An algorithm for learning phonological classes from distributional similarity

Connor Mayer

Abstract

An outstanding question in phonology is to what degree the learner uses distributional information rather than substantive properties of speech sounds when learning phonological structure. This paper presents an algorithm that learns phonological classes from only distributional information: the contexts in which sounds occur. The input is a segmental corpus, and the output is a set of phonological classes. The algorithm is first tested on an artificial language with both overlapping and nested classes reflected in the distribution. It retrieves the expected classes, and performs well as distributional noise is added. It is then tested on four natural languages. It distinguishes between consonants and vowels in all cases, and finds more detailed, language-specific structure. These results improve on past approaches, and are encouraging given the paucity of the input. Further refined models may provide additional insight into which phonological classes are apparent in the distributions of sounds in natural languages.

1 Introduction

An outstanding question in all areas of linguistics is how much of human language is innate and how much is learned from data (e.g., Chomsky, 1957, 1965, 1988; Elman et al., 1996; Pullum & Scholz, 2002; Tomasello, 2003, a.o.). From this perspective, the question of how much information about phonological categories can be retrieved strictly from distributional information is of considerable interest to the field of phonology.

One of the central observations of phonological theory is that speech sounds tend to pattern according to phonetic similarity, both within and across languages (e.g., Chomsky & Halle, 1968; Mielke, 2008). For example, processes like final obstruent devoicing, where voiced obstruents become voiceless word- or syllable-finally, are common across languages (e.g., Wetzels & Mascaró, 2001; Iverson & Salmons, 2011). This process often prohibits all voiced stops, affricates, and fricatives in a language from occurring in these positions, as in Dutch or Polish. Despite the differences in place and manner of articulation across these sounds, they share two phonetic properties that cause them to be treated as a single class: near or complete impediment of the airflow out of the vocal tract, and vocal fold vibration.

*This research was supported by the Social Sciences and Humanities Research Council of Canada. I would like to thank Bruce Hayes, Kie Zuraw, Yizhou Sun, Tim Hunter, Robert Daland, and Pat Keating for their guidance and support throughout this project. Thanks also to the attendees of the UCLA phonology seminar and the 2018 Annual Meeting on Phonology for their valuable questions and insights. Finally, thanks to three anonymous reviewers for their valuable feedback and suggestions.

Based on this robust typological generalisation, classic work has suggested that there is a universal tendency for language learners to group sounds based on their phonetic properties (e.g., Chomsky & Halle, 1968). Languages may use classes differently in their phonologies, but in principle the set of classes available across languages should be the same by virtue of shared human physiology.

There is evidence, however, for the existence of classes that do not appear to be phonetically coherent, such as the notorious Sanskrit "ruki" class (e.g., Kiparsky, 1973; Vennemann, 1974); the triggers for Philadelphia /æ/-tensing (Labov et al., 2006); or Cochabamba Quechua, where etymological /q/ has become [ʁ], but still patterns with the voiceless stops (Gallagher, 2019). Mielke (2008) presents many such cases. Instances of variable patterning of a segment across languages also bear on this issue. For example, /l/ varies in whether a language treats it as $\left[\ +\text{continuant}\ \right]$ or $\left[\ -\text{continuant}\ \right]$ (e.g., Kaisse, 2002; Mielke, 2008). In line with this observation, I use the term *phonological class* throughout this paper, rather than *natural class*, to refer to a set of sounds that behave in a uniform way in the phonology of a language without necessarily sharing any phonetic properties.

These observations have led some researchers to propose that phonological classes may be learned and language-specific (e.g., Mielke, 2008; Dresher, 2014; MacWhinney & O'Grady, 2015; Archangeli & Pulleyblank, 2015, 2018). Under such theories, phonologically salient classes need not be phonetically coherent, and distributional learning must account for a larger part of phonological acquisition than previously thought. That is, a phonological class is identified based (at least in part) on how its members pattern in the language, rather than some shared phonetic quality.

The typological observation that classes tend to be phonetically coherent is accounted for by suggesting a tendency for similar sounds to undergo similar phonetically-driven diachronic processes that lead to this patterning (e.g., Blevins, 2004). In other words, typology is governed primarily by pressures on language transmission, rather than biases in the learner. These pressures will tend to generate phonetically natural outcomes, though unnatural outcomes are also possible in certain circumstances (Beguš, 2018a, 2018b). This claim remains controversial (see, e.g., Kiparsky, 2006, 2008), though unnatural outcomes have been frequently documented (e.g., Bach & Harms, 1972; Mielke, 2008; Scheer, 2015; Beguš & Nazarov, to appear).

Regardless of whether one is willing to commit to the position of emergent classes, these ideas raise theoretically interesting questions. Namely, to what extent are phonological classes apparent in the distribution of sounds in a language, and to what extent might learners use this information?

For a class to be *apparent* in the distribution of sounds in a language, the sounds in that class must impose similar restrictions on which sounds may occur nearby, and this effect must be *learnable*: that is, robust enough to be detectable by some statistical learning algorithm. Distributional learning, however, is only one source of information available to the human learner. Even if a class is apparent in the input data, this does not mean that learners are bound to use it. Learners may rely on such statistical information only when it is robust enough to override other (possibly conflicting) influences, such as phonetic information or learning biases (e.g., Moreton, 2008; Hayes et al., 2009).

This paper will investigate the learning of phonological classes when *only* distributional information is available. That is, it deals with the question of what classes are apparent, rather than the question of what the learner might actually make use of. It will do so by detailing an algorithm that attempts to learn as much phonological information as possible solely from the contexts in which sounds do and do not occur, building on past work (e.g., Goldsmith & Xanthos, 2009; Nazarov, 2014, 2016). Again, this is not to suggest that phonetic information does not play an important role in characterising phonological classes: rather it is an attempt to see how far we can get while restricting ourselves to only one of the many sources of information available to the learner.

From a high level, the algorithm consists of four components that each contribute to accurate learning of phonological classes. These are:

1. **Vector embedding** (Section 4-4.1): Sounds in a phonological corpus are projected into a vector space (represented as points in a high-dimensional space) based on the contexts in which they occur. This allows for numerical comparison between sounds.

2. **Normalisation** (Section 4.2): The values of the vector representations of sounds are transformed using a statistical technique that highlights informative contexts.

3. **Principal Component Analysis** (Section 5.1): The normalised vector representations are transformed into lower dimensional representations that retain maximal variance. This allows generalisation across contexts that provide similar information.

4. **Clustering** (Section 5.2): A clustering algorithm is applied to the low-dimensional, normalised vector representations to find classes within the data.

Steps 3 and 4 are recursively performed on the discovered classes, allowing classes of different sizes to be found (Section 5.3).

Aside from eschewing phonetic information, this algorithm operates under two additional assumptions. First, it focuses only on phonotactic information: there is no explicit attempt to capture alternations. Although this may be a reasonable assumption about the initial phonological learning done by infants (e.g., Hayes, 2004; Prince & Tesar, 2004; Jarosz, 2006), it is expressly adopted here as a simplifying assumption. More sophisticated models may benefit from incorporating this information. Second, because it takes phonemic text corpora as input, it necessarily assumes that the learner has access to a segmental representation of speech (e.g., Lin, 2005; Feldman et al., 2013).

The output of the algorithm is a set of phonological classes that may be viewed as implicitly reflecting a feature system, in that any class contained in this set can be uniquely characterized by some combination of feature/value pairs. The process of deriving an explicit feature system from a set of classes is described in a related paper (C. Mayer & Daland, in press).

This paper is structured as follows. Section 2 reviews past research that has taken a distributional approach to learning phonological structure. Section 3 describes a toy language with well defined phonotactic properties, which will serve as a running example throughout the paper and a basic test case for the algorithm. The next two sections describe the components of the algorithm. Section 4 details how a normalised vector space representation of the sounds of a language can be generated from a phonological corpus. Section 5 shows how a combination of Principal Component Analysis and clustering algorithms can be used to extract potential phonological classes from such embeddings, and details its performance on the toy language. Section 6 presents the results of its application to Samoan, English, French, and Finnish. It is able to successfully distinguish consonants and vowels in every case, and retrieves interpretable classes within those categories for each language. Finally, Section 7 compares these results against past work, and Section 8 offers discussion of the results and proposals for future research.

2 Previous work

Distributional learning has been proposed in most areas of linguistics, suggesting that it may be a domain-general process. Examples include word segmentation and morphology (e.g., Saffran et al., 1996; Goldwater et al., 2009; Goldsmith, 2010), syntax (e.g., Redington et al., 1998; Wonnacott et al., 2008), and semantics (e.g., Andrews et al., 2009; Bruni et al., 2014).

Distributional approaches to phonology have been explored since the early part of the 20th century (e.g., Harris, 1946), but as Goldsmith and Xanthos (2009) point out, most of this work is not well known today. A number of pre-generative phonologists discussed the merits of such approaches and potential implementations, but this work was necessarily limited by technological factors.[1] The increasing availability of ever more powerful computers, together with advances in statistical and machine learning research, have recently rendered such approaches more viable.[2]

Powers (1997) provides an extremely detailed empirical comparison of early work building on these advances applied to English. Notable additions include the abstraction of representing sounds as points in a high dimensional space, normalization to probabilities, and the use of matrix factorisation and bottom-up clustering algorithms to group sounds together. While these approaches were a notable step forward, they frequently failed to achieve the basic distinction between consonants and vowels. This should be taken with some caution, however, as Powers ran his evaluations on orthographic data, whose vowels (a, e, i, o, u, and sometimes y) do not map straightforwardly onto phonemic vowels.

In the same time period, Ellison (1991, 1994) explored a *minimum description length* analysis, which uses an information theoretic objective function to evaluate how well a set of classes fits an observed data set. The optimal candidate set of classes is found using simulated annealing. Ellison reports that his method is generally successful in differentiating consonants and vowels across a wide range of languages, as well as identifying aspects of harmony systems. Ellison also runs his models on orthographic data.

More recently, Goldsmith and Xanthos (2009) compared three different approaches to learning phonological classes in English, French, and Finnish. The first, Sukhotin's algorithm, is mostly of historical interest, but can differentiate between consonants and vowels reasonably well using calculations simple enough to be performed by hand. Their second approach uses *spectral clustering*, which models distributional similarity between segments as an undirected graph with weighted edges. By representing the graph as a matrix and using spectral decomposition to find an optimal partition into two or more groups, Goldsmith &

[1] See Appendix A in Goldsmith and Xanthos (2008) for a detailed summary of this work.

[2] Some of the terms in this section may be unfamiliar to readers. Those terms that are relevant to understanding the algorithm presented in this paper will be defined in later sections.

Xanthos were able to successfully distinguish between consonants and vowels, and provide a basic characterisation of harmony systems. The final approach they examine is *maximum likelihood hidden Markov models*. These use a finite state machine with some small number of states (e.g., two for vowel vs. consonant). The model is trained to calculate transition and emission probabilities that maximise the likelihood of the data. The ratio of emission probabilities for each segment between states can then be used to classify them. This approach worked well for distinguishing vowels and consonants, identifying vowel harmony, and (to some extent) syllable structure.

Calderone (2009) used a similar approach to spectral clustering, *independent component analysis*, which decomposes a matrix of observed data into a mixture of statistically independent, non-Gaussian components. This resulted in a qualitative separation between consonants and vowels, as well as suggesting some finer grained distinctions within these sets.

Taking a different approach, Nazarov (2014, 2016) details an algorithm for jointly learning phonological classes and constraints using a combination of maximum entropy learning and Gaussian mixture models. Nazarov's method calculates the information gain from introducing a constraint that penalises a segment in a particular context, then clusters segments based on this information gain using a Gaussian mixture model. Segments that are clustered together are hypothesised to form a natural class, and specific constraints are in turn combined into more general ones using these classes. This performs well on a simple artificial language.

Finally, some recent work has investigated using neural networks to discover phonological classes from distributional data. Silfverberg et al. (2018) use a recurrent neural network to generate phoneme embeddings of sounds in phonological corpus. They show that these embeddings correlate well with embeddings based on provided distinctive features, but do not attempt to identify classes explicitly. Their non-neural comparison model employs vector embedding, singular value decomposition, and normalisation using positive pointwise mutual information, all of which are used in the algorithm presented below, but they do not take phoneme ordering into account. Similarly, Mirea and Bicknell (2019) generate phoneme embeddings using a long-term short-term memory neural network, and perform hierarchical clustering on these embeddings. This clustering does not cleanly separate consonants and vowels, though some suggestive groupings are present.

The goal of this paper is to expand on the successes of this ongoing, collective research program. The algorithm described below shares many aspects with past work, such as vector embedding (Powers, 1997; Goldsmith & Xanthos, 2009; Calderone, 2009; Nazarov, 2014, 2016; Silfverberg et al., 2018; Mirea & Bicknell, 2019), normalization (Powers, 1997; Silfverberg et al., 2018), matrix decomposition (Powers, 1997; Goldsmith & Xanthos, 2009; Calderone, 2009; Silfverberg et al., 2018), and clustering algorithms (Powers, 1997; Nazarov, 2014, 2016; Mirea & Bicknell, 2019). The innovations that will be presented below are largely in the combination and extension of these techniques, though the clustering methodology presented is relatively novel.

I will show that these innovations allow this algorithm to successfully find classes that stand in a complex relationship to one another in an artificial language and is successful in learning finer-grained categories in natural languages, while requiring fewer assumptions than past work. The modular structure of the algorithm provides a useful general framework in which further studies of distributional learning might proceed by altering individual components. The code implementing this algorithm is publicly available, and researchers are encouraged to use and modify it for their own purposes.[3]

3 Parupa: An artificial language

Because it is not clear a priori what classes might be apparent in the distribution of a natural language, it is useful to begin with a case where the target classes are known in advance, a practice adopted by past work (Goldsmith & Xanthos, 2009; Nazarov, 2014, 2016). To this end, I introduce an artificial language called *Parupa*,[4] which has well-defined phonotactic properties. Parupa serves as a running example throughout the paper and an initial test case for the algorithm. Its consonant and vowel inventories are shown in Table 1.

p	t	k
b	d	g
	r	

i		u
e		o
	a	

Table 1: Parupa consonants and vowels

Parupa has the following distributional properties:

1. All syllables are CV.
2. Vowel harmony: words must contain only front (/i/, /e/) or back (/u/, /o/) vowels. /a/ may occur in either case (i.e., it is transparent to harmony).
3. Words must begin with /p/ or /b/.
4. Consonant-vowel co-occurrence restrictions: /p/, /t/, and /k/ must be followed by high vowels or /a/. /b/, /d/, and /g/ must be followed by mid vowels or /a/. /r/ may be followed by any vowel. In other words, the full set of consonants is only in contrast before /a/.

Note that although these properties vary in their 'phonetic naturalness', there is no notion of phonetic substance in this model. These particular properties were chosen in part to emphasise that the distributional learning algorithm does not distinguish between natural and unnatural patterns. More importantly, however, they were chosen to produce multiple, overlapping partitions of the sets of vowels and consonants. For example, the vowel set is partitioned in two different ways: high-mid, and front-back. This structure is common in natural languages, and introduces challenges for many clustering algorithms (see Section 5). Given these properties, the algorithm should retrieve the classes shown in Figure 1.

A language corpus was generated using a Hidden Markov Model, shown in Figure 2. Although all transition and emission probabilities for any state were equal, the phonotactic

[3] The source code can be found at `https://github.com/connormayer/distributional_learning`.

[4] Named after one of the first words generated by the Hidden Markov Model in Figure 2.

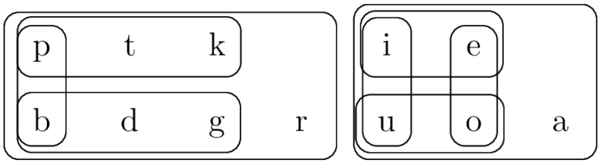


Figure 1: The phonological classes of Parupa.

constraints meant that not all segments were equally common in the corpus (e.g., /a/ was the most frequent vowel). The generated corpus had 50k word tokens, resulting in a total of about 18k word types. The input to the algorithm consists only of the word types.[5] The average word length was three syllables. Examples of Parupa words are shown in Table 2.

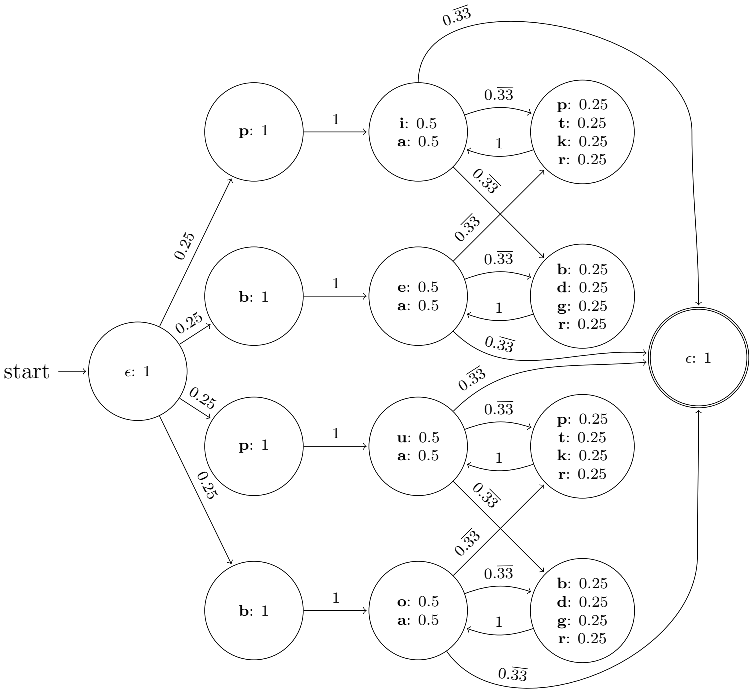


Figure 2: The Hidden Markov Model used to generate Parupa. Edges are labeled with their transition probabilities, and states are labeled with their segment emission probabilities. ϵ is the empty string.

I will use Parupa as a running example throughout the rest of the paper to illustrate the various components of the algorithm.

[5] Consistent with previous phonological modeling done over corpora like Bybee (1995), type frequency produces more interpretable results than token frequency, and is used throughout. In other words, the corpora employed are dictionary-like lists of word types rather than texts containing multiple word tokens.

berari	pupabopa
pa	paka
boka	padoropa
bo	pakubatuda
bopu	piretiba
pabarubo	barika

Table 2: Some Parupa words

4 Quantifying similarity: Vector space models

This model operates under the assumption that phonologically similar sounds[6] in a language should have similar *distributions* (e.g., Harris, 1946; Mielke, 2008). A distribution is a description of how frequently each outcome in a set of possible outcomes is observed in a data set. In this case, the outcomes we are interested in are the *contexts* in which a sound occurs: i.e., the other sounds that occur near it.

This assumption can be broken down into two components. First, I assume that if two sounds participate in the same phonological pattern, then they must share some abstract representational label: the label indicates the "sameness" of the two sounds with respect to this pattern. Second, I assume that all abstract representational labels must be discoverable from some phonological pattern, and hence for every abstract representational label, there will be a phonological pattern that uses this label. That is to say, we should not expect the learner to posit abstract structure in the absence of some detectable influence on the data.

To quantify the distributions of each sound, I adopt *vector space modelling* (e.g., Manning & Schütze, 1999; Jurafsky & Martin, 2008). The principle behind this approach is to represent objects as vectors or points in an n-dimensional space whose dimensions reflect some of their properties. Embedding objects in a vector space allows for convenient numerical manipulation and comparison between them.

This approach is commonly applied in many language-related domains: in document retrieval, where documents are represented by vectors whose dimensions reflect words that occur in the document; in computational semantics, where words are represented by vectors whose dimensions reflect other words that occur near the target word; and in speech/speaker recognition, where sounds are represented by vectors whose components are certain acoustic parameters of the sound. This is also essentially the approach taken by many of the papers discussed in the previous section, where sounds are represented as vectors whose dimensions reflect counts of sounds that occur near them. Note that sounds and words are different from documents, in their embeddings are defined by what occurs *around* them, rather than any intrinsic property. Whether we are dealing with documents, words, or sounds, the projection of these objects into a vector space should be done in such a way that similar objects end up closer in the space than less similar ones.

An important distinction between applying this approach to documents or words and applying it to sounds is that *order* is crucially important for sounds. When considering the

[6] I intentionally use the word "sounds" rather than "phonemes" or "phones" because this model is independent of the level of transcription used in the corpus.

semantics of words or documents, it is generally more useful to know that a word occurs in a document or that a word occurs near another word than it is to know that a word is the nth word in a document, or that a word occurs exactly n words before another word. In contrast, ordering is crucial for phonology, since adjacency and directionality play important roles in phonological processes.[7]

The method I use for generating embeddings combines aspects of the approaches described above. Before going into more detail, I will first provide a simple, concrete example of how we can construct a vector representation of sounds in a phonological corpus.

4.1 A simple vector embedding of sounds

Suppose we have a language with only two sounds, /t/ and /a/, and a corpus containing the following five words:

$$\text{ta, ata, tata, atta, taa} \tag{1}$$

Σ is the set of all unique symbols in the corpus. There is an additional symbol not in Σ, #, which represents a word boundary.[8] Here $\Sigma = \{\mathrm{t}, \mathrm{a}\}$.

To go from this corpus to a vector representation of the sounds, we must decide how to define the dimensions of the vector space: i.e., which aspects of a sound's distribution we wish to be sensitive to, and how to quantify these aspects. For this simple example, I will define each dimension in the space as the *bigram* counts of the number of times a particular symbol occurs immediately *before* the target symbol. That is, the corresponding vector for each symbol in Σ consists of dimensions with labels $s_1\_$, where _ indicates the position of the target sound (the sound whose vector we are constructing), $s_1 \in \Sigma \cup \{\#\}$, and the value of each element is the number of times s_1 occurs before the target sound in the corpus. In general when discussing dimensions, I will use _ to indicate the position of the target sound, and s with subscripts to indicate sounds in the context.

A matrix consisting of the resulting count vectors is represented in Table 3.

	t_	a_	#_
t	1	3	3
a	6	1	2

Table 3: A matrix consisting of count vectors for a toy language.

For example, the cell in the bottom left corner of this table has the value 6 because /a/ occurs after /t/ six times in the corpus. Note that although these sounds have overlapping distributions, these vectors capture the general pattern of alternation between the two. It is straightforward to see how these counts can be interpreted as points or vectors in 3D space, where $\mathrm{t} = (1, 3, 3)$ and $\mathrm{a} = (6, 1, 2)$.

[7] Not all aspects of ordering are important for phonology: knowing that a sound is the third sound in a word is not generally useful, although knowing that a sound is first or last in a word can be.

[8] For clarity, I omit word boundaries in the presentation of the data here. In practice when using n-gram counts, the words in a corpus are padded on either side with $n - 1$ word boundary symbols.

4.2 What do we count when we count sounds?

The previous example counts sounds that occur immediately preceding the target sound. This is unlikely to be informative enough for anything but the simplest languages. There are many other ways we might choose to count contexts. Here I adopt a type of *trigram* counting, which counts all contiguous triples of sounds that contain the target sound.[9] Thus our dimension labels will be of the form $s_1s_2\_$, $s_1\_s_3$, and $\_s_2s_3$, where $s_1, s_2, s_3 \in \Sigma \cup \{\#\}$.

Formally, we define an m by n matrix C, where $m = |\Sigma|$ (the number of sounds in the language), and n is the number of contexts. Under the trigram contexts considered here, $n = 3|\Sigma \cup \{\#\}|^2$. s and c are indexes referring to a specific sound and context respectively, and each matrix cell $C(s, c)$ is the number of times sound s occurs in context c in the corpus.

4.3 Normalised counts

Raw counts tend to not be particularly useful when dealing with vector embeddings of words, because many different types of words can occur in the same contexts (e.g., near *the* or *is*). A common technique is to somehow normalise the counts, such as by converting them to probabilities, conditional probabilities, or more sophisticated measures. Normalisation proves to be valuable for sounds as well. The basic assumption I make is that the most fundamental partition of the sounds in any language should be between consonants and vowels (or alternatively, sounds that occupy syllable nuclei and sounds that do not). A suitable normalisation method should make this distinction apparent. Of the normalisations tested, only *Positive Pointwise Mutual Information* (PPMI) was able to consistently produce a clean distinction between consonants and vowels across data sets.[10]

PPMI is an information theoretic measure that reflects how frequently a sound occurs in a context compared to what we would expect if sound and context were independent (Church & Hanks, 1990). It has been used in previous models of distributional phonological learning (Silfverberg et al., 2018). It is defined as follows

$$\text{PPMI}(s, c) = \max\Big(\log_2 \frac{P(s, c)}{P(s)P(c)}, 0\Big) \tag{2}$$

where s is a sound and c is a context. If $P(s)$ and $P(c)$ are independent, then $P(s, c) \approx P(s)P(c)$ and hence the value of the inner term $\log_2 \frac{P(s,c)}{P(s)P(c)}$ will be close to 0. If $P(s, c)$ occurs more frequently than the individual probabilities of s and c would predict then the value will be positive, and if $P(s, c)$ occurs less frequently than expected, it will be negative.

PPMI converts all negative values of the inner term to 0 (as opposed to *Pointwise Mutual Information*, which does not; Fano, 1961). This is desirable when dealing with words, because the size of the vocabulary often requires an unreasonable amount of data to distinguish between words that tend not to co-occur for principled reasons and words that happen not to co-occur in the corpus (e.g., Dagan et al., 1993; Niwa & Nitta, 1994). Although this seems as though it should be less of a concern with phonological data given the relatively

[9] The provided software allows for bigram, trigram, and a combination of bigram and trigram counts. Trigram counts resulted in the best performance, and are used throughout the paper. See Appendix A.1.

[10] The provided software also allows normalisation using raw counts, probabilities, conditional probabilities, and PMI. PPMI is used throughout the paper. See Appendix A.2.

small number of sounds, in practice PPMI provides more interpretable results than PMI on several of the data sets examined here (see Appendix A.2).[11]

Table 4 shows a matrix consisting of the values from Table 3 converted to PPMI. The separation between the two vectors on each dimension has become even more pronounced.

	t_	a_	#_
t	0	0.78	0.46
a	0.61	0	0

Table 4: Count vectors for a toy language normalised using PPMI.

The three probabilities used to calculate (2) can be straightforwardly calculated from the matrix C containing phoneme/context counts using maximum likelihood estimation:

$$P(s,c) = \frac{C(s,c)}{\sum\limits_{i,j} C(i,j)} \tag{3}$$

$$P(s) = \sum_{c} P(s,c) \tag{4}$$

$$P(c) = \sum_{s} P(s,c) \tag{5}$$

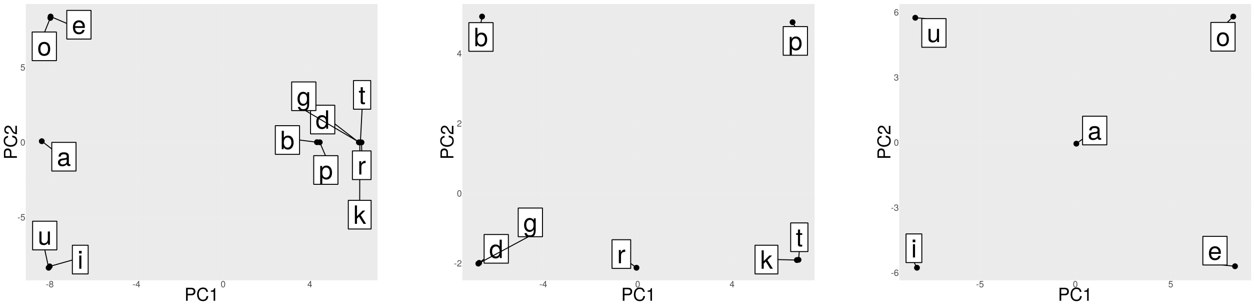
We can then define a new matrix M, corresponding to Table 4, containing our PPMI-normalised values, where

$$M(s,c) = \max\Big(\log_2 \frac{P(s,c)}{P(s)P(c)}, 0\Big) \tag{6}$$

4.4 PPMI Vector Embeddings of Parupa

A challenge in dealing with high dimensional spaces is visualising the data. Here I use Principal Component Analysis (PCA) (Hotelling, 1933), which geometrically projects points in a space onto a smaller set of dimensions. These dimensions are chosen such that the variance of the projected data is maximised. This technique is useful for reducing high dimensional spaces to two or three dimensions so they can be visualised. It will also be of crucial importance in the clustering stage described in Section 5, and a more detailed description will be given there. Here it is simply used to visualise the Parupa embeddings.

[11] This result appears to be at odds with the idea that negative information is a crucial component for learning grammars (e.g., Trubetzkoy, 1939; Hayes & Wilson, 2008). I suspect that, as for words, the number of coincidentally unattested sequences of sounds overwhelms the number of sequences that are phonotactically illicit. For example, the English CMU pronouncing dictionary is transcribed using 39 phonemes, and contains 27,209 words of length six. There are $39^6 = 3,518,743,761$ possible words of length six that could be generated from an inventory of 39 phonemes. This means that attested six-sound words only make up about 0.0007% of possible six-sound words. Because there are so many unattested sequences, it may be the case that it is more informative to know where sounds do occur than where they do not. I leave a detailed exploration of this as a topic for future research.



(a) All segments

(b) Consonants

(c) Vowels

Figure 3: A PCA visualisation of the vector embeddings of Parupa generated using trigram counts and PPMI normalization.

Figure 3a shows a two-dimensional PCA visualisation of the vector space embedding of Parupa using trigram counts and PPMI weighting. Here we can see that the vowel/consonant distinction is clear along PC1, and vowel height is reflected on PC2.[12]

Figures 3b and 3c show PCAs generated using only rows of M corresponding to consonants and vowels respectively. For the consonants, the distinction between sounds that must precede high vowels and sounds that must precede mid vowels is reflected in PC1, while the distinction between sounds that can begin a word and sounds that cannot is reflected in PC2. For the vowels, the height distinction is reflected on PC1, while the backness distinction is reflected on PC2. Note the intermediate position of /r/ and /a/ in the plots, reflecting their shared distributions within the consonant and vowel classes.

PCA visualisations must be interpreted with caution, since they generally lose information present in the full space. In the simple case of Parupa, however, it seems clear that there should be sufficient information in the vector embeddings to retrieve the intended classes.

5 Finding classes using PCA and k-means clustering

Once we have normalised vector embeddings of the sounds in our corpus, we need a way to extract phonological classes from the space. It is intractable to consider every possible set of classes, since given an alphabet Σ, there are $2^{|\Sigma|}$ possible classes, and hence $2^{2^{|\Sigma|}}$ sets of classes that could be chosen. One approach to generating a reasonable set of candidate classes is using *clustering algorithms*. Broadly speaking, such algorithms attempt to assign each point in a data set to one or more clusters, such that the points in each cluster are more similar to other points in the cluster by some criterion than to points outside of the cluster.

Many clustering algorithms with different properties and assumptions have been proposed (Aggarwal & Reddy, 2013), but the nature of the current task imposes several restrictions on the type of algorithm that should be used.

1. It must be *unsupervised*, meaning that the algorithm requires no access to training data (i.e., sounds that have already been assigned to classes).

[12] The reader should keep in mind that referring to a phonetic property here is a shorthand for referring to particular aspect of the distribution, since there is no notion of phonetic substance in this model.

2. It must not require the number of classes to be specified in advance.

3. It must allow *multiple class membership*. This is analogous to saying that it must allow a set of sounds to be partitioned in multiple ways. In Parupa, for example, /i/ patterns as both a front vowel and a high vowel.

4. Distributional evidence for class membership may be present only in some contexts. For example, the high/mid vowel distinction in Parupa is signaled only by the preceding consonant, while the front/back distinction is apparent only from the preceding and following vowels. A suitable clustering algorithm should be able to look at meaningful subsets of all contexts when clustering sounds.

There are clustering algorithms that meet these criteria, particularly certain *subspace clustering* algorithms (Müller et al., 2009), but properties of the data considered here make them difficult to apply for practical reasons. First, these algorithms are generally difficult to parameterise in a principled way, requiring assumptions about the number of clusters or the distributional properties of the data. Second, phonological data by definition has no outliers: even if a sound is systematically underspecified for most features (e.g., Lahiri & Marslen-Wilson, 1991), it should still belong to at least one cluster, such as the class of consonants or vowels. Finally, our data consist of a small number of points, one per sound, and a large number of dimensions, one per context. Most clustering algorithms are optimised to handle the opposite situation well, and this leads to severe efficiency issues.

In light of these problems, I propose a clustering technique that is well suited to this task. It works by recursively applying Principal Component Analysis and one-dimensional k-means clustering. The next sections will show that this combination allows for multiple partitions of the same set of data, while simultaneously exploiting the generally hierarchical structure of phonological classes.

5.1 Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction technique. It takes a matrix consisting of points in some number of possibly correlated dimensions and geometrically projects that data onto a set of new, uncorrelated dimensions called *principal directions*. These principal directions are linear combinations of the original dimensions. The dimensions of the data after they have been projected onto the principal directions are called *principal components*.

The number of principal components is the minimum of $m - 1$ and n, where m is the number of rows in the data set and n is the original number of dimensions. Principal components are ordered descending by proportion of variance captured, with PC1 capturing the most variance, followed by PC2, and so on. This has several useful consequences:

1. To reduce a data set to p dimensions while minimising the amount of information lost, we can simply choose the first p principal components (e.g., in Section 4, I chose the first two PCs to visualise the Parupa embeddings in two dimensions).

2. Because we know how much variance each principal component captures, we can choose the number of new dimensions using a variance-based criterion. This could be choosing

the number of dimensions required to capture some percentage of the original variance, or choosing only dimensions whose variance contribution exceeds some threshold.

Performing PCA on an $m \times n$ input matrix X can be done by calculating the eigenvalues and eigenvectors of the covariance matrix of X, which I will notate as K. K is a symmetrical $n \times n$ matrix whose cells $K(i, j)$ encode the covariance between the ith and jth dimensions of X. The eigenvectors of K are the principal directions, and the magnitude of the ith eigenvalue of K, λ_i, reflects the amount of variance in the original data that is captured by the ith principal component.

Rather than calculating the covariance matrix directly, it is common to use *singular value decomposition* (SVD) to do PCA. This approach is used by the Python SKLEARN library (Pedregosa et al., 2011), which I use here. SVD is a type of matrix decomposition that factors a rectangular matrix X into the product of three matrices USV^T, where U and V are unitary matrices whose columns are called the left and right *singular vectors* respectively, and S is a diagonal matrix of *singular values*. SVD is typically computed using iterative numerical methods (e.g., Golub & Reinsch, 1970).

The ith singular value s_i is related to the ith eigenvalue of K by $\lambda_i = s_i^2/(n-1)$, and the columns of V are the principal directions (i.e., the eigenvectors of K). The principal components (projections of X onto the principal directions) can be calculated as XV or equivalently as US. Note crucially that these relationships only hold when the input matrix X has been *centered*: the mean of each dimension has been subtracted from each value in that dimension, resulting in all dimensions having a mean of 0.

The algorithm applies PCA to the matrix M_c, which contains centered, normalised vector embeddings of sounds calculated from a corpus. Each element $M_c(i, j)$ is calculated from M as follows:

$$M_c(i,j) = M(i,j) - \frac{1}{n}\sum_{k=1}^{n} M(k,j) \tag{7}$$

I will notate the matrix of PCs generated from applying PCA to M_c as A, where

$$M_c = USV^T \tag{8}$$

and

$$A = M_c V \tag{9}$$

PCA is useful for clustering phonological data for several reasons: first, because our matrix consists of few rows and many dimensions, its dimensions are highly correlated. Applying PCA reduces the matrix to a set of uncorrelated dimensions, which makes interpretation more straightforward. Second, PCA helps to highlight robust sources of variance while reducing noise. Finally, the resulting principal components provide some insight into the different ways to partition a set of sounds. Consider again Figure 3c. PC1, which captures the largest proportion of the original variance, shows the distinction between high and mid vowels while revealing little about the front/back vowel distinction. This distinction is apparent in PC2, however. Thus looking at different principal components *individually* has the potential to expose multiple ways to partition a single set of sounds.

The generalisation performed by PCA is achieved in various ways by previous work. Silfverberg et al. (2018) use PCA in an analogous way, while Calderone (2009) uses independent component analysis, which is closely related. Nazarov (2014, 2016) uses interaction between constraint selection and grammar induction to achieve a similar outcome: constraint selection chooses particular contexts to focus on, and feature induction allows multiple contexts to be aggregated into a single context by inferring a phonological class.

5.2 *k*-means clustering

Given a principal component, we would like to determine how many classes the distribution of sounds suggests. In Figure 3b, for example, a visual inspection suggests PC1 should be grouped into three classes: {b,d,g}, {r}, and {p,t,k}, while PC2 should be grouped into two classes: {b,p} and {d,g,r,k,t}. k-means clustering can be used to group a set of points into k clusters by assigning points to clusters in such a way that the total distance from each point to its cluster centroid is minimised (MacQueen, 1967). That is, we assign our data points $x_1 \dots x_m$ to clusters $\mathbf{c} = c_1 \dots c_k$ such that we minimise the within-cluster sum of squares WCSS:

$$\text{WCSS} = \sum_{i=1}^{k} \sum_{x \in c_i} \|x - \mu_i\|^2 \tag{10}$$

where μ_i is the cluster centroid, c_i and $\|x - \mu_i\|$ is the Euclidean distance between x and μ_i.

In order to determine the optimal value of k, information theoretic measures such as the Akaike Information Criterion (AIC) (Akaike, 1974) or Bayesian Information Criterion (BIC) (Schwarz, 1978) can be used. These measures attempt to strike a balance between model complexity and model fit by penalising more complex models (in this case, higher values of k) while rewarding fit to the data (in this case, distances from the cluster centroids). I use a custom Python implementation of the X-means algorithm (Pelleg & Moore, 2000), based on the R code provided by Wang and Song (2011), which finds the optimal number of clusters using the BIC as an evaluation metric. When applied to PC1 and PC2 of the set of consonants discussed in the previous paragraph, this algorithm finds exactly the expected classes: namely {b,d,g}, {r}, and {p,t,k} on PC1, and {b,p} and {d,g,r,k,t} on PC2.

Readers familiar with clustering techniques might find it odd that clustering is done over single principal components rather than all dimensions, whether these be the original dimensions representing specific contexts, or the reduced dimensions after PCA is performed. This is a sensible choice because of the properties of the vector embeddings.

First, the columns in the vector space are massively redundant. Each principal component in a PCA can be thought of as an aggregation of the information in a correlated set of columns in the original data. Put another way, PCA does some of the work of finding meaningful subspaces over which clustering is likely to be effective. Thus each principal component can be thought of as representing some number of dimensions in the original space.

Additionally, clustering over individual principal components rather than sets of principal components allows us to find broad classes in the space that might otherwise be overlooked. This is apparent when examining Figure 3c: clustering over PC1 and PC2 separately allows us to find distinct partitions of the vowel space based on height and backness. If PC1 and PC2 were considered together, the only likely clusterings would be either a single cluster containing all vowels, missing the class structure completely, or one cluster per sound. The latter is equivalent to finding classes that reflect the *intersections* of different height and backness values, but overlooks the broader class structure from which these subclasses are generated. Finding such classes is a property that many subspace clustering algorithms have, but, as described above, these algorithms are generally unsuited to this type of data. Clustering over single principal components is simple way to achieve this property while mitigating many of these issues.

Since principal components capture increasingly less and less of the total variance of the data, we may wish to cluster on only a subset of them that capture robust patterns. I return to this issue in Section 5.4.

Goldsmith and Xanthos (2009)'s spectral clustering algorithm is similar to the method presented here, in that it clusters sounds one-dimensionally along a single eigenvector by choosing an optimal cut point. Their use of maximum entropy hidden Markov models also performs a kind of one-dimensional clustering on emission probability ratios, setting a threshold of 0 as the boundary between clusters. Powers (1997) and Mirea and Bicknell (2019) both use hierarchical clustering to extract classes from embeddings. Hierarchical clustering is simple, but not well-suited to phonological class discovery: it cannot find multiple partitions of the same set of sounds, and requires the number of classes to be decided by an analyst. Finally, Nazarov (2014, 2016) uses Gaussian mixture models (GMMs) to do one-dimensional clustering on the embeddings of segments in a context. GMMs assume that the data was generated by some number of underlying Gaussian distributions, and attempts to learn the parameters of these distributions and provide a probabilistic assignment of points to each. This is perhaps the most similar approach to the one taken here, since k-means can be considered a special case of GMM that does hard cluster assignment and does not take into account the (co-)variance of its discovered clusters.

5.3 Recursively traversing the set of classes

The final component of this clustering algorithm leverages the generally hierarchical nature of phonological classes. In many cases a distinction is only relevant to segments in a particular class: for example, the feature $\left[\ +/-\ \text{strident}\ \right]$ is only relevant for coronal fricatives and affricates. Thus patterns that do not contribute a great deal to the variance of the entire set of sounds might become more apparent when only a subset of the sounds is considered. In order to exploit this hierarchical structure and detect such classes, this clustering algorithm is called recursively on the sets of classes that are discovered.

Suppose we perform k-means clustering on the first principal component of A, and discover two classes, c_1 and c_2. The recursive traversal consists of

1. Creating a matrix M' containing just the rows in M that correspond to sounds in c_1.

2. Centering and performing PCA on M', producing a new matrix A' whose columns are the principal components.

3. Performing k-means clustering on the individual columns of A'.

This process will then be repeated on c_2, on any classes discovered when clustering on A', and on the remaining principal components of A. Recursive traversal stops when (a) M'

consists of only a single row (i.e., we have a cluster containing just one sound); or (b) the clustering step produces a single cluster. Note that the original normalised embedding M is always used as the starting point: recursive traversal does not recalculate embeddings, but simply performs PCA and clustering on a subset of the rows in this matrix.

5.4 Putting it all together

To summarise, this algorithm runs Principal Component Analysis on a matrix of normalised vector embeddings of sounds and attempts to find clusters on the most informative principal components. For each cluster found, the algorithm is recursively applied to that cluster to find additional subclusters. Considering multiple principal components for each set of sounds allows multiple partitions of these sets, and the recursive character allows it to exploit the generally hierarchical nature of phonological classes to discover more subtle class distinctions.

The steps of the algorithm and the necessary parameters are detailed below:

1. Calculate the normalised vector embedding matrix M.
2. Perform Principal Component Analysis using M as input, producing matrix A.
3. For each principal component, or column of A, $A_{:,i}$, where $1 \leq i \leq p$:
   (a) Cluster the sounds in $A_{:,i}$ into between 1 and k clusters.
   (b) If more than one cluster is found, run steps 2 and 3 again on each cluster that has more than one member, using as input the matrix M', which contains only the rows of M corresponding to the sounds found in the cluster.
4. Return the clusters that were found by this and all recursive calls.

The two parameters that must be set here are p, the number of principal components we consider for each input, and k, the maximum number of clusters we attempt to partition each principal component into.

I choose k by assuming the typical properties of phonological feature systems, where a class is either $+$, $-$, or 0 (unspecified) for a particular feature. This suggests we should partition each principal component into either one (no distinction), two (a $+/-$ or $+/0$ distinction, as in PC2 in Figure 3b), or three (a $+/-/0$ distinction, as in PC1 in Figure 3b). Thus, setting $k = 3$ seems like a principled choice.

When choosing p, we want to select only those principal components that are sufficiently informative. If p is too high, principal components that contain mostly noise will be included and result in spurious classes being detected. If p is too low, important classes may be overlooked. There have been many proposals for how to choose the number of components (e.g., Minka, 2000; Cangelosi & Goriely, 2007). Here I use a variant of the relatively simple Kaiser stopping criterion (Kaiser, 1958). This takes only the principal components that account for above-average variance (i.e., whose eigenvalues are greater than the average of the eigenvalues of all principal components). This criterion is simple to calculate and works well in practice here.

In general, however, choosing how many components to use can be more of an art than a science. It is useful to consider this as a parameter that might be tuned for different purposes (e.g., we might want to consider less robustly-attested classes with the intention of later evaluating them on phonetic grounds). Increasing or decreasing the number of components used has the effect of increasing or decreasing the algorithm's sensitivity to noise, and determines how robust a pattern must be to be retrieved. I will return to this point in Section 6.[13]

5.5 Simplifying assumptions

I make two simplifying assumptions when applying the algorithm to the data presented in the rest of the paper: I restrict partitions of the full set of sounds to be into a maximum of two classes, and to only use the first principal component. Assuming that the most obvious partition is between consonants and vowels, this is equivalent to stipulating that the first partition of a segmental inventory must be into these two categories, and that subclasses must be contained entirely in the set of vowels or the set of consonants. This potentially misses certain classes that span both sets (like the class of $\left[\text{ +voice }\right]$ sounds, or classes containing vowels and glides, such as {i, j} and {u, w}, for example), but greatly reduces the number of classes generated and facilitates interpretation.

Note that previous work makes similar assumptions about how the full set of sounds should be partitioned: the spectral clustering algorithm in Goldsmith and Xanthos (2009) explicitly assumes a partition into two classes, while the maximum entropy hidden Markov model algorithm assumes as many classes as there are states. Studies that use hierarchical clustering (Powers, 1997; Mirea & Bicknell, 2019) also implicitly make this assumption, as hierarchical clustering always performs binary splits. Thus, relative to past work, this assumption does not provide undue guidance towards a clean consonant-vowel division.

In fact, lifting the restriction that the first principal component must be clustered into at most two classes produces different results only in the cases of Parupa and French. In both cases the full set of sounds is partitioned into three classes instead of two, and only in French are the consonant and vowel classes placed in overlapping partitions. Allowing clustering of the full set of sounds on other principal components produces additional classes in every case, but does not affect the classes that are retrieved from the first principal component. See Appendix A.3 for additional discussion.

In the next section, I present the results of the algorithm applied to Parupa.

5.6 Running the algorithm on Parupa

Recursively applying PCA and k-means clustering to the Parupa vector embeddings detailed in Section 4 produces the classes in Table 5:

All of the expected classes indicated in Figure 1 are present in this set (i.e., those classes that might be expected from a pencil-and-paper analysis). Although there are classes that do not obviously participate in the phonotactic restrictions described above, these are derivable

[13] It may be the case that this parameter can be chosen based on phonological criteria by looking at how many different partitions of a single set of sounds are typical in natural languages.

{i, e, u, o, a}	**{p, t, k, b, d, g, r}**
{i, u}	**{b, d, g}**
{e, o}	**{p, t, k}**
{i, e}	**{p, b}**
{u, o}	{t, k, d, g, r}
{a}	{d, g}
	{k, t}
	{p}
	{b}
	{r}

Table 5: Classes learned from Parupa. Bolded classes indicate classes that might be expected from a pencil-and-paper analysis.

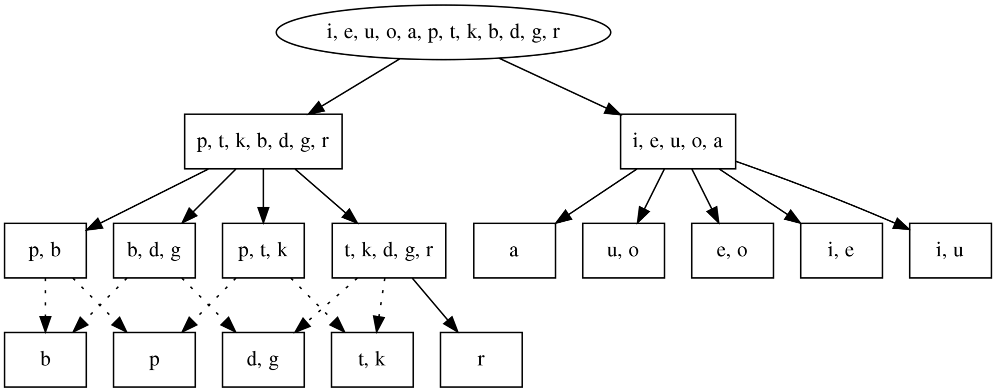


Figure 4: The classes retrieved from Parupa. Arrows indicate a parent/child relationship between classes. Dotted arrows indicate that a class is the intersection of two or more parents.

from the expected classes: e.g., {t, k, d, g, r} is the class of non-word-initial consonants, {d, g} is the class of non-word-initial consonants that can precede mid vowels, {t, k} is the class of non-word-initial consonants that can precede high vowels, etc. The hierarchical relationship between these classes is shown in Figure 4, which was generated using code from C. Mayer and Daland (in press).

These diagrams are used throughout the paper, and do not reflect the order in which the classes were retrieved by the algorithm. Rather, they arrange the classes in a hierarchical structure, where arrows between classes represent a parent-child relationship (i.e. the child class is a proper subset of the parent class, and there is no other class that intervenes between the two). Dotted arrows indicate that a class is the intersection of two or more parents. These diagrams give a sense of the overall relationship between the classes retrieved by the algorithm.

Note that the singleton classes consisting of individual segments are not retrieved in general. This is the consequence of the *k*-means clustering component deciding that no partition of a class into two or three classes is justified. This is not of great concern, however, since the assumption of a segmental representation necessarily implies that singleton classes are available to the learner. These may be appended to the list of retrieved classes if desired.

This algorithm performs well on Parupa, successfully retrieving all of the intended classes, including those that involve partitioning sets of sounds in multiple ways.

5.7 Evaluating the robustness of the algorithm on Noisy Parupa

Parupa is a pathologically tidy language: its phonotactic constraints are never violated. Although the algorithm does well on retrieving the class structure to which these constraints are sensitive, no natural language is so well behaved. In order to evaluate how well the algorithm handles noise, I examine its performance on a more unruly variant of Parupa: Noisy Parupa.

Noisy Parupa is identical to Parupa, except that some percentage of the generated word tokens not subject to the phonotactic generalisations described in Section 3: these tokens constitute *noise* with respect to these generalisations (see, e.g., Archangeli et al., 2011). Noisy words still maintain a CV syllable structure, but the consonants and vowels in each position are chosen with uniform probability from the full sets of consonants and vowels. Examples of Noisy Parupa words are shown in Table 6, and the Hidden Markov Model for generating noisy words is shown in Figure 5. Transition probabilities were chosen so that the average word length is still three syllables.

gogi	kikuka
reku	toredi
duke	pipu

Table 6: Some Noisy Parupa words.

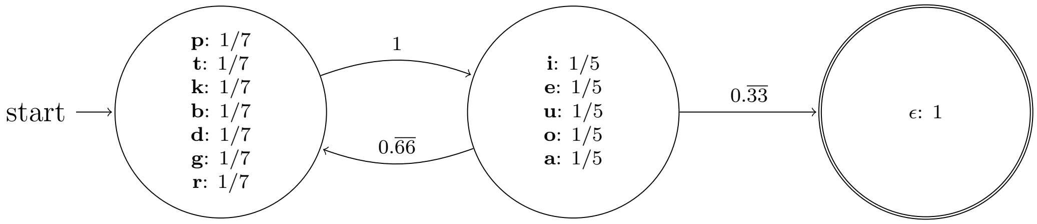


Figure 5: The Hidden Markov Model used to generate Noisy Parupa words. Edges are labeled with their transition probabilities, and states are labeled with their segment emission probabilities. ϵ is the empty string.

A *noise parameter* determines what percentage of the words are noisy. Standard Parupa can be thought of as a special case where this parameter is set to 0. As the value of this parameter increases, the algorithm should have more difficulty finding the expected phonological classes.

The model was tested on 110 corpora. The noise parameter was varied from 0% to 100% in increments of 10%, and ten corpora were generated for each parameter value.

Figure 6 shows the median number of expected and unexpected classes found by the algorithm as the percentage of noisy words increases. The expected classes are defined as

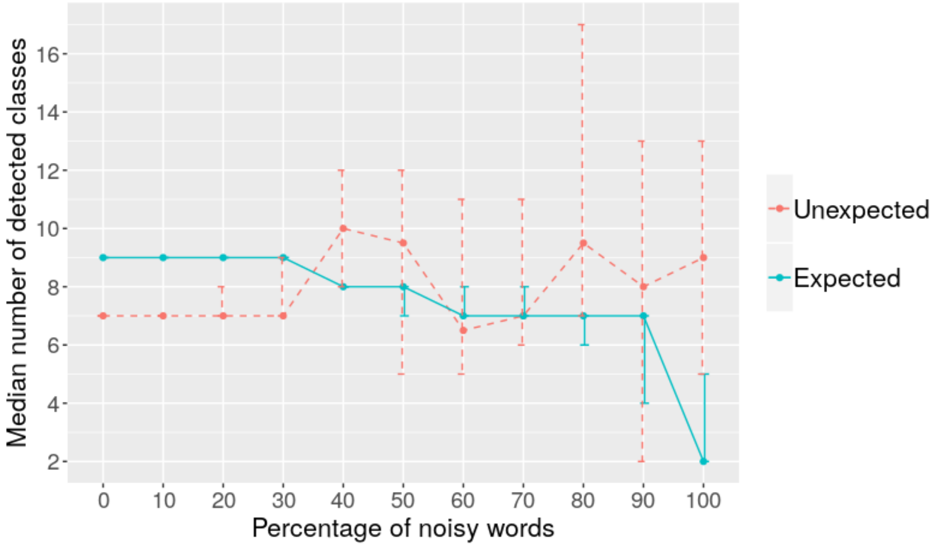


Figure 6: A plot of the median number of expected and unexpected classes found by the algorithm as the percentage of noisy words increases. Error bars span the minimum and maximum number of classes retrieved from a corpus at that noise level.

exactly the classes in Figure 1. The number of unexpected classes varies stochastically with the contents of each corpus, but the number of expected classes found remains reasonably high until 100% noise. From 40% to 70% noise, the expected classes that are not detected are either {p, t, k}, {b, d, g}, or both. In about half the cases (19/40) the unexpected classes include {p, t, k, r} and/or {b, d, g, r}.[14] In 20 of the remaining 21 cases, the sets {p, t} and/or {b, g} are recovered. This indicates that the pattern is still detected to some extent, although the participating classes are less clear due to the increase in noise.

From 80% to 90% noise, the algorithm reliably fails to detect the classes {p, t, k} and {b, d, g}, while occasionally also overlooking other classes: {p, b} (4/20), {u, o} (3/20), {i, u} (1/20), {i, e} (3/20) and {e, o} (1/20).

Finally, at 100% noise, the consonants and vowels are the only classes reflected in the distribution, and these are successfully retrieved in all cases. The other expected classes that are sometimes retrieved are the result of chance.

The results of the algorithm on Noisy Parupa suggest that it is robust to noise. All expected classes are discovered in up to 30% noise, and even up to 90% noise most of the expected classes are still found. Even when expected classes are lost at higher noise levels, these are often still reflected in aspects of the unexpected classes that are found.[15]

In the next section I examine the results of the algorithm on several natural language corpora.

[14] When both these classes are present, they will necessarily have been discovered while clustering on separate principal components, since they overlap.

[15] An anonymous reviewer wonders, following Archangeli et al. (2011), if certain types of exceptions are more disruptive to the operation of this algorithm than others, and whether these correspond to what we see in natural language. I leave this as an interesting area for future research.

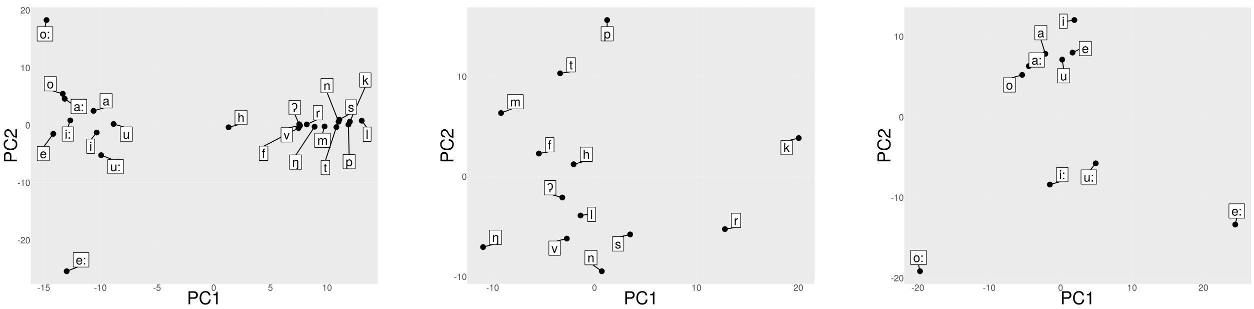
6 Testing the algorithm on real language data

In this section I show how the algorithm performs on several real languages: Samoan, English, French, and Finnish. I include Samoan because it has a relatively small segmental inventory and fairly restrictive phonotactics, providing a simple test case. English, French, and Finnish are included for continuity with previous studies (Goldsmith & Xanthos, 2009; Calderone, 2009; Silfverberg et al., 2018; Mirea & Bicknell, 2019).

For the data in this section, I will vary the parameter that determines how many principal components of a class are considered. Recall that the default is to cluster only on principal components that account for a greater than average proportion of the variance in the data. I scale this by multiplying it by a factor (so, for example, we might only consider principal components that account for two times the average variance). This is useful because of the varying levels of distributional noise in different data sets. It is important to remember that all classes returned with a higher threshold will also be returned when the threshold is lowered, but in the latter case some additional, less robust classes will be returned as well. I vary this parameter primarily to keep the number of discovered classes suitably small for expositional purposes. If the parameter is not specified in the text, a default of 1 is used. Some discussion of the effect of varying this parameter can be found in Appendix A.4.

6.1 Samoan

The Samoan corpus was generated from a Samoan dictionary (Milner, 1993) and contained 4226 headwords.[16] This is an orthographic representation of Samoan, but there is a close correspondence between orthography and pronunciation. Symbols have been converted to IPA for clarity. Figure 7 visualises the vector embedding of Samoan.



(a) All segments **(b)** Consonants **(c)** Vowels

Figure 7: A PCA visualisation of the vector embeddings of Samoan.

The retrieved classes are shown in Figure 8. The algorithm was able to successfully distinguish between consonants and vowels. It also makes a rough distinction between long and short vowels, although /a:/ is grouped with the short vowels. Finally, the set of short vowels and /a:/ is split into low and non-low, while the set of long vowels is partitioned and high and mid sets. There does not appear to be sufficient distributional information to make

[16] Thanks to Kie Zuraw for providing this data.

any partitions of the set of consonants. Lowering the variance threshold for which principal components to consider did not result in more classes being learned.

The patterning of /a:/ with the short vowels is surprising, but can be explained by examining its distribution. While VV sequences are quite common in Samoan (1808 occurrences in the corpus), VV:, V:V, and V:V: sequences are rarer (226 total occurrences). In 171 of these 226 occurrences, the long vowel is /a:/. Thus /a:/ patterns more like a short vowel than a long vowel with respect to its distribution in vowel sequences, and the algorithm reflects that in its discovered classes. This is an example of a class that cannot be captured using phonetic features, but is valid in the sense that it is salient in the distribution of the language.[17]

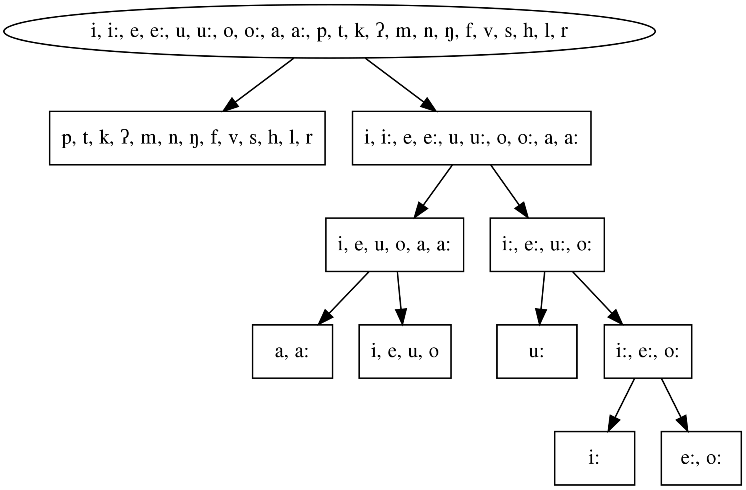

i, i:, e, e:, u, u:, o, o:, a, a:, p, t, k, ʔ, m, n, ŋ, f, v, s, h, l, r
p, t, k, ʔ, m, n, ŋ, f, v, s, h, l, r
i, i:, e, e:, u, u:, o, o:, a, a:
i, e, u, o, a, a:
i:, e:, u:, o:
a, a:
i, e, u, o
u:
i:, e:, o:
i:
e:, o:

Figure 8: Retrieved classes from Samoan. Arrows indicate parent/child relationships.

To examine whether the trigram window is too small to capture information that might allow the consonants to be grouped, I also ran the algorithm on Samoan with the vowels removed. This should allow it to better capture any word-level co-occurrence restrictions that might differentiate groups of consonants (e.g., McCarthy, 1986; Coetzee & Pater, 2008). A PCA of the resulting vector embedding of the Samoan consonants is shown in Figure 9.

I report results from running the algorithm with a scaling factor of 1.3 on the variance threshold (i.e., only principal components with at least 1.3 times the average variance were considered). The constraint that the initial partition of the set of sounds must be in two was also removed, because the consonant/vowel distinction is no longer relevant for this data set. This resulted in the classes shown in Figure 10. Here /r/ and /k/ are clearly set apart from the other consonants. These sounds are relatively uncommon in Samoan, being found predominantly in loanwords, and this is reflected in their distribution. We might suggest, following Chomsky and Halle (1968), that these sounds are characterised by something like a $\left[+ \text{foreign} \right]$ feature.

Aside from the marginal status of /k/ and /r/ in Samoan phonology, it has hard to justify these classes in a linguistically satisfying way. The additional classes found when

[17] An anonymous reviewer wonders whether this pattern may have emerged from perceptual expectations for low vowels to be longer than non-low vowels, making /a:/ more perceptually similar to other short vowels.

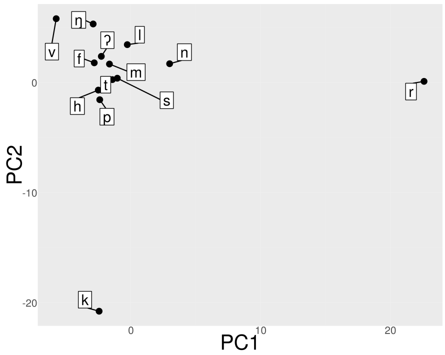


Figure 9: A PCA of the Samoan consonants from a corpus without vowels.

the variance threshold was lowered were similarly arbitrary. This suggests that consonant co-occurrence restrictions reflect little more than the special status of /k/ and /r/. Samoan is known to have phonotactic restrictions on root forms (e.g., Alderete & Bradshaw, 2013), and it is possible that running the algorithm on roots rather than headwords would make these patterns more detectable.

Given Samoan's strict (C)V phonotactics, it is perhaps not surprising that distribution yielded few distinctions in the set of consonants. This raises an interesting question of whether speakers of Samoan actually use phonological features to categorise consonants (other than perhaps /k/ and /r/), or if they are treated as atomic segments. I turn now to English, where the presence of consonant clusters may give us a better chance of retrieving additional phonological information.

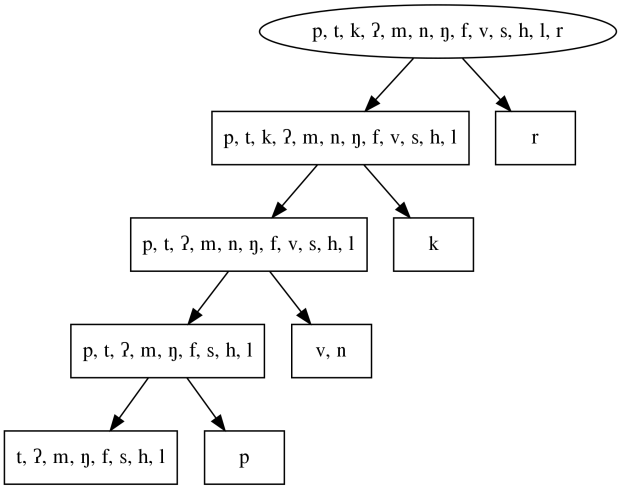
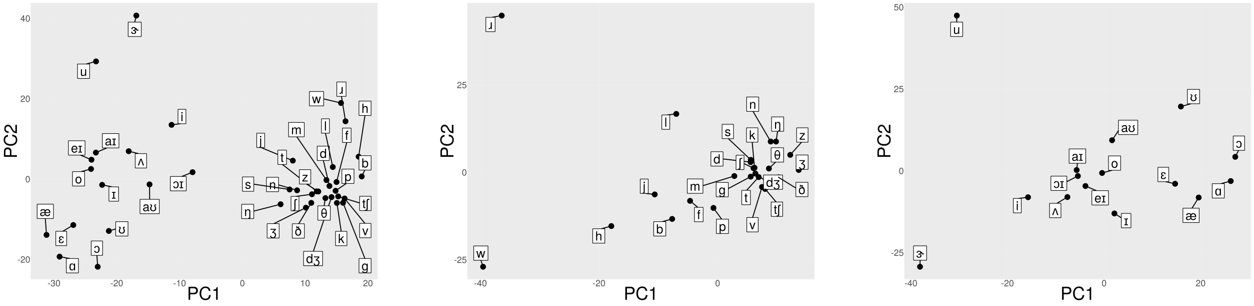

p, t, k, ʔ, m, n, ŋ, f, v, s, h, l, r
p, t, k, ʔ, m, n, ŋ, f, v, s, h, l
r
p, t, ʔ, m, n, ŋ, f, v, s, h, l
k
p, t, ʔ, m, ŋ, f, s, h, l
v, n
t, ʔ, m, ŋ, f, s, h, l
p

Figure 10: Retrieved classes from Samoan with no vowels. Arrows indicate parent/child relationships.

6.2 English

The English corpus was generated from the CMU pronouncing dictionary,[18] which is phonemically transcribed. Diphthongs are treated as single vowels, rather than VV sequences. Only words with a frequency of at least 1 in the CELEX database were included (Baayen et al., 1995), and some manual error correction was performed.[19] The resulting corpus consisted of 26,552 word types. Figure 11 visualises the vector embedding of English.



(a) All segments **(b)** Consonants **(c)** Vowels

Figure 11: A PCA visualisation of the vector embeddings of English.

I report results from using a scaling factor of 1.1 on the variance threshold. The retrieved classes are shown in Figure 12. The sets of vowels and consonants are correctly retrieved. Within the consonants, there is an eventual distinction between the class of coronal obstruents, nasals, and /v/, and all other consonants. The class of velar obstruents {k, g} is recovered, as well as the class of labial obstruents {p, b, f} minus /v/, and the set of labial approximants {w, ɹ}. The vowels are more difficult to interpret, but there are splits that are suggestive of the tense vs. lax distinction.

In a language like Samoan, with a small number of sounds and restricted syllable structure, it is relatively simple to identify the specific distributional properties that lead to a particular class being detected. More phonotactically complex languages like English are not as straightforward. We can, however, get a sense of what distributional information is reflected in a principal component by looking at how contexts are linearly combined to produce that principal component. A context's coefficient in this linear combination reflects the correlation between the principal component and the context. Therefore, to get a sense of what information is being aggregated by a principal component, we can impressionistically look for patterns in the contexts that are highly positively and negatively correlated with that principal component. It is important to remember that these impressionistic descriptions only provide a partial characterisation of a principal component.

There is not enough space here for a detailed exploration of the distributional properties that define each of the classes discovered here, but I will briefly look at the topmost splits within the consonant and vowel classes. The topmost split in the consonant class is between three classes: {w, ɹ}, {p, b, f, h, j, l}, and the remaining consonants. This split is discovered on PC1 of the consonant embeddings, which is visible in Figure 11b. This principal

[18] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

[19] See `http://linguistics.ucla.edu/people/hayes/EnglishPhonologySearch`.

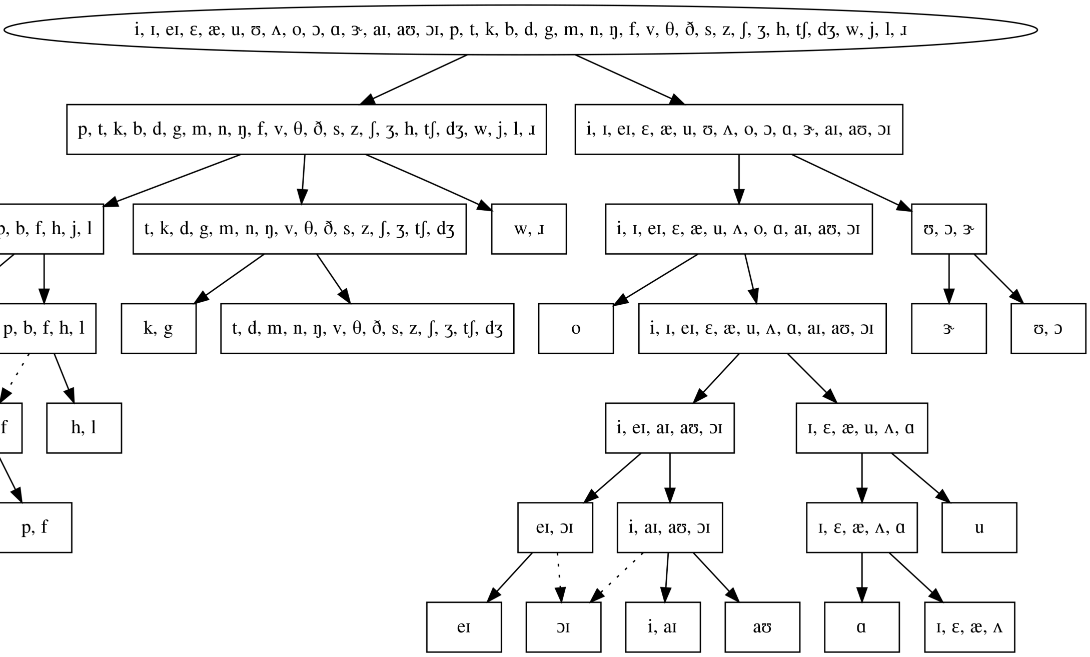

i, ɪ, eɪ, ɛ, æ, u, ʊ, ʌ, o, ɔ, ɑ, ɝ, aɪ, aʊ, ɔɪ, p, t, k, b, d, g, m, n, ŋ, f, v, θ, ð, s, z, ʃ, ʒ, h, tʃ, dʒ, w, j, l, ɹ
p, t, k, b, d, g, m, n, ŋ, f, v, θ, ð, s, z, ʃ, ʒ, h, tʃ, dʒ, w, j, l, ɹ
i, ɪ, eɪ, ɛ, æ, u, ʊ, ʌ, o, ɔ, ɑ, ɝ, aɪ, aʊ, ɔɪ
p, b, f, h, j, l
t, k, d, g, m, n, ŋ, v, θ, ð, s, z, ʃ, ʒ, tʃ, dʒ
w, ɹ
i, ɪ, eɪ, ɛ, æ, u, ʌ, o, ɑ, aɪ, aʊ, ɔɪ
ʊ, ɔ, ɝ
p, b, f, j
p, b, f, h, l
k, g
t, d, m, n, ŋ, v, θ, ð, s, z, ʃ, ʒ, tʃ, dʒ
o
i, ɪ, eɪ, ɛ, æ, u, ʌ, ɑ, aɪ, aʊ, ɔɪ
ɝ
ʊ, ɔ
j
p, b, f
h, l
i, eɪ, aɪ, aʊ, ɔɪ
ɪ, ɛ, æ, u, ʌ, ɑ
b
p, f
eɪ, ɔɪ
i, aɪ, aʊ, ɔɪ
ɪ, ɛ, æ, ʌ, ɑ
u
eɪ
ɔɪ
i, aɪ
aʊ
ɑ
ɪ, ɛ, æ, ʌ

Figure 12: Retrieved classes from English. Arrows indicate parent/child relationships.

component appears to primarily capture whether segments tend to occur before or after a vowel. In the 100 most positively correlated contexts, the target consonant is followed by a vowel in 100% of contexts with a following sound (i.e., contexts with the form $\_s_2s_3$ or $s_1\_s_3$, and is preceded by a consonant or word boundary in 96% of contexts with a preceding sound (i.e., contexts of the form $s_1\_s_3$ or $s_1s_2\_$). Conversely, in the 100 most negatively correlated contexts, the target consonant is more frequently followed by a consonant or word boundary than in the highly correlated contexts (39%), and is almost always preceded by a vowel (97%). This may be interpreted as expressing a gradient preference for onset position, though the presence of /h/ in the intermediate class {p, b, f, h, j, l} indicates that this is not a complete characterisation, since this sound occurs only in onsets.

The topmost split in the vowel class is between the class {ʊ, ɔ, ɝ}, and the remaining vowels. This split is discovered on PC3 of the vowel embeddings, which is visualised in Figure 13. This principal component appears to primarily capture how a vowel sequences with English liquids. In the 100 most positively correlated contexts, the target vowel frequently precedes /ɹ/ (54%) or /l/ (9%), but rarely follows them (/ɹ/: 3%; /l/: 2%; though it frequently follows /w/: 29%). Conversely, in the 100 most negatively correlated contexts, the target vowel frequently follows /ɹ/ (17%) and /l/ (12%), but rarely precedes them (/ɹ/: 0%; /l/: 3%). Thus, broadly speaking, this principal components appears to encode a preference for vowels to precede vs. follow liquids (particularly /ɹ/), with the class {ʊ, ɔ, ɝ} tending to precede them. I leave a more detailed investigation of the distributional properties that give rise to the remaining classes as a topic for future research.

I turn now to French, a language with similarly complex phonotactics to English.

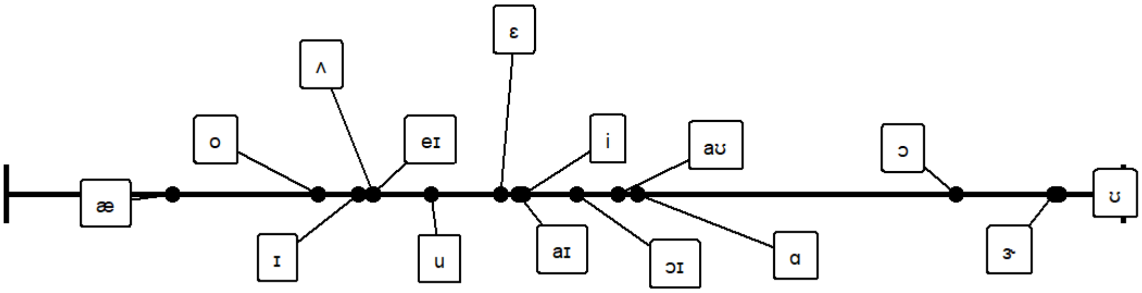
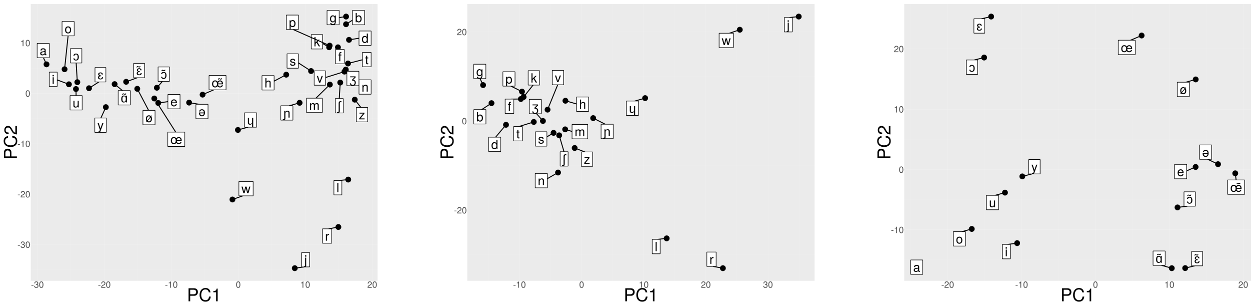


Figure 13: English vowels projected onto PC3.

6.3 French

The French corpus is the one used in Goldsmith and Xanthos (2009).[20] It consists of 21,768 word types in phonemic transcription. Figure 14 visualises the vector embedding of French.



(a) All segments **(b)** Consonants **(c)** Vowels

Figure 14: A PCA visualisation of the vector embeddings of French.

I present results from using a scaling factor of 1.7 on the variance threshold.[21] The retrieved classes are shown in Figure 15. The sets of consonants and vowels are correctly retrieved. Within the consonants, there is a clean split between approximants and non-approximants, and, within the approximants, between liquids and glides. The glides are further split into rounded and unrounded glides. The vowels are more difficult to interpret, but there is a general split between nasalised vowels and vowels with unmarked roundness on one hand, and the remaining vowels on the other (/y/, /e/, and /ə/ are the exceptions).

I will again examine the distributional properties leading to the topmost splits in the consonant and vowel classes. The topmost split in the consonant class is between the set of approximants {w, j, ɥ, l, r} and the remaining consonants. This split is discovered on PC1 of the consonant embeddings, which is shown in Figure 14b. This principal component seems to capture generalisations about syllable structure. In the 100 most positively correlated contexts, the target segment is followed by a vowel in 99% of contexts with a following sound, and preceded by a consonant in 93% of contexts with a preceding sound. The 100

[20] Thanks to John Goldsmith for this data.

[21] The higher value of the scalar here than for other languages indicates that more classes are robustly attested in the distribution of sounds in the French corpus than in other corpora.

most negatively correlated contexts are more likely to be followed by a consonant (43%; most commonly /l/ or /r/) and are generally preceded by a vowel (89%). This appears to capture the patterning of approximants as tending to occur in complex onsets following a non-approximant. This is similar grouping to the one discovered by Goldsmith and Xanthos (2009) using their maximum likelihood hidden Markov model, although the partition here is cleaner: the class they find corresponding to onset-final segments also contains several vowels and other consonants.

The topmost split of the vowel class is between {i, y, ɛ, u, o, ɔ, a} and {e, ø, œ, ə, ɛ̃, œ̃, ɔ̃, ɑ̃}. This split is discovered on PC1 of the vowel embeddings, which is shown in Figure 14c. This principal component seems to capture a tendency for vowels to be adjacent types of consonants, though it is difficult to describe succinctly. In the 100 most positively correlated contexts, the target segment is followed a sonorant in 70% of contexts with a following sound (mostly /l/ and /r/), while in the 100 most negatively correlated contexts, it is followed by an obstruent or word boundary in 91% of relevant contexts. The preceding contexts appear to differ according to place of articulation: of the 100 most positively correlated contexts, the preceding sound is coronal in 32% of relevant contexts, while in the 100 negatively correlated contexts, it is coronal in 70% of relevant contexts. The apparent correlation between preceding coronals and following obstruents is interesting, and I leave a detailed exploration for future work.

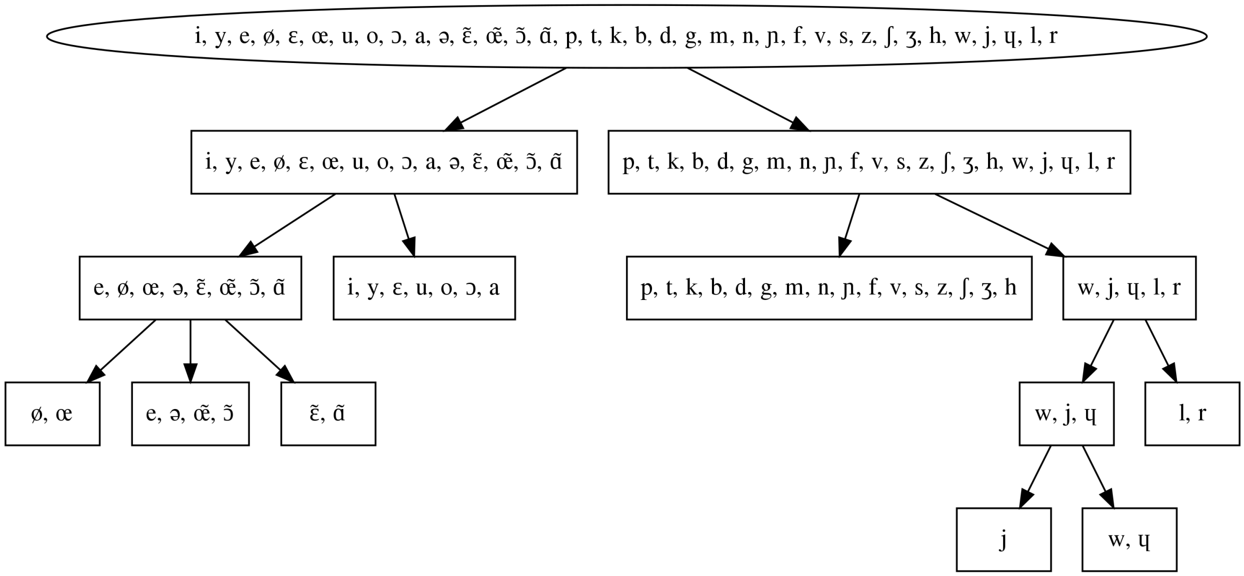

i, y, e, ø, ɛ, œ, u, o, ɔ, a, ə, ɛ̃, œ̃, ɔ̃, ɑ̃, p, t, k, b, d, g, m, n, ɲ, f, v, s, z, ʃ, ʒ, h, w, j, ɥ, l, r
i, y, e, ø, ɛ, œ, u, o, ɔ, a, ə, ɛ̃, œ̃, ɔ̃, ɑ̃
p, t, k, b, d, g, m, n, ɲ, f, v, s, z, ʃ, ʒ, h, w, j, ɥ, l, r
e, ø, œ, ə, ɛ̃, œ̃, ɔ̃, ɑ̃
i, y, ɛ, u, o, ɔ, a
p, t, k, b, d, g, m, n, ɲ, f, v, s, z, ʃ, ʒ, h
w, j, ɥ, l, r
ø, œ
e, ə, œ̃, ɔ̃
ɛ̃, ɑ̃
w, j, ɥ
l, r
j
w, ɥ

Figure 15: Retrieved classes from French. Arrows indicate parent/child relationships.

6.4 Finnish

Finnish is a central example used in Goldsmith and Xanthos (2009). The Finnish vowel harmony system is sensitive to three classes of vowels: the front harmonising vowels {y, ö, ä} (IPA: {y, ø, æ}), the back harmonising vowels {u, o, a}, and the transparent vowels {i, e}. Words tend not to contain both front and back harmonising vowels, and the transparent vowels can co-occur with either class. Goldsmith and Xanthos show that both spectral clustering and hidden Markov models are able to detect these classes (though see Section 7 for additional discussion).

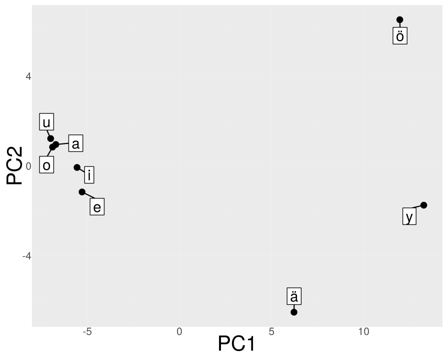


Figure 16: A PCA of the vector embedding of the corpus consisting only of Finnish vowels.

Because the corpus used in Goldsmith and Xanthos (2009) is not publicly available, I use a corpus generated from a word list published by the Institute for the Languages of Finland.[22] Finnish orthography is, with a few exceptions, basically phonemic, and so a written corpus serves as a useful approximation of a phonemic corpus. Words containing characters that are marginally attested (i.e., primarily used in recent loanwords) were excluded.[23] This resulted in the omission of 564 word types, leaving a total of 93,821 word types in the corpus. Long vowels and geminate consonants were represented as VV and CC sequences respectively.

The algorithm was first run on a modified version of the corpus containing only vowels. This mirrors the corpus used in Goldsmith and Xanthos (2009). The vector embedding of this corpus is shown in Figure 16. As with Samoan consonants, the restriction on the number of classes retrieved in the initial partition was lifted. The retrieved classes are shown in Figure 17. The relevant harmony classes are successfully discovered, and, consistent with the results in Goldsmith and Xanthos (2009), the transparent vowels {i, e} pattern more closely with the back vowels than with the front. In addition, classes suggestive of a low/non-low distinction are discovered among the front vowels.

The algorithm was then run on the corpus containing both consonant and vowels. The vector embeddings are shown in Figure 18. I present results from using a scaling factor of 1.2 on the variance threshold. Consonants and vowels were successfully distinguished. Because the focus here is on vowel harmony, I present only the vowel subclasses here.[24] The retrieved

[22] http://kaino.kotus.fi/sanat/nykysuomi/

[23] These characters are c, x, q, z, š, ž, and å. It is important to note that these are uncommon *orthographic representations* of sounds that are more robustly attested by other characters, and their omission is justified under the assumption that we are trying to model speakers' phonological rather than orthographic knowledge. Goldsmith and Xanthos (2009) and Goldsmith and Riggle (2012) remove the same set of characters except for c, x, and q.

[24] The apparent distinction between /f/, /b/, /g/, and /r/ and the other consonants is interesting. Other work using the same corpus has found that certain CV sequences are underrepresented: /fy/, /jø/, /fø/, /gø/, /fæ/, /gy/, /dø/, /gæ/, /bæ/, /by/, and /vø/ (C. Mayer & Nelson, submitted). Most of these contain /f/, /g/, or /b/. The case of /r/ is likely related to syllable structure: although the canonical syllable in Finnish is CVC, CC onsets are possible, particularly in loanwords. The second C in these onsets is frequently /r/, as in *tragedia*, *kromosomi*, or *professori*.

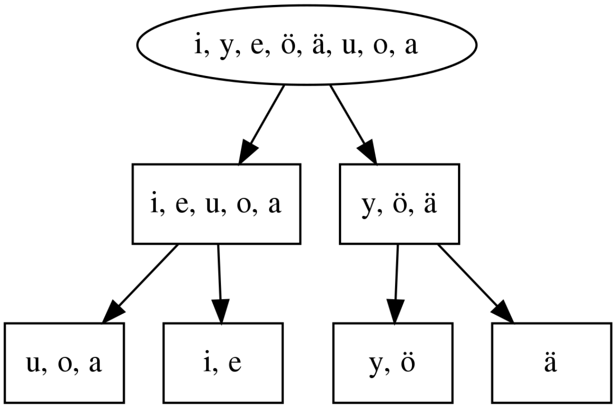
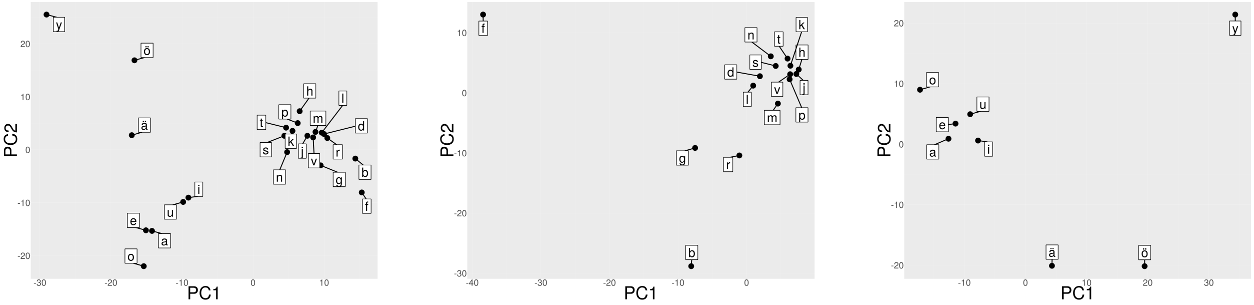


Figure 17: Retrieved classes from the Finnish corpus containing only vowels. Arrows indicate parent/child relationships.



(a) All segments

(b) Consonants

(c) Vowels

Figure 18: A PCA visualisation of the vector embeddings of Finnish.

vowel classes are shown in Figure 19.

Here the front harmonising vowels are differentiated from the transparent and back harmonising vowels, although the split is not as clean as in the vowel-only corpus: the non-high front harmonisers {ö, ä} form their own class, and only later is {y} split off from the remaining vowels. In addition, the distinction between transparent and back harmonising vowels is not made, although the set of both is split into classes suggesting a high/non-high contrast. The loss of clear class distinctions when consonants are added back in is likely a function of the trigram counting method: because Finnish allows consonant clusters, trigrams are not able to capture as much of the vowel co-occurrence as they need to generate the expected classes. More will be said on this point in Section 8.

The algorithm presented here is able to retrieve the correct classes on the corpus containing only vowels, and retrieves classes that capture aspects of the harmony pattern when run on the full corpus. Although the results on the vowel-only corpus seem quite comparable to those in Goldsmith and Xanthos (2009), the next section will discuss why the current results constitute an improvement over Goldsmith and Xanthos, in other ways.

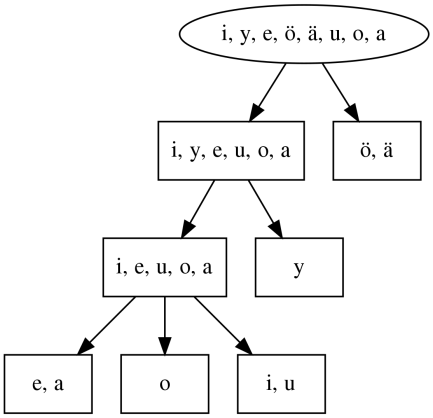


Figure 19: Retrieved vowel classes from the full Finnish corpus. Arrows indicate parent/child relationships.

7 Comparison with past work

A direct comparison of this algorithm to past approaches is difficult because of the lack of a clear quantitative measure of success, the lack of publicly available implementations, and the use of different data sets. Qualitative comparison is possible, however, particularly for the English, French, and Finnish data sets, which are similar or identical to some of those used by previous studies (particularly Goldsmith & Xanthos, 2009; Calderone, 2009). From this perspective, the current algorithm offers several notable improvements.

In all past approaches, except Nazarov (2016), there is no clear method for producing multiple partitions of the same set of sounds (i.e., multiple class membership), and no clear method to partition subsets of the segmental inventory without tailoring the input to include only these subsets. The current algorithm is capable of both these things. Because multiple class membership and privative specification are important properties of most phonological characterisations of a language, these are desirable properties.

The spectral clustering algorithm detailed in Goldsmith and Xanthos (2009) is similar to the current approach in that it decomposes a matrix representation of the distribution of sounds into a single dimension along which sounds may be clustered. There are several aspects in which the current algorithm outperforms spectral clustering. First, spectral clustering is not able to produce an accurate separation of consonants and vowels in any of the languages it is applied to (English, French, and Finnish), although they suggest performance could be improved by considering additional contexts when generating the matrix. The current algorithm was able to produce this separation accurately in all cases tested here. Second, spectral clustering operates by choosing a numerical boundary between clusters that minimises conductance. The decision of how many boundaries to choose must be specified by the analyst (e.g., we choose one boundary when partitioning consonants and vowels, but two when partitioning back, front, and transparent vowels). The algorithm presented here determines the optimal number between one and three clusters.

The maximum entropy hidden Markov model approach, also detailed in Goldsmith and Xanthos (2009), performs better on the consonant and vowel distinction, accurately retrieving it in English and French (Finnish is not presented). Further, it is able to identify vowel classes that participate in harmony processes in Finnish when the input consists only of vowels, and loosely captures a distinction between intervocalic and post-consonantal consonants in French. The algorithm presented here performs at least as well, and, again, does not require that the number of classes be specified in advance, which represents a significant increase in robustness and generalisability.

The independent component analysis method described in Calderone (2009) seems to be able to distinguish between consonants and vowels, as well as suggesting the existence of subclasses within these. However, Calderone does not provide a method for determining exactly how many classes are present: evidence for classes comes from visual inspection of the individual components represented as self-organising maps, which use neural networks to generate two-dimensional grid visualisations based on these components (Kohonen, 2002).

The algorithm presented here could be seen as a more theory-neutral variant of Nazarov (2014, 2016), in that it shares components that perform similar tasks but does not assume a particular structure for the grammar. The toy language on which Nazarov's model is tested contains three phonotactic constraints that refer to a single segment (no word-final /m/), one class of segments (no nasals word-initially), and two classes of segments (no labials between high vowels). Nazarov's algorithm is generally successful in learning constraints that refer to these classes, although less reliably so for the final constraint involving two interacting classes, finding it in only about half of the simulations. I will briefly compare the output of running the current algorithm on Nazarov's data.

When the current algorithm is run on the set of licit words in the toy language, it successfully distinguishes between consonants and vowels, and finds the consonant classes shown in Table 7. The algorithm finds classes corresponding to the labials ({p, b, m}), the nasals that can occur word-finally ({n, ŋ}), and the nasals that cannot occur word-finally ({m}). It misses the full class of nasals because it immediately partitions the set of consonants into three classes: the two nasal sets and the remaining consonants. Thus it succeeds in making a generalisation that Nazarov's algorithm has difficulty with (the set of labials) while not fully generalising to the nasal class, at which Nazarov's algorithm generally succeeds.

{p, t, k, b, d, g, m, n, ŋ}	{p, b, m}
{p, t, k, b, d, g}	{p, b}
{t, k, d, g}	{n, ŋ}
	{m}

Table 7: Consonant classes learned from Nazarov's toy language.

There are two additional considerations when comparing the performance of the two algorithms: first, the toy language employed has strict CVCVC candidate word forms. This is necessary to allow Nazarov's algorithm to construct a probability distribution over possible forms. Since syllable structure is pre-encoded in the candidate set, there is no opportunity for Nazarov's model to find constraints against other types of syllable structures. This means that the simulations do not test whether the learner is able to induce a consonant/vowel distinction. More generally, candidate classes must be restricted to some subset of interest, since the set of all possible words in a language is infinite. This limits the flexibility of the algorithm, since different subsets must be chosen when investigating different properties.

Second, the phonotactic constraints of the toy language are never violated. It is unclear how well Nazarov's algorithm performs on more gradient cases, which are common in natural language phonotactics (e.g., Anttila, 2008; Coetzee & Pater, 2008; Albright, 2009). The algorithm present here functions well as noise is added.

8 Discussion and conclusion

The question of how much and what kinds of information about phonological classes can be retrieved from distributional information is of considerable interest to phonological theory. The algorithm described in this paper accurately retrieves the intended classes from an artificial language with a reasonably complex class structure, even in the presence of distributional noise. When applied to real languages, it successfully distinguishes consonants from vowels in all cases, and makes interpretable distinctions within these categories.

Although the results may seem modest, they are encouraging considering the paucity of the data. No recourse at all is made to the phonetic properties of the sounds, and the representation of the data is simple strings of phonemes. Combining this algorithm with a distributional approach to syllabification (e.g., T. Mayer, 2010) would likely increase performance.

In a more fully realised model of phonological learning, a necessary subsequent step would be to derive a feature system from the learned classes. This step is not treated in this paper, but is discussed in C. Mayer and Daland (in press), where we show that, given certain assumptions about what kinds of featurisations are allowed, a sufficient feature system is derivable from a set of input classes. These two papers may be seen as complementary, and as potential components of a more realistic model of phonological learnability that takes into account other important sources of information, such as phonetic similarity (e.g., Lin, 2005; Mielke, 2012) and alternations (e.g., Peperkamp et al., 2006).

An additional interesting result here is that distributional information is not equally informative for all classes across all languages. Distributional information produced an interpretable partition of vowels in Samoan, but there was little meaningful structure within the class of consonants, even when vowels were removed from the corpus. Indeed, the phonology of the language (including alternations) might not justify any such structure. French and English, on the other hand, had more interpretable results for consonants, but of the two, the result for French more closely match a typical linguistic description. This suggests that the phonotactics of any given language may refer only to a limited set of phonological classes, and accordingly that in some languages, phonotactics may be a stronger indicator of phonological classhood than in others.

This study suggests a variety of possibilities for future research, both in terms of improving the performance of the algorithm and of more broadly exploring the role of distributional learning in phonological acquisition.

A desirable property of the structure of the algorithm presented here is that it is *modular*, in the sense that the four components, vector embedding, normalisation, dimensionality reduction, and clustering, are essentially independent of one another, and can be in principle be modified individually (though different forms of embedding will likely require different methodologies in other components). This general structure, first quantifying similarity between sounds and subsequently using clustering to extract classes, provides a useful conceptual framework from which to approach problems of distributional learning in phonology in general, and lends itself to exploration and iterative improvement.

The counting method employed in the vector embedding step is almost certainly a source of difficulty in the results presented here. Trigrams use a small enough window that long distance dependencies may be overlooked. For the case of the artificial language Parupa, trigram counts were sufficient to capture all phonological constraints in the language, and the model performed accordingly well. It is likely the case that considering additional aspects of context would improve performance on the real languages, although simply increasing the size of the contexts considered in an n-gram model will lead to data sparsity issues. Using sequential neural networks (e.g., Mikolov et al., 2010; Sundermeyer et al., 2012) to generate phoneme embeddings is a particularly promising possibility, since they can produce vector representations of sounds without being explicitly told which features of the context to attend to (Silfverberg et al., 2018; Mirea & Bicknell, 2019). Alternatively, integrating a mechanism for tier projection (e.g., Heinz et al., 2011) into this algorithm based on classes that have already been discovered could help mitigate the limitations of trigram counting.

An additional consideration is that this algorithm makes a fairly broad pass over the language. Meaningful distributional information about a class might be present in only very specific contexts, and this information may be indistinguishable from noise and similarly suppressed by PCA. A principled way of attending to specific contexts, perhaps along the lines of Nazarov (2014, 2016), has the potential to allow more granular classes to be revealed.

Turning to more general considerations, there are many broad questions about the role of distributional learning that could be addressed by experimental work, particularly artificial grammar learning (AGL) experiments. Substantive bias effects (a preference for learning phonetically coherent classes) are notoriously elusive in such studies (Moreton & Pater, 2012), which seems at odds with the hypothesis that phonological classes in real languages should be phonetically coherent. To investigate the role of distributional learning, researchers might perform studies that investigate whether classes that are both phonetically coherent and highly salient in the distribution of participants' native languages are generalised more robustly in AGL tasks than classes that are just distributionally salient or just phonetically coherent. In addition, it would be interesting to investigate whether distributional learning of phonological classes is a strategy available to infants (similar to word segmentation; e.g. Saffran et al., 1996), or if it is a higher level strategy that does not become available until after further phonological development. Finally, although I have presented this algorithm using phonological data, it has potential applications in other domains, such as the learning of morphosyntactic classes.[25]

Several current debates in phonology revolve around how great a role distributional learning plays in the acquisition and transmission of phonological structure. The algorithm presented in this paper provides some insight into what kinds of phonological information are salient in distributional data. It is my hope that this might subsequently inform further study of the extent to which human learners are able to integrate this information into their

[25] Thanks to the anonymous reviewer who suggested this application.

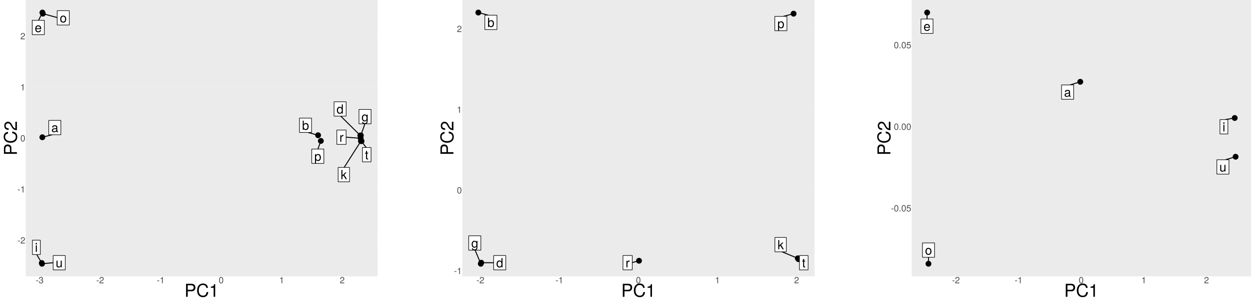
phonological grammars.

A Comparing algorithm parameters

The results of the algorithm vary depending on how it is parameterised. While there are too many possible parameter settings to compare every combination in depth here, this appendix will present several comparisons specifically relevant to the paper. The reader is encouraged to explore the consequences of different parameters on different data sets by downloading the supplemental code.[26]

A.1 Comparing counting methods

The algorithm presented here can count contexts using bigrams, trigrams, and a combination of bigrams and trigrams. Trigrams produce optimal results, while the combination of bigrams and trigrams does not differ significantly from using only trigram counts. Plots of the Parupa embeddings using bigram counts are shown in Figure 20.



(a) All segments

(b) Consonants

(c) Vowels

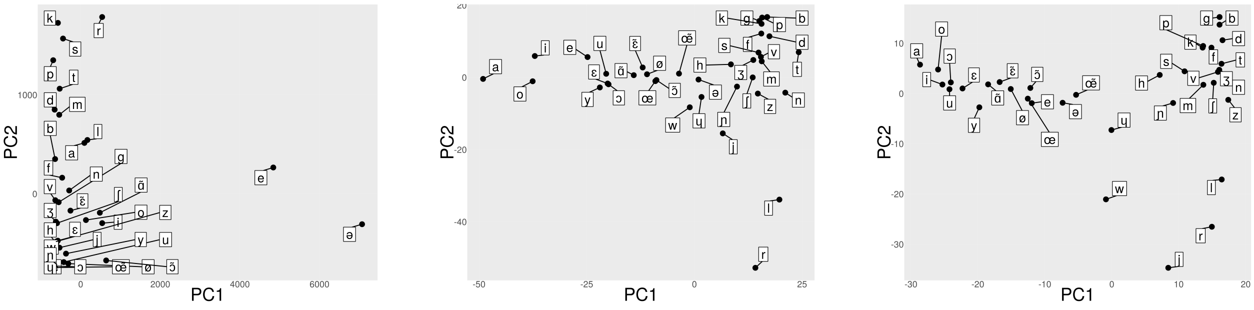
Figure 20: A PCA visualisation of the vector embeddings of Parupa generated using bigram counts and PPMI normalisation.

Note that in the bigram embeddings, the classes involved in the phonotactic restrictions on initial consonants ({p, b} and all other consonants) and consonant vowel co-occurrence ({p, t, k}, {b, d, g}, and {r}) are still apparent in the PCA. These restrictions all operate over bigram windows. The classes involved in vowel harmony (front and back vowels), which operates over a trigram window, are no longer well defined (see PC2 in Figure 20c). Having a sufficiently large window to detect long distance phonotactic restrictions is necessary for the remaining steps of the algorithm to succeed in extracting the correct classes.

A.2 Normalisation method

The effects of changing the normalisation method varied to some extent between corpora. Using raw counts produced poor results in all cases, while using PMI instead of PPMI made the algorithm unable to correctly separate consonants and vowels for French and Finnish. A

[26] `https://github.com/connormayer/distributional_learning`



(a) Raw counts **(b)** PMI **(c)** PPMI

Figure 21: A PCA visualisation of the vector embeddings of French generated using trigram counts and (a) no normalisation; (b) PMI; (c) PPMI.

PCA visualisation of the vector embedding of sounds in French using raw counts, PMI, and PPMI is shown in Figure 21. Note that PPMI shows the most distinct separation between consonants and vowels.

A.3 Restrictions on the initial partition

The results presented in this paper apply two restrictions on how the full set of sounds is partitioned: the partition must be into a maximum of two classes, and it must only be on the first principal component. This section details results from Parupa and French when these restrictions are removed.

The resulting classes for Parupa are shown in Table 8.

{i, e, u, o, a}	{t, k, d, g, r}	{a, b, p, t, k, b, d, g, r}
{i, u}	{p, b}	{i, e, u, o, a, r}
{e, o}	{b, d, g}	{a, r}
{i, e}	{p, t, k}	
{u, o}	{d, g}	
{a}	{t, k}	
	{p}	
	{b}	
	{r}	

Table 8: Classes learned from Parupa when restrictions on the initial partition are lifted.

All the classes recovered in Table 4 are recovered here except for the full class of consonants {p, t, k, b, d, g, r}. Instead, the initial partition of the segmental inventory is into three classes: {p, b}, {t, k, d, g, r}, and {i, e, u, o, a}. The algorithm successfully distinguishes consonants from vowels, but immediately partitions the consonants into two sets due to the distributional differences between {p, b} and the other consonants. There are also additional classes containing a mixture of consonants and vowels that are found because the algorithm is allowed to cluster on more than just the first PC: the class {a, b, p, t, k, b, d, g, r} are the sounds that do not participate in vowel harmony; {i, e, u, o, a, r} are the sounds that do not participate (or participate indirectly) in consonant-vowel co-occurrence restrictions; and {a, r} are the sounds that neither participate in vowel harmony nor consonant-vowel co-occurrence restrictions.

I will not present the full set of classes found for French, but I note that without the restriction forcing an initial partition into two classes, the recovered classes from the first partition are {p, t, k, b, d, g, m, n, ɲ, f, v, s, z, ʃ, ʒ, h, j, l, r}, {i, y, ø, ɛ, u, o, ɔ, a, ɛ̃, ɑ̃} and {e, œ, ə, œ̃, ɔ̃, w, ɥ}. This is the only case where the restriction on the initial partition is necessary to cleanly separate the vowels and consonants.

In all languages, allowing the algorithm to cluster on more than one PC of the full embedding produces additional classes containing both consonants and vowels, as in the third colum in Table 8 for Parupa. All other classes that are discovered from recursive clustering on the first PC are still recovered.

A.4 Effects of the variability parameter

This section will provide an example of how the retrieved classes vary as the variability scaling parameter V is changed. I will use English as an example. Table 9 shows the classes that are recovered as the variability parameter is gradually lowered from 2 to 1. Note that all classes recovered with higher values of the parameter are also recovered at lower levels: that is, for a given value of $V = v$ in Table 9, the classes in all rows where $V >= v$ will be retrieved.

V	Consonant classes	Vowel classes
3	{p, t, k, b, d, g, m, n, ŋ, f, v, θ, ð, s, z, ʃ, ʒ, h, tʃ, dʒ, w, j, l, ɹ}	{i, ɪ, eɪ, ɛ, æ, u, ʊ, ʌ, o, ɔ, ɑ, ɝ, aɪ, aʊ, ɔɪ}
2	{t, k, d, g, m, n, ŋ, v, θ, ð, s, z, ʃ, ʒ, tʃ, dʒ} {p, b, f, h, j, l} {w, ɹ}	
1.5	{t, d, m, n, ŋ, v, θ, ð, s, z, ʃ, ʒ, tʃ, dʒ} {k, g} {p, b, f, h, l} {j}	
1.4	{p, b, f} {h, l} {p, f} {b}	{i, ɪ, eɪ, ɛ, æ, u, ʌ, o, ɑ, aɪ, aʊ, ɔɪ} {ʊ, ɔ, ɝ} {ʊ, ɔ} {ɝ}
1.1	{p, b, f, j}	{i, ɪ, eɪ, ɛ, æ, u, ʌ, ɑ, aɪ, aʊ, ɔɪ} {o} {ɪ, ɛ, æ, u, ʌ, ɑ} {i, eɪ, aɪ, aʊ, ɔɪ} {ɪ, ɛ, æ, ʌ, ɑ} {u} {ɪ, ɛ, æ, ʌ} {ɑ} {i, aɪ, aʊ, ɔɪ} {eɪ} {i, aɪ} {aʊ} {ɔɪ} {eɪ, ɔɪ}
1		{i, ɛ, æ, u, ʊ, ʌ, ɔ, ɑ, ɝ, aɪ, aʊ, ɔɪ} {ɪ, eɪ} {i, ɛ, æ, ʊ, ʌ, ɔ, ɑ, aɪ, aʊ, ɔɪ} {ɛ, æ, ʊ, ɔ, ɑ} {i, ʌ, aɪ, aʊ, ɔɪ} {ɛ, æ, ʊ} {ɔ, ɑ} {i, ʌ} {aɪ, ɔɪ}

Table 9: Classes learned from English using various variability scalar values.

References

Aggarwal, C. C., & Reddy, C. K. (2013). *Data clustering: Algorithms and applications.* CRC Press.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*, 716-723.

Albright, A. (2009). Feature-based generalization as a source of gradient acceptability. *Phonology*, *26*, 9-41.

Alderete, J., & Bradshaw, M. (2013). Samoan root phonotactics: Digging deeper into the data. *Linguistic Discovery*, *11*.

Andrews, M., Vigliocco, G., & Vinson, D. (2009). Integrating experiential and distributional data to learn semantic representations. *Psychological Review*, *116*, 463-498.

Anttila, A. (2008). Gradient phonotactics and the Complexity Hypothesis. *Natural Language and Linguistic Theory*, *26*, 695-729.

Archangeli, D., Mielke, J., & Pulleyblank, D. (2011). Greater than noise: Frequency effects in Bantu height harmony. In B. Botma & R. Noske (Eds.), *Phonological explorations: Empirical, theoretical and diachronic issues* (p. 191-222). Berlin: Mouton de Gruyter.

Archangeli, D., & Pulleyblank, D. (2015). Phonology without universal grammar. *Frontiers in Psychology*, *6*, 1229.

Archangeli, D., & Pulleyblank, D. (2018). Phonology as an emergent system. In S. Hannahs & A. R. Bosch (Eds.), *The Routledge Handbook of Phonological Theory* (p. 476-503). London: Routledge.

Baayen, R., Piepenbrock, R., & Gulikers, L. (1995). *CELEX2 LDC96L14.* Web Download. Philadelphia: Linguistic Data Consortium.

Bach, E., & Harms, R. T. (1972). How do languages get crazy rules? In R. P. Stockwell & R. K. Macaulay (Eds.), *Linguistic change and generative theory* (p. 1-21). Bloomington and London: Indiana University Press.

Beguš, G. (2018a). *Unnatural phonology: A synchrony-diachrony interface approach* (Unpublished doctoral dissertation). Harvard University.

Beguš, G. (2018b). A formal model of phonological typology. In W. G. Bennett, L. Hracs, & D. Storoshenko (Eds.), *Proceedings of the 35th west coast conference on formal linguistics* (p. 104-113). Somerville, MA: Cascadilla Proceedings.

Beguš, G., & Nazarov, A. (to appear). Gradient trends against phonetic naturalness: The case of Tarma Quechua. In *Proceedings of the 48th annual meeting of the north east linguistic society (nels 48).*

Blevins, J. (2004). *Evolutionary phonology: The emergence of sound patterns.* Cambridge: Cambridge University Press.

Bruni, E., Tran, N. K., & Baroni, M. (2014). Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, *49*, 1-47.

Bybee, J. (1995). Regular morphology and the lexicon. *Language and Cognitive Processes*, *10*.

Calderone, B. (2009). Learning phonological categories by independent component analysis. *Journal of Quantitative Linguistics*, *16*.

Cangelosi, R., & Goriely, A. (2007). Component retention in principal component analysis with application to cDNA microarray data. *Biology Direct*, *2*.

Chomsky, N. (1957). *Syntactic structures.* The Hague: Mounton.

Chomsky, N. (1965). *Aspects of the theory of syntax.* Cambridge, MA: MIT Press.

Chomsky, N. (1988). *Language and the problem of knowledge.* Cambridge, MA: MIT Press.

Chomsky, N., & Halle, M. (1968). *The sound pattern of English.* New York: Harper & Row.

Church, K. W., & Hanks, P. (1990). Word association, norms, mutual information, and lexicography. *Computational Linguistics*, *16*, 22-29.

Coetzee, A. W., & Pater, J. (2008). Weighted constraints and gradient restrictions on place co-occurrence in Muna and Arabic. *LI*, *26*, 289-337.

Dagan, I., Marcus, S., & Markovitch, S. (1993). Contextual word similarity and estimation from sparse data. In *ACL-93* (p. 164-171). Columbus, Ohio.

Dresher, B. E. (2014). The arch not the stones: Universal feature theory without universal features. *Nordlyd*, *41*, 165-181.

Ellison, T. M. (1991). The iterative learning of phonological constraints. *Computational Linguistics*, *20*.

Ellison, T. M. (1994). *The machine learning of phonological structure* (Unpublished doctoral dissertation). University of Western Australia.

Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., & Plunkett, K. (1996). *Rethinking innateness: A connectionist perspective on development.* Cambridge, MA: Bradford Books/MIT Press.

Fano, R. M. (1961). *Transmission of information: A statistical theory of communications.* MIT Press.

Feldman, N. H., Griffiths, T. L., Goldwater, S., & Morgan, J. L. (2013). A role for the developing lexicon in phonetic category acquisition. *Psychological Review*, *120*(4), 751-778.

Gallagher, G. (2019). Phonotactic knowledge and phonetically unnatural classes: the plain uvular in Cochabamba Quechua. *Phonology*, *36*, 37-60.

Goldsmith, J. (2010). Segmentation and morphology. In A. Clark, C. Fox, & S. Lappin (Eds.), *The handbook of computational linguistics and natural language processing* (p. 364-393). Wiley Blackwell.

Goldsmith, J., & Riggle, J. (2012). Information theoretic approaches to phonological structure: the case of Finnish vowel harmony. *Natural Language and Linguistic Theory*, *30*, 859–896.

Goldsmith, J., & Xanthos, A. (2008). Three models for learning phonological categories. Technical report 2008-8. Chicago: Department of Computer Science, University of Chicago.

Goldsmith, J., & Xanthos, A. (2009). Learning phonological categories. *Lg*, *85*, 4-38.

Goldwater, S., Griffiths, T. L., & Johnson, M. (2009). A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, *112*, 21-54.

Golub, G. H., & Reinsch, C. (1970). Singular value decomposition and least squares solutions. *Numerische Mathematik*, *14*, 403-420.

Harris, Z. (1946). From morpheme to utterance. *Language*, *22*, 161-183.

Hayes, B. (2004). Phonological acquisition in Optimality Theory: The early stages. In R. Kager, J. Pater, & W. Zonneveld (Eds.), *Fixing priorities: Constraints in phonological acquisition* (p. 158-203). Cambridge: Cambridge University Press.

Hayes, B., & Wilson, C. (2008). A maximum entropy model of phonotactics and phonotactic learning. *LI*, *39*(3), 379 - 440.

Hayes, B., Zuraw, K., Siptar, P., & Londe, Z. (2009). Natural and unnatural constraints in hungarian vowel harmony. *Language*, *85*, 822-863.

Heinz, J., Rawal, C., & Tanner, H. G. (2011). Tier-based strictly local constraints in phonology. In *Proceedings of the 49th annual meeting of the association for computational linguistics* (p. 58-64).

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*.

Iverson, G. K., & Salmons, J. C. (2011). Final devoicing and final laryngeal neutralization. In M. van Oostendorp, C. J. Ewen, E. Hume, & K. Rice (Eds.), *The blackwell companion to phonology* (p. 1622-1643). Malden, MA: Blackwell.

Jarosz, G. (2006). *Rich lexicons and restrictive grammars: Maximum likelihood learning in Optimality Theory* (Unpublished doctoral dissertation). John Hopkins University.

Jurafsky, D., & Martin, J. (2008). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech processing.* Upper Saddle River, NJ: Prentice-Hall.

Kaiser, H. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, *23*, 187-200.

Kaisse, E. M. (2002). *Laterals are [-continuant].* MS, University of Washington.

Kiparsky, P. (1973). Phonological representations. In O. Fujimura (Ed.), *Three dimensions of linguistic theory* (p. 1-136). Tokyo: TEC Co.

Kiparsky, P. (2006). Amphichronic linguistics vs. evolutionary phonology. *Theoretical Linguistics*, *32*.

Kiparsky, P. (2008). Universals constrain change: change results in typological generalizations. In J. Good (Ed.), *Language universals and language change* (p. 25-53). Oxford: Oxford University Press.

Kohonen, T. (2002). *Self-organizing maps.* Heidelberg: Springer-Verlag.

Labov, W., Ash, S., & Boberg, C. (2006). *The atlas of North American English.* Berlin: Mouton de Gruyter.

Lahiri, A., & Marslen-Wilson, W. (1991). The mental representation of lexical form: a phonological approach to the recognition lexicon. *Cognition*, *38*(3), 245-294.

Lin, Y. (2005). *Learning features and segments from waveforms: A statistical model of early phonological acquisiton* (Unpublished doctoral dissertation). UCLA.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematics, statistics, and probability* (Vol. 1, p. 281-296).

MacWhinney, B., & O'Grady, W. (Eds.). (2015). *The handbook of language emergence.* Chichester: John Wiley & Sons.

Manning, C., & Schütze, H. (1999). *Foundations of statistical natural language processing.* Cambridge, MA: MIT Press.

Mayer, C., & Daland, R. (in press). A method for projecting features from observed sets of phonological classes. *Linguistic Inquiry*.

Mayer, C., & Nelson, M. (submitted). Phonotactic learning with neural models.

Mayer, T. (2010). Toward a totally unsupervised, language-independent method for the syllabification of written texts. In *Proceedings of the 11th meeting of the acl-sigmorphon, acl 2010* (p. 63-71). Uppsala, Sweden: Association for Computational Linguistics.

McCarthy, J. J. (1986). OCP effects: Gemination and antigemination. *LI*, *17*, 207-263.

Mielke, J. (2008). *The emergence of distinctive features.* Oxford: Oxford University Press.

Mielke, J. (2012). A phonetically-based metric of sound similarity. *Lingua*, *1222*, 145-163.

Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., & Khundanpur, S. (2010). Recurrent neural network based language model. In *Proceedings of Interspeech* (p. 1045-1048).

Milner, G. (1993). *Samoan dictionary: Samoan-English, English-Samoan.* Polynesian Press.

Minka, T. P. (2000). *Automatic choice of dimensionality for pca.* Technical Report 514, MIT Media Lab.

Mirea, N., & Bicknell, K. (2019). Using lstms to assess the obligatoriness of phonological distinctive features for phonotactic learning. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (p. 1595-1605).

Moreton, E. (2008). Analytic bias and phonological typology. *Phonology*, *25*, 83-128.

Moreton, E., & Pater, J. (2012). Structure and substance in artificial-phonology learning. part i: Structure, part ii: Substance. *Language and Linguistics Compass*, *6*, 686-701 and 702-718.

Müller, E., Günnemann, S., Assent, I., & Seidl, T. (2009). Evaluating clustering in subspace projections of high dimensional data. *Proceedings of VLDB '09*.

Nazarov, A. (2014). A radically emergentist approach to phonological features: Implications for grammars. *Nordlyd*, *41*(1), 21-58.

Nazarov, A. (2016). *Extending hidden structure learning: Features, opacity, and exceptions* (Unpublished doctoral dissertation). University of Massachusetts Amherst.

Niwa, Y., & Nitta, Y. (1994). Co-occurence vectors from corpora vs. distance vectors from dictionaries. In *ACL-94* (p. 304-309).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pelleg, D., & Moore, A. (2000). X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the 17th international conference on machine learning* (p. 727-734). Morgan Kaufmann.

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

Powers, D. M. W. (1997). Unsupervised learning of linguistic structure: An empirical evaluation. *International Journal of Corpus Linguistics*, *2*, 91-132.

Prince, A. S., & Tesar, B. B. (2004). Learning phonotactic distributions. In R. Kager, J. Pater, & W. Zonneveld (Eds.), *Fixing priorities: Constraints in phonological acquisition* (p. 245-291). Cambridge: Cambridge University Press.

Pullum, G. K., & Scholz, B. C. (2002). Empirical assessment of stimulus poverty arguments. *The Linguistic Review*, *19*, 9-50.

Redington, M., Chater, N., & Finch, S. (1998). Distributional information: A powerful cue for acquiring syntactic categories. *Cognitive Science*, *22*, 425-469.

Saffran, J. R., Newport, E. L., & Aslin, R. N. (1996). Word segmentation: The role of distributional cues. *Journal of Memory and Language*, *35*, 606-621.

Scheer, T. (2015). How diachronic is synchronic grammar? crazy rules, regularity, and naturalness. In P. Honeybone & J. C. Salmons (Eds.), *The handbook of historical phonology* (p. 313-336). Oxford: OUP.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461-464.

Silfverberg, M., Mao, L. J., & Hulden, M. (2018). Sound analogies with phoneme embeddings. In *Proceedings for the society for computation in linguistics (scil) 2018* (p. 136-144).

Sundermeyer, M., Schlüter, R., & Ney, H. (2012). LSTM neural networks for language modeling. In *Proceedings of the thirteenth annual conference of the international speech communication association.*

Tomasello, M. (2003). *Constructing a language: A usage-based theory of language acquisition.* Harvard University Press.

Trubetzkoy, N. S. (1939). *Principles of phonology* (C. A. Baltaxe, Trans.). Los Angeles: University of California Press.

Vennemann, T. (1974). Sanskrit *ruki* and the concept of a natural class. *Linguistics*, *130*, 91-97.

Wang, H., & Song, M. (2011). Ckmeans.1d.dp: Optimal *k*-means clustering in one dimension by dynamic programming. *The R Journal*, *3*, 29-33.

Wetzels, W. L., & Mascaró, J. (2001). The typology of voicing and devoicing. *Language*, *77*, 207-244.

Wonnacott, E., Newport, E., & Tanenhaus, M. (2008). Acquiring and processing verb argument structure. *Congitive Psychology*, *56*, 165-209.