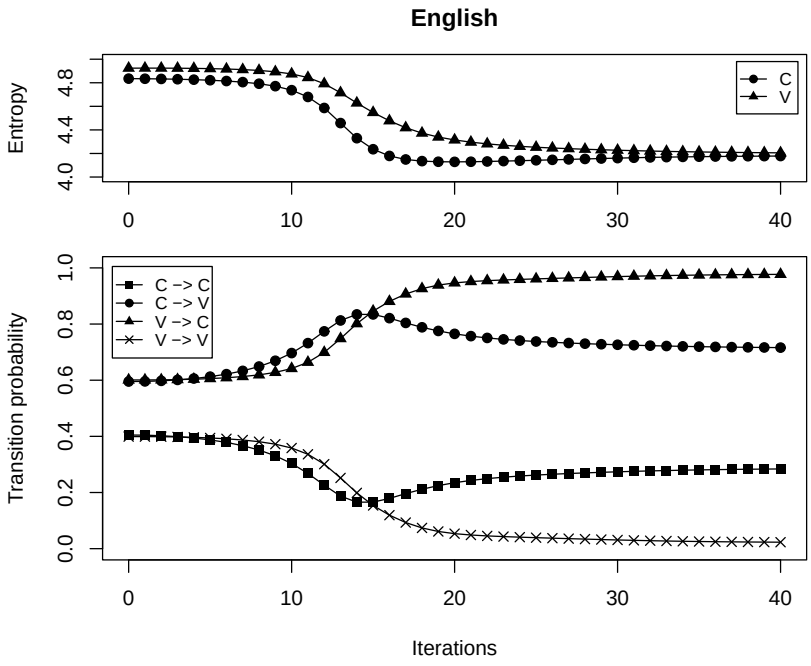

Figure 6: English transitions

ArpaBet	Log ratio
DH	-999
NG	-999
W	-999
N	-999
L	-999
HH	-999
SH	-999
R	-999
M	-999
V	-999
ZH	-999
JH	-999
B	-999
Y	-999
F	-999
G	-829
K	-576
CH	-361
TH	-5.19
P	-4.37
D	-3.95
S	-2.75
T	-2.20
Z	-1.37

ArpaBet	Log ratio
UW0	2.22
ER0	2.30
IY0	2.31
AW0	2.32
AY0	2.83
OW0	3.93
EY0	4.99
AY1	5.11
OY1	5.81
IY1	7.39
OW1	12.7
AW1	275
EY1	262
OY0	263
UW1	999
AH0	999
EH0	999
AE0	999
ER1	999
AA0	999
IH0	999
AE1	999
AO0	999
EH1	999
AA1	999
AO1	999
IH1	999
AH1	999
UH1	999
UH0	999

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

(2) We can inspect the *transition* probabilities for the two states. We can do this is several ways. First, we can consider the final steady-state values of the four state transition probabilities, as shown in Table 6, and the same information is displayed more graphically in Figure 5. Second, 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 Figure 6. We present graphically the evolution of the transition probabilities over the course of the first 40 iterations during the learning phase.[15]

[15] A moment's study of the data displayed in Table 5 leads one to the question as to *why* there seems to be a span of vowels (UW0 ER0 IY0 AW0 OW0 EY0 AY1 OY1 IY1) and of consonants(TH P D S T Z) whose log ratio is surprising close to zero. There appear to be two

	To State 1	To State 2
From State 1	.29	.71
From State 2	.98	.02

Table 6: Transition probabilities, 2-state HMM for English

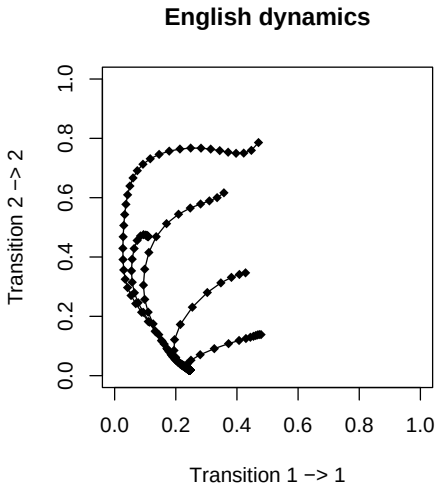

Figure 7: 5 paths to the learning of English transitions. x axis is prob (State 1→State 1), y axis is prob (State 2→State 2). All movement is downward and to the left.

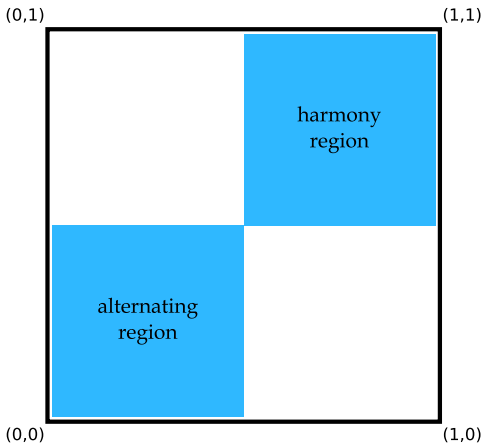

Figure 8: Phase space, defined by probability of each state transitioning to itself

(3) 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 2-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 will 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]

3.3.3 Alternating and harmony systems

When the transitions 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 "C" and "V" once we inspect the

separate answers for these questions. The data which we have used, a CMU wordlist widely available on the Internet, includes a number of words in which two vowels appear adjacent to each other: e.g., overarching = OW1 V ER0 AA1 R CH IH0 NG, biotic = B IY0 AA1 T IH0 K. This appears to be the reason why a number of diphthongs marked with 0stress 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 3-state HMMs analyze this data.

[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 9: French transitions

identity of the segments being generated by them). We will see, when we analyze vowel harmony data in parallel fashion below, the system will reach equilibrium at a point in a different quadrant, one where the probability of the transitions from one state back to itself is close to 1; 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.

3.3.4 Observing results for French

Turning now to a corpus of French, we find essentially the same results; the results are given in Table 7 and Table 8 after 1200 iterations. Figure 9 presents the transition data graphically. Figure 10 illustrates the early and 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 three different initial random values. Again, as in English, the end point of the learning is a spot in the alternating region of phase space.

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.

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 2-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* the linguist defines, at a high level of abstraction, his goal to be the development of a model that maximizes the probability of the data, then if he chooses 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 will *emerge* from this distributional task are the segments

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
ñ	999	œ̃	-5.53
v	999	e	-4.93
ʃ	999		
h	999		
ɥ	999		
w	999		
j	999		
z	999		

Table 7: Phones and the log ratios of their emissions, comparing the two states of the HMM for French.

	To State 1	To State 2
From State 1	.23	.77
From State 2	.98	.02

Table 8: Transitions probabilities, 2-state HMM for French

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 vowels and consonants (distinct from, and largely simpler, than that of Ellison, discussed above). These two categories are doubtless the most important and fundamental category in all of phonology. What question, or questions, come 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

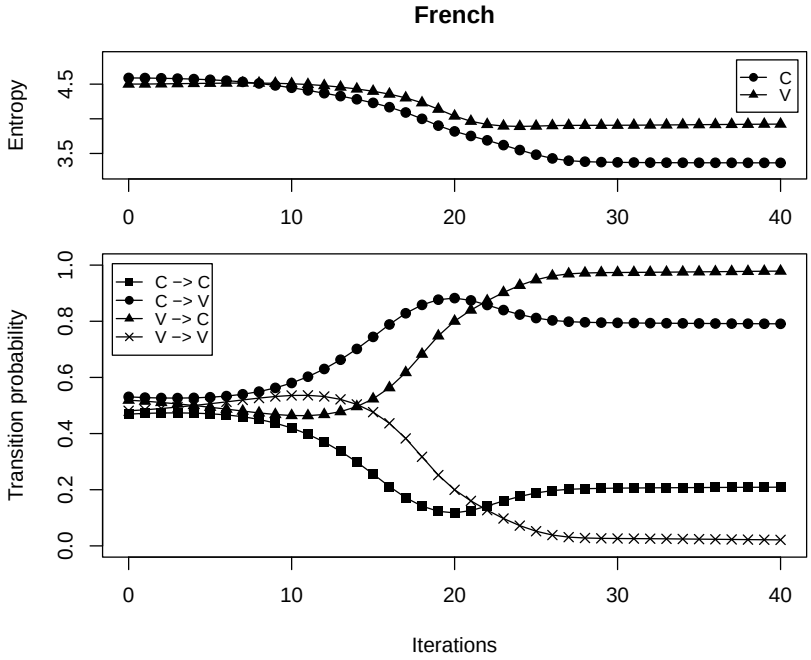

Figure 10: French transitions

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 (typically 2) of subsets 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 and van de Weijer (1995) for an overview of vowel harmony systems.

The task of identifying vowel harmony is thus a problem of category discovery. Our question then is this: is there an algorithm which takes as its input

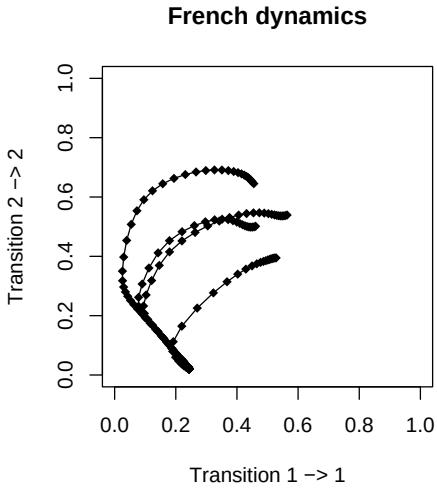

Figure 11: Dynamics of learning French c/v. All movement is downward to the left.

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 to use as our training set. The traditional account of Finnish is that there are two neutral vowels, 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 $\ddot{o}$, $\ddot{a}$ 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 sections 3.1 and 3.2, we have 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 alternat-

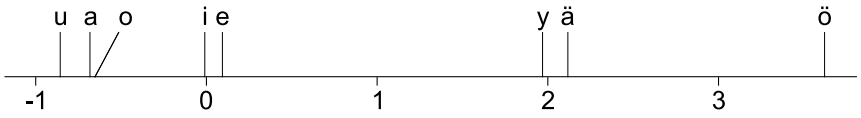

Figure 12: Second eigenvector of the graph of Finnish vowels.

ing 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.

Thus we have applied the spectral method introduced in section 3.2 and Appendix D 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.[17] As shown on 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 $\ddot{a}$ and another cluster containing only the vowel $\ddot{o}$.

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 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.

[17] Recall that the clustering algorithm that we use invariably returns two categories. From Figure 12, it may seem that neutral vowels are more similar to back vowels. However, the spectral representation just serves as a filter that discards a large proportion of possible partitionings; ultimately, the crucial criterion is the *conductance* (see Appendix C) associated with each partitioning, and not the distances induced by the spectral projection.

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 9: Log ratios of emission probabilities for Finnish vowels.

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 consonant-vowel 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 consonant/vowel distinction described in 3.3 above. The method is simplicity itself: we train an HMM which 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 which 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.

We find that the vowels in our Finnish corpus are quickly and easily distributed along a single dimension, as in Table 9. 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

	To Front Vs	To Back Vs
From Front Vs	.90	.10
From Back Vs	.03	.97

Table 10: Transition probabilities, 2-state HMM for Finnish vowel harmony

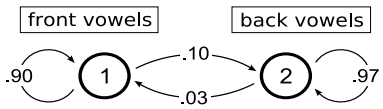

Figure 13: Finnish vowel transitions

with only back vowels (or a mixture of back vowels and neutral vowels) in State 2. The emission and transition results are given in Table 9 and Table 10, after 1,000 iterations of training. Table 9 shows the separation of the vowels into two groups, and Table 10 shows that this is a harmony system, by virtue of the fact that the transitions 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 in this paragraph.

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 non-zero 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

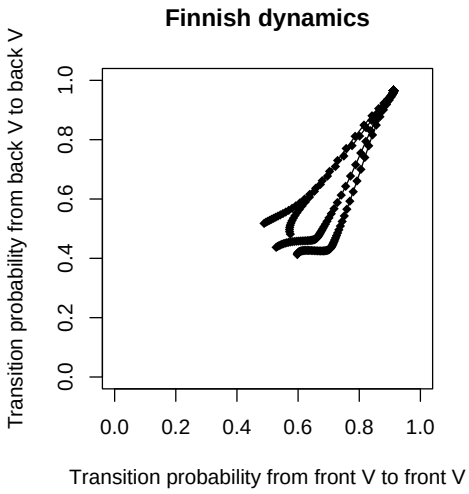

Figure 14: Finnish transition evolution. All movement is upward to the right.

generated by the front vowel state, while a neutral vowel in a back vowel word will be generated by the back vowel state.

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 system that are in the upper right quadrant are naturally labeled as *harmony* system: once in a given state, they prefer to remain in that state; see section 3.3.3.

5 Learning aspects of syllable structure

5.1 Syllable structure as maximum likelihood

The discussion in section 3.3 assumed without discussion that we would divide the segments of a language into two categories, consonants and vowels. However,

	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 11: Transitions probabilities, 3-state HMM for French

there is no reason 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 which expresses the next most important statistical dependency in the data, compared to the 2-state model– but what would that be? The 2-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 solely responsible for generating the second element of an onset cluster, while the other was responsible for generating all other consonants. In this section, we will describe that result, and suggest some areas for future research.

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 is displayed in Table 11. 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

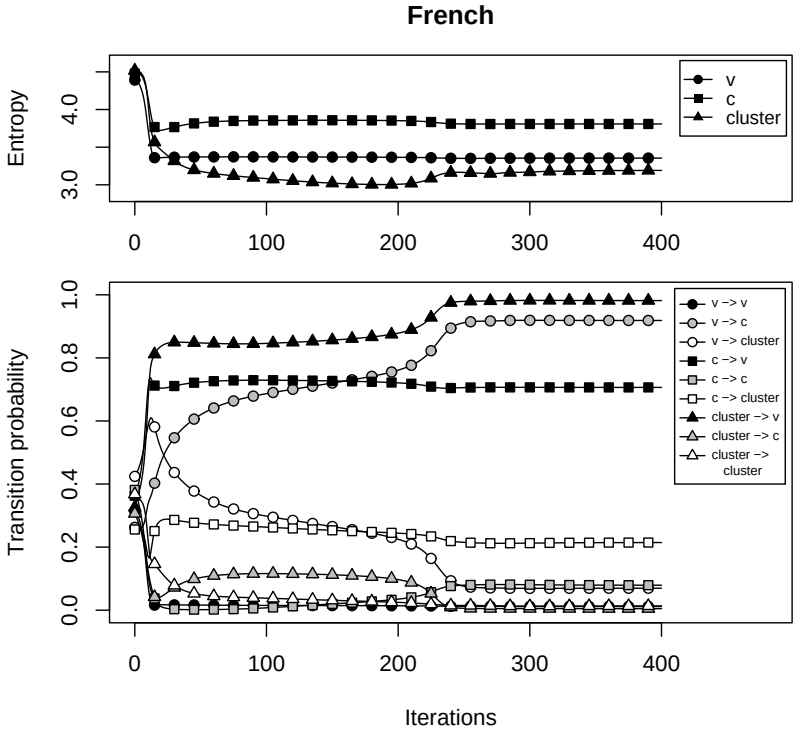

Figure 15: French 3 states learning dynamics for transition and entropy

quiescence up to iteration 200, followed by 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. However, 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 second member of an onset cluster, and it *always* transitions to the vowel state. For mnemonic purposes, we will 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 6-dimensional space (a hypercube), then we may describe that 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)),

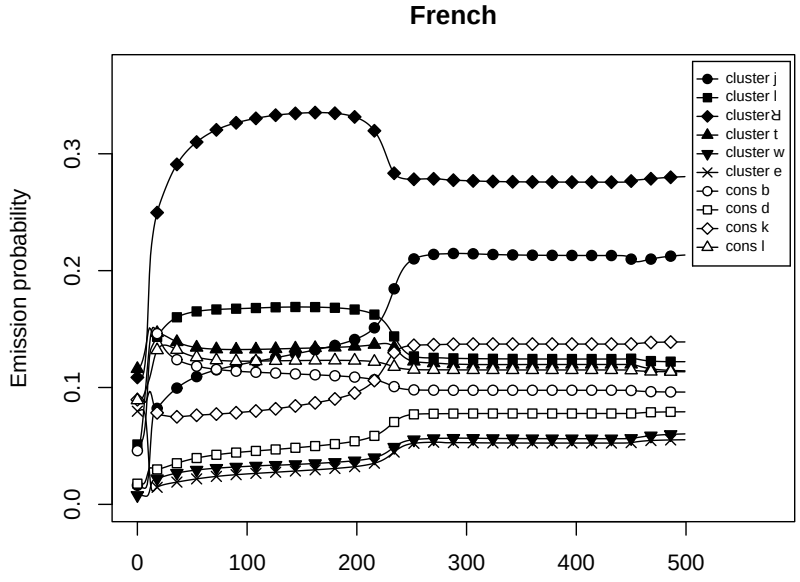

Figure 16: Crucial emission changes during French 3 states learning dynamics

which in some sense is suggestive of a categorical, rather than a gradient, analysis.[18] 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 11 and Table 12. We omit segments whose emission probabilities fall below 0.01. See Figure 17 for a partial graphical summary.

This model generates sequences like /abʁa/ and those like /aʁba/ in different ways (the logic of the situation is parallel to that which we 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 calculations are given explicitly in Table 13. The path through the HMM which produces the sequence /abʁa/ with maximum probability is the one which emits those symbols by following the sequence

[18]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 6 parameters, and hence a specification of the transition probabilities can be thought of as specifying a point in a part of a 6-dimensional space.

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
ã	.06	p	.07	e	.06
y	.05	m	.06	m	.03
ə	.04	d	.06	ɥ	.02
ø	.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 12: Emission probabilities, 3-state HMM for French

of states *1 2 3 1*, while the path which 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, i.e 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 $3 \rightarrow 1$ or $3 \rightarrow 2$, since those transition probabilities are close to 0. We have chosen this example to illustrate the point we noted above, that state 3 is effectively dedicated to generating the second element of an onset cluster.

Needless to say, a range of further cases should be studied. We would predict, for example, that a language which 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 which, 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.

6 Discussion

In this final section, we wish to address three questions. The first is rather general: what kind of work is this? The second and third are more serious. What are the consequences of this approach for our understanding of universal grammar? What are the consequences of this approach for what we take the

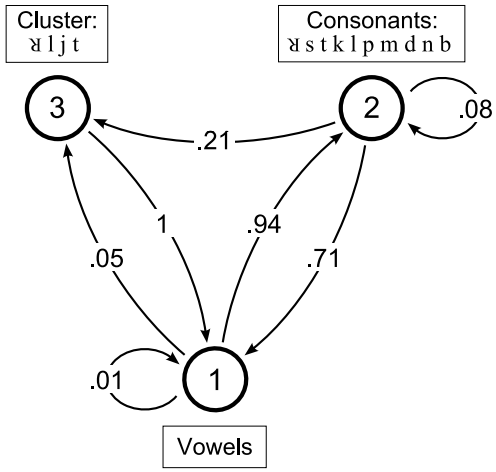

Figure 17: Three states for generating French strings

object of study to be in linguistics?

What kind of work is this? This is a question several of our colleagues, a bit quizzical, have asked: Is this *phonology*? Of course the answer is *yes*; it is certainly not phonetics nor morphology nor syntax, and it tries to answer questions that are those of the phonologist. But something obviously lies behind the question, and there is a chance that the reader is asking himself a question along these lines. So we will try to make explicit what may seem odd about the present account.

First, *it may seem odd to use numbers, especially non-integral numbers,* in a phonological account. Answer: There appears to us to be no interesting response to this observation. Of course, there *have* been phonological accounts that employ non-integral numbers (e.g., Goldsmith 1994, to mention just one), but even if there had not been, only someone who thinks (as we do not) that we are within striking distance of the Final Theory of Phonology could draw substantial conclusions from this observation. Some models use numbers; some do not.

Second, it seems odd that *there are no phonological representations* anywhere. We have talked about paths through automata generating certain strings, but we have not used phonological representations in any direct way. Answer: As working phonologists, it seems to us that the arguments for articulated phonological representations are overwhelming, and the fact that we have not employed them in this work is not to be taken as an argument against them.

Emit:	while in state:	prob	transition	prob	
a	1	.19	1 → 2	.94	
b	2	.05	2 → 2	.08	probability: 1.35 x 10^{-5}
ʁ	2	.14	2 → 1	.71	
a	1	.19			
Emit:	while in state:	prob	transition	prob	
a	1	.19	1 → 2	0.94	
b	2	.05	2 → 3	0.21	probability: 9.98 x 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 x 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 x 10^{-8}
b	3	.001	3 → 1	1	
a	1	.19			

Table 13: Structural differences between /abʁa/ and /aʁba/

Part of the motivation for the present work was the desire to be able to speak coherently about the vowel/consonant distinction within universal phonological theory; that is, for the purposes of represenational phonological theory, we wanted to be able to make a statement like, "a language may segregate vowels and consonants onto separate autosegmental tiers," but saying that requires an independent characterization of what a vowel and a consonant is, and we have in this paper offered several characterizations. Still, a question is raised: to what degree is there an equivalence, based on explicit methods of translation, between models of phonological knowledge that incorporate phonological representations of some complexity (e.g., autosegmental tiers, metrical grids with constituency) and models of the sort used in this paper? We raise the question, without being in a position to answer it at this point.

Third, we *use finite state automata*, which are devices not found elsewhere in generative grammar, and arguments have been made that finite state automata are not good models for natural language syntax. Answer: Finite state automata are minimalistic sorts of objects, and are natural ways of expressing the uncontroversial observation that many aspects of language occur sequentially

in time; the question is not *whether* we need them, but rather, how much do we need to move beyond them? In short, how inadequate are they? We have explored them here in order to see just how much we can get out of them.

Fourth, this work is not generative phonology, not optimality theory, not autosegmental phonology, and not quite a few other things. Answer: Yes, of course, that is true. It is largely inspired by work over the last several decades in machine learning. Current phonological theories have not been developed with the problem of learning kept front and center. Such theories have largely been developed with the goal in mind of accounting for morphophonemic alternations, and only after their design have phonologists asked how the language-specific aspects of a phonological grammar could be learned. We have turned the question around, and asked whether very simple devices whose abilities to learn are reasonably well understood can be used to help understand grammar.

Beyond these four less interesting observations, there are two general points to arise out of the work that we have discussed in this paper. The first is that a non-trivial 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; 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 will 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 paper 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: on 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 over-made, and it can lead to a perspective in which the scientist feels she may pick and choose the data that serves her hypothesis best.[19] 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).

In terminology suggested by 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 which 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-a-vis linguists who study I-language, and this advantage is only growing as improvements in computational and statistical

[19] 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 with regard to 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 acknowledgement that one's job is not yet finished. Those are two very different things.

methods become available.[20] Our purpose in this paper has been to demonstrate this proposition in several case studies.

Acknowledgements

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.

A On structuralist approaches to vowel/consonant definitions

Eli Fischer-Jorgensen's paper entitled "On the definition of phoneme categories on a distributional basis", is a major statement of the state of the art as of mid twentieth century (Fischer-Jörgensen 1952). She surveys the approaches to this problem that had been proposed by a range of phonologists, including Sapir (1925), Bloomfield (1933:219), Vogt (1942), Trager (1939), Togeby (1951); see also Sigurd (1955), and Householder (1962) and Householder (1971), especially chapter 11, as well as Fischer-Jörgensen (1975:375ff), and also Harary & Helmreich (2002). There is considerable discussion in some of these works about the question as to *why* phonologists should undertake the development of explicit methods of determining phonological categories and syllable structure; some of the discussion, such as Fischer-Jörgensen's, proposes a method which can be used on data from any language, while other discussions, such as Householder's discussion of English, aim essentially to provide an English-specific algorithm for assigning phonological categories and structure to strings of English phones. Fischer-Jörgensen (1975:376) summarizes Fischer-Jörgensen (1952) as follows:

[20] Lurking behind the discussion of the relative merits of studying E-language and I-language is the question whether methods typically applied to corpora (that is, E-language) will uncover properties best viewed as characterizing the *language*, or best viewed as characterizing the particular *corpus*—that is, when will a method produce a consistent results across different corpora selected from what we know, in a pre-theoretic sense, to be corpora selected from the same language. This is strictly an empirical question, requiring testing across real linguistic data. Needless to say, the discovery of properties that are *not* consistent across samples from the same language may be of considerable interest, if they identify instead other characteristics of interest, such as style, author identity, and so forth. In any event, methods of evaluating hypotheses regarding E-language are well-established, and shared across disciplines, while methods of evaluating I-language are not—whence comes the advantage to the E-language studies, alluded to in the text.

> In a paper of 1952 the PRESENT AUTHOR has discussed the possibility of establishing distributional catgories of phonemes which can be used for comparisons between languages. It is proposed to use positions within the syllable as the basic criterion. The paper further contains a discussion of the relation between syllable and minimum utterance and a discussion of structural law versus accidental gaps. It is argued that the placement of an exact borderline between structural law and accidental gap is arbitrary, since the rules determining the syllabic structure of a language form a hierarchy from the most specific to the most general laws. The more general the rule with which a given cluster would come into conflict, the safer is the statement that its absence is due to a structural law. Moreover, the frequency of the phonemes in question and perhaps the possibility of formulating the rule in terms of distinctive features should be taken into account. Some empirical observations concerning accident or law in the combination of different parts of the syllable are also mentioned.

From a contemporary perspective, it is curious that Fischer-Jörgensen's discussion fails to really come to grips with testing a method against any specific set of data. In order to find the category of "consonant" and "vowel", she writes, "It will probably be possible in nearly all languages to divide the phonemes into two classes, in such a way that the members of each class are mutually commutable, whereas memebers of the two different classes are not commutable...If members of one of the two (or three) categories can constitute a syllable base by themselves (e.g., i, a, u) there is an old tradition for calling members of this category vowels, and members of the other category consonants." In our experience, actual data rarely, if ever, provide all of the relevant contexts; the *existing* forms are only a subset of the *possible* forms, a well-known problem in corpus-based work in linguistics. To put the point more sharply, it will probably be *im*possible in nearly all languages to divide the phonemes into two classes in the way that Fischer-Jörgensen suggests.

Fischer-Jörgensen's method builds then on the prior categorization of segments into Cs and Vs. She proposes (after some discussion of how to determine what stretches of segments should be considered as the relevant domain for the investigation, which we may call a "word," recognizing that there are a number of serious questions being glossed over) that we determine what single con-

sonants can appear strictly before the first vowel of a word, and what single consonants can appear after the last vowel of a word. These categories typically (though not always) differ, and after they are established, further possibilities of combination can be made explicit word-initially and word-finally, as well as word-internally.

We intend no sharp criticism of Fischer-Jörgensen's work; since it ws done in the days before easy access to computers, her style of work was entirely reasonable. What is perhaps of most interest to us today is that both glossematic approaches (which Fischer-Jörgensen's work is an example of) and American structuralist work of the period recognized this as a significant and meaningful research project. The most detailed investigation and exploration of this general problem that we are aware of is that given by Spang-Hanssen (1959), whose work is even more in the glossematic tradition than Fischer-Jörgensen's is.

It is interesting to note that in work from this first generation, two phonemes being members of the same class is essentially equivalent to their appearing in exactly the same contexts: that is, classification depends on an absolute conception of distributional similarity. The first description adopting a gradualist definition of similarity that we are aware of is that found in O'Connor & Trim (1953:105–109):

> The method followed was to list all those phoneme combinations actually occurring...in the first two and the last two places in words....The number of contexts occupied in common by every pair of phonemes...was determined....In assessing the similarities and differences in the distributions of two phonemes, three figures must be taken into consideration, namely, the number of contexts held in common and the total number of occurrences of each of the two phonemes.

The similarity between phonemes is thus defined as the ratio of the number of contexts that they share to the number of total occurrences of the more restricted phoneme. For example, suppose two phonemes X and Y share 15 contexts (to be concrete, let us say that there are exactly 15 phonemes after which X and Y appear, though in fact O'Connor and Trim use a somewhat more complex notion of context, involving the preceding segment, the following segment, and word-initial and word-final positioning), and X appears in 24 different contexts, and Y in 20 different contexts. The the similarity between X and Y would be 15/20, or 75%.

O'Connor and Trim observe that, for their corpus, (phonetic) vowels almost always have a pairwise similarity of 50 percent or more, and less than 50 percent with consonants—which have a pairwise similarity almost always greater than 50 percent. But they note that the *optimal* value for such a threshold (that which best leads to a classification into vowels and consonants) can vary from language to language. In fact, in a similar treatment of French, Arnold (1956) finds the optimal value to be 60 percent. O'Connor & Trim (1953) also mention the case of Birman, where all words consist of a vowel either alone or preceded by a single consonant, and all combinations are attested (see **?**, p 264); in this extreme case, the optimal threshold would be 100 percent.

The work of O'Connor, Trim, and Arnold shows that the quantitative approach to the distributional analysis of phonemes had been undertaken by an earlier generation of phonologists. In the meantime, both progress in statistics and machine learning and increasing ease with which data and compuational power has become available have made the development of truly algorithmic procedures feasible. But these issues were generally excluded from the generative agenda, and would not return at all until the early 1990s.

B Sample corpus

Throughout the paper, we use the following corpus for our examples:

```
ban
banana
bib
binis
nab
saab
sans
sins
```

Table 14 below gives the number of occurrences of each phoneme and each sequence of two phonemes in this corpus. Sequences incing 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 D).

Phoneme	Count
b	7
n	7
s	6
a	8
i	4

Sequence	Count
aa	1
ab	2
an	4
ba	2
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 14: Number of occurrences of phonemes and sequences of phonemes.

C Conductance

As indicated in section 3.2.2, partitioning a graph by merely minimizing the resulting cut may lead to the dissociation of a small number of weakly connected nodes from a bulk of more strongly connected ones. In order to avoid this generally undesirable result, it is useful to add the constraint that the volumes of the sets S and T composing the partition should be balanced. One way of building this insight into the model is by seeking a partition that minimizes the cut and simultaneously maximizes the volume of the *smaller* of the two groups. Along these lines, the *conductance* of a partition $\{S, T\}$ is defined as the ratio of the cut to the volume of the smaller of S or T (Kannan *et al.* 2000):

$$\phi(S,T) := \frac{cut(S,T)}{\min(d(S), d(T))} \tag{4}$$

Note that $d(S) := \sum_{i \in S} d_i$ represents the sum of the degrees of the nodes of S in the original graph; $d(S)$ is generally greater than the volume of S, since it includes the contribution of edges that were cut: $d(S) = vol(S) + cut(S,T)$. The definition of conductance is clearly reminiscent of the earlier *normalized*

cut, defined by Shi & Malik (1997) as:

$$Ncut(S,T) := \frac{cut(S,T)}{d(S)} + \frac{cut(S,T)}{d(T)} \tag{5}$$

D 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 will 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[21], 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, we have already seen that the sample corpus given in appendix B, has an inventory of $n = 5$ phonemes $P = \{\texttt{b}, \texttt{n}, \texttt{s}, \texttt{a}, \texttt{i}\}$; the inventory of contexts is the same with the addition of the word boundary symbol `#`: $C = \{\texttt{\#\_}, \texttt{b\_}, \texttt{n\_}, \texttt{s\_}, \texttt{a\_}, \texttt{i\_}\}$.[22] Using the frequencies reported in Table 14 (p. 46), we construct the (5×6) matrix F as indicated: $f_{11} = Count(\texttt{\#b}) = 4$, $f_{12} = Count(\texttt{bb}) = 0$, and so on:

$$F = \begin{pmatrix} 4 & 0 & 0 & 0 & 2 & 1 \\ 1 & 0 & 0 & 0 & 4 & 2 \\ 3 & 0 & 2 & 0 & 0 & 1 \\ 0 & 2 & 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 1 & 0 & 0 \end{pmatrix} \tag{6}$$

Our goal is to use F to build the adjacency matrix A of a weighted undirected graph. Following Bavaud & Xanthos (2005), we do so by means of a two step

[21] So here m is at most equal to $n + 1$, the number of phonemes plus the word boundary symbol.

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

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.

We consider first the construction of the matrix W on the basis of F. As we have seen, the value f_{jk} found at the intersection of the j-th row and the k-th column of F represents the number of occurrences of phoneme j in context k in the relevant corpus. Let us focus on a single row of F, say the j-th row. The set of all values found in this row constitute the *distribution* of phoneme j. The sum $f_{j\bullet}$ of these values gives the total number of occurrences of j irrespective of the context. By dividing the k-th value in this row by the total frequency $f_{j\bullet}$, we obtain the *transition probability*[23] from phoneme j to context k:

$$pr(j \rightarrow k) := \frac{f_{jk}}{f_{j\bullet}} \tag{7}$$

For example, consider the phoneme `b` in the matrix F given in (6) above. It corresponds to row $j = 1$. Its total frequency in the corpus is equal to the sum of that row: $f_{1\bullet} = 4 + 2 + 1 = 7$. The column that corresponds to the word-initial context `#_` is $k = 1$, and the frequency of phoneme `b` in that context is $f_{11} = 4$. Thus, the transition probability from phoneme `b` to context `#_` is $pr(\texttt{b} \rightarrow \texttt{\#\_}) = f_{11}/f_{1\bullet} = 4/7$. Similarly, we find the other values on this row to be $pr(\texttt{b} \rightarrow \texttt{b\_}) = pr(\texttt{b} \rightarrow \texttt{n\_}) = pr(\texttt{b} \rightarrow \texttt{s\_}) = 0$, $pr(\texttt{b} \rightarrow \texttt{a\_}) = 2/7$, and $pr(\texttt{b} \rightarrow \texttt{i\_}) = 1/7$.

For any row j of F, it can be verified that the sum of the transition probabilities from phoneme j to each possible context k is equal to 1. Thus, dividing a row by its sum really amounts to *normalizing* it. By applying this normalization procedure to all the rows of F, we may define a new $(n \times m)$ matrix H, where the cell at the intersection of the j-th row and the k-th column is defined as

[23] 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).

$h_{jk} := pr(j \rightarrow k)$:

$$H = \begin{pmatrix} 4/7 & 0 & 0 & 0 & 2/7 & 1/7 \\ 1/7 & 0 & 0 & 0 & 4/7 & 2/7 \\ 3/6 & 0 & 2/6 & 0 & 0 & 1/6 \\ 0 & 2/8 & 3/8 & 2/8 & 1/8 & 0 \\ 0 & 2/4 & 1/4 & 1/4 & 0 & 0 \end{pmatrix} \tag{8}$$

The very same procedure can be applied to the *columns* of F. Although this is a less usual conception in phonology, the k-th column of F can be viewed as the distribution of context k. The sum $f_{\bullet k}$ of the values found in this column is the total number of occurrences of that context. Thus we may define the transition probability from context k to phoneme j as:

$$pr(k \rightarrow j) := \frac{f_{jk}}{f_{\bullet k}} \tag{9}$$

By normalizing all the columns of F in this fashion, we may construct another $(n \times m)$ matrix V, where the cell at the intersection of the j-th row and the k-th column is defined as $v_{jk} := pr(k \rightarrow j)$:

$$V = \begin{pmatrix} 4/8 & 0 & 0 & 0 & 2/7 & 1/4 \\ 1/8 & 0 & 0 & 0 & 4/7 & 2/4 \\ 3/8 & 0 & 2/6 & 0 & 0 & 1/4 \\ 0 & 2/4 & 3/6 & 2/3 & 1/7 & 0 \\ 0 & 2/4 & 1/6 & 1/3 & 0 & 0 \end{pmatrix} \tag{10}$$

The newly constructed matrices H and V enable us to calculate the probability for a given phoneme j to occur in the same context as another phoneme i. Let us consider first the case of a single context k. The cell h_{ik} at the intersection of the i-th row and the k-th column of H gives the transition probability $pr(i \rightarrow k)$ from phoneme i to that context; the cell v_{jk} at the intersection of the j-th row and the k-th column of V gives the transition probability $pr(k \rightarrow j)$ from that context to phoneme j. The product of these two values can be interpreted as the probability of picking context k among the contexts in which phoneme i occurs, *then* picking phoneme j among the phonemes which occur in context k; we call this product the transition probability from phoneme i to

phoneme j *via context* k:

$$pr(i \rightarrow k \rightarrow j) := pr(i \rightarrow k)\, pr(k \rightarrow j) = h_{ik} v_{jk} \tag{11}$$

For example, consider phonemes `b` ($i = 1$) and `n` ($j = 2$), and context `a_` ($k = 5$). The transition probability from phonemes `b` to `n` via context `a_` is $pr(\texttt{b} \rightarrow \texttt{a\_} \rightarrow \texttt{n}) := pr(\texttt{b} \rightarrow \texttt{a\_})\, pr(\texttt{a\_} \rightarrow \texttt{n}) = h_{15} v_{25} = 2/7 \cdot 4/7 = 8/49$. The transition probability between the same phonemes via context `i_` ($k = 6$) is $pr(\texttt{b} \rightarrow \texttt{i\_} \rightarrow \texttt{n}) := h_{16} v_{26} = 1/7 \cdot 2/4 = 1/14$.

This notion can be extended to *all* the contexts in which phonemes i and j occur. Thus, the transition probability from phoneme i to phoneme j via *all* contexts is simply defined as the sum of the transition probabilities from i to j via each possible context k:

$$pr(i \rightarrow j) := \sum_k pr(i \rightarrow k \rightarrow j) = \sum_k h_{ik} v_{jk} \tag{12}$$

In our example, the transition probability from phonemes `b` to `n` via all contexts is $pr(\texttt{b} \rightarrow \texttt{n}) := \sum_k h_{1k} v_{2k} = 1/14 + 0 + 0 + 0 + 8/49 + 1/14 = 15/49 = .31$.

We may eventually build the $(n \times n)$ square matrix W by calculating the transition probability for each pair of phonemes i and j, and storing the result in the cell at the intersection of the i-th row and the j-th column of W: $w_{ij} := pr(i \rightarrow j)$. In our case, the resulting matrix is:

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

The probability $pr(i \rightarrow j)$ is maximal when i and j have the same distribution (in the mathematical sense)[24] and minimal when i and j never occur in the same context, i.e. when their distributions are complementary. In spite of this correlation with distributional similarity, however, $pr(i \rightarrow j)$ is not suitable as an actual measure of it, insofar as it is not symmetric: $pr(i \rightarrow j) \neq pr(j \rightarrow i)$ in general.

On the other hand, W has certain properties that entail a natural way

[24] or, to be precise, when their distributions are exactly *proportional*.

of turning it into a symmetric matrix.[25] 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 defined as:

$$a_{ij} := \pi_i \cdot pr(i \rightarrow j) \tag{14}$$

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:

$$A = \begin{pmatrix} .09 & .07 & .05 & .01 & 0 \\ .07 & .11 & .03 & .02 & 0 \\ .05 & .03 & .06 & .03 & .01 \\ .01 & .02 & .03 & .13 & .07 \\ 0 & 0 & .01 & .07 & .05 \end{pmatrix} \tag{15}$$

This is actually the adjacency matrix that we used as an example in section 3.2.1 and represented in Figure 1 (p. 13). 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.[26] Phonemes with similar distributions are strongly connected, whereas phonemes with dissimilar distributions are weakly or not connected. As we have seen in section 3.2.2, 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.[27]

[25] In particular, W is a *reversible* transition matrix.

[26] 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 (see Bavaud & Xanthos 2005).

[27] For mathematical reasons that are beyond the scope of this paper, the actual matrix that undergoes the spectral decomposition discussed in section 3.2.2 is a *normalized* version of A, defined as $C := \Pi^{-\frac{1}{2}} A \Pi^{-\frac{1}{2}}$, where Π stands for the matrix containing the stationary probabilities of phonemes on the main diagonal and 0's everywhere else (see e.g. Bavaud & Xanthos 2005 for details on this).

References

Arnold, Gerald F. 1956. A phonological approach to vowel, consonant, and syllable in modern French. *Lingua* 5.253–287.

Bavaud, F., & A. Xanthos. 2005. Markov associativities. *Journal of Quantitative Linguistics* 12.123–137.

Belkin, M., & J. Goldsmith. 2002. Using eigenvectors of the bigram graph to infer morpheme identity. In *Morphological and Phonological Learning: Proceedings of the 6th Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON)*, 41–47.

Bezdek, J.C. 1981. *Pattern Recognition with Fuzzy Objective Function Algoritms*. New York: Plenum Press.

Biggs, N. 1993. *Algebraic Graph Theory*. Cambridge: Cambrige University Press, second edition.

Bloomfield, L. 1933. *Language*. New York: H. Holt and Company.

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

Chomsky, N. 1986. *Knowledge of Language*. New York: Praeger.

——, 2000. An interview on minimalism.

Chung, F.R.K. 1997. *Spectral Graph Theory*. Providence: American Mathematical Society.

Dowman, M. ms. Minimum description length as a solution to the problem of generalization in syntactic theory.

Ellison, T. Mark. 1992. The iterative learning of phonological constraints.

——, 1994. *The Machine Learning of Phonological Structure*. University of Western Australia dissertation.

——, 2001. Induction and inherent similarity.

Finch, Steven, 1993. *Finding Structure in Language*. University of Edinbugh dissertation.

FISCHER-JÖRGENSEN, E. 1952. On the definition of phoneme categories on a distributional basis. *Acta Linguistica* 7.8–39.

—— 1975. *Trends in Phonological Theory*. Copenhagen: Akademisk Forlag.

GOLDSMITH, J. 1994. A dynamic computational theory of accent systems. In *Perspectives in Phonology*, ed. by Jennifer Cole & Charles Kisseberth, 1–28. Stanford: Center for the Study of Language and Information.

—— 2001. The unsupervised learning of natural language morphology. *Computational Linguistics* 27.153–198.

——, & J. O'BRIEN. 2006. Learning inflectional classes. *Language Learning and Development* 2.219–250.

GOLDWATER, SHARON, 2006. *Nonparametric Bayesian Models of Lexical Acquisition*. Brown University dissertation.

GUY, J. 1991. Vowel identification: an old (but good) algorithm. *Cryptologia* XV.258–262.

HARARY, FRANK, & STEPHEN HELMREICH. 2002. On the bipartite distribution of phonemes. In *The Legacy of Zellig Harris: Language and information into the 21st Century Vol. 2. Mathematics and computability of language*, ed. by Bruce Nevin. Amsterdam: John Benjamins.

HOUSEHOLDER, F. 1962. The distributional determination of English phonemes. *Lingua* 11.186–191.

—— 1971. *Linguistic Speculations*. Cambridge: Cambridge University Press.

JELINEK, F. 1997. *Statistical Methods for Speech Recognition*. Cambridge: MIT Press.

JURAFSKY, D., & J. MARTIN. 2000. *Speech and Language Processing*. Upper Saddle River, NJ: Prentice-Hall.

KANNAN, R., S. VEMPALA, & A. VETTA. 2000. On clusterings: Good, bad, and spectral. In *Proceedings of the 41st Annual Symposium on the Foundation of Computer Science*, 367–380.

O'CONNOR, J., & J. TRIM. 1953. Vowel, consonant and syllable—a phonological definition. *Word* 9.103–122.

Peperkamp, S., R. Le Calvez, J.-P. Nadal, & E. Dupoux. 2006. The acquisition of allophonic rules: statistical learning with linguistic constraints. *Cognition* 101.B31–B41.

Powers, D. 1997. Unsupervised learning of linguistic structure: an empirical evaluation. *International Journal of Corpus Linguistics* 2.91–132.

Powers, David M. W. 1991. How far can self-organization go? Results in supervised language learning. *Proceedings of AAAI Spring Symposium on Machine Learning of Natural Language and Ontology* 131–137.

Reichenbach, H. 1938. *Experience and Prediction*. Chicago: University of Chicago Press.

Rissanen, Jorma. 1989. *Stochastic Complexity in Statistical Inquiry*. New Jersey: World Scientific Publishing Company.

Saffran, J., R. Aslin, & E. Newport. 1996. Statistical learning by 8-month-old infants. *Science* 274.1926–1928.

Sapir, E. 1925. Sound patterns in language. *Language* 1.37–51.

Schifferdecker, G., 1994. Finding structure in language. Master's thesis, University of Karlsruhe.

Shi, J., & J. Malik. 1997. Normalized cuts and image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 731–737.

Sigurd, B. 1955. Rank order of consonants established by distributional criteria. *Studia Linguistica* IX.8–20.

Spang-Hanssen, H. 1959. *Probability and Structural Classification*. Copenhagen: Rosenkilde and Bagger.

Sukhotin, B.V. 1962. Eksperimental'noe vydelenie klassov bukv s pomoščju EVM. *Problemy strukturnoj lingvistiki* 234.189–206.

—— 1973. Méthode de déchiffrage, outil de recherche en linguistique. *T.A. Informations* 2.1–43.

Togeby, K. 1951. *Structure immanente de la langue française*. Copenhagen: Nordisk Sprog- og Kulturforlag.

TRAGER, G. 1939. La systématique des phonèmes du polonais. *Acta Linguistica* 1.179–188.

VOGT, HANS. 1942. The structure of the Norwegian monosyllables. *Norsk Tidsskrift for Sprogvidenskap* XII.5–29.

WARD, JOE H. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association* 58.236–244.