# An algorithm for learning phonological classes from distributional similarity


Connor Mayer

*University of California, Los Angeles*



AMP 2018



connormayer@ucla.edu | www.connormayer.com


## 1. Introduction

**Where do features come from?**

1. Are features *innate?*
   - No learning required
2. Are features *universal?* (e.g. Chomsky & Halle, 1968)
   - Learned from general **phonetic** properties
   - Classes are phonetically coherent
3. Are features *learned* and *language-specific?* (e.g. Mielke, 2008; Archangeli & Pulleyblank, 2015)
   - Learned from **phonetics *and* data**
   - Classes need not be phonetically coherent

**Current study: can we learn features/classes *without phonetics*?**

- i.e. from *distributional information*:
  - Where sounds do and do not occur in the data

I present an algorithm that can learn complex class structures without recourse to phonetic information.

## 2. The task at hand

Our learner needs to find *distributionally salient* classes from a phonological corpus.

- We don't know how many!
- Classes may be *nested* or *overlapping*
- Learner must be robust to *distributional noise*

Most importantly, it needs to find the "right" classes…

- C vs. V would be a good start
- But how to evaluate the other classes we find?

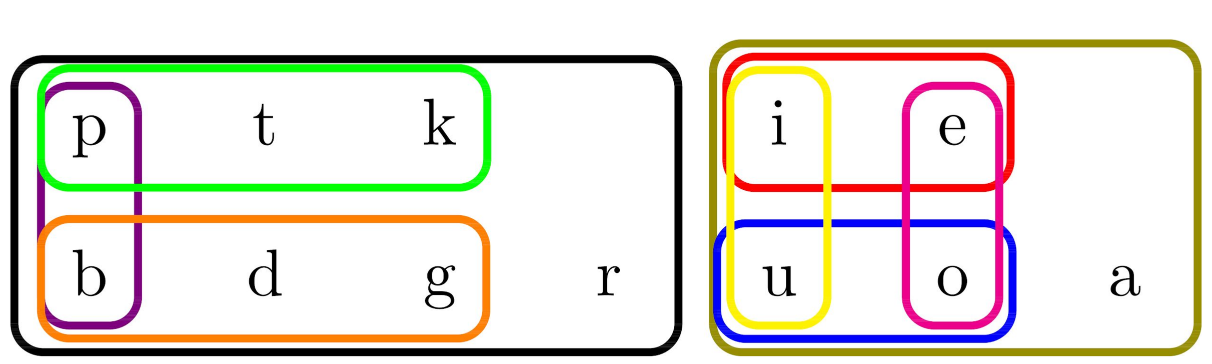
## 3. Parupa: a toy language

**Solution:** Test on a toy language with known classes!

**Parupa** is a CV language with several distributional constraints:

- Words **harmonize for backness**; /a/ is transparent
- Words **must begin** with **/p/ or /b/**
- **/p/, /t/, /k/** are followed by **high vowels** or /a/
- **/b/, /d/, /g/** are followed by **mid vowels** or /a/

*berari*
*pupabopa*
*boka*
*padoropa*
*pakubatuda*
*bopu*
*piretiba*
*pabarubo*
*barika*
*…*




Inventory and phonological classes of Parupa

## 4. The nuts and bolts of the algorithm

**<u>Step 1</u>:** Build a ***vector representation*** of each sound in the corpus

1. Count all the ***trigram contexts*** in which each sound occurs
   - These are the dimensions of the vector
2. Weight contexts using ***positive pointwise mutual information (PPMI)***
   - Emphasizes contexts where a sound occurs more frequently than chance

**Why?**

1. Representing sounds as vectors lets us find classes numerically!
   - Distributionally similar sounds should be close in space
2. Using PPMI lets us focus on sound/context pairs that are *informative*

**<u>Step 2</u>:** Perform ***Principal Component Analysis (PCA)*** on the embeddings

1. PCA projects points from a high dimensional space to a lower dimensional space while minimizing loss of variance
2. New dimensions (*principal components*) are ordered by variance captured

**Why?**

1. Highlights robust sources of variance and reduces noise
2. Different PCs reveal multiple partitions of the same set of sounds

**<u>Step 3</u>:** Perform ***clustering*** on *individual* principal components to find classes

1. Do ***1-dimensional k-means clustering*** on all PCs with "high variance"
2. Find optimal number of classes using *Bayesian Information Criterion*
3. Do clustering recursively on all discovered classes

**Why?**

1. Clustering on individual PCs allows us to find *overlapping classes*
2. Recursive clustering allows us to find *nested classes*

## 5. Results on Parupa

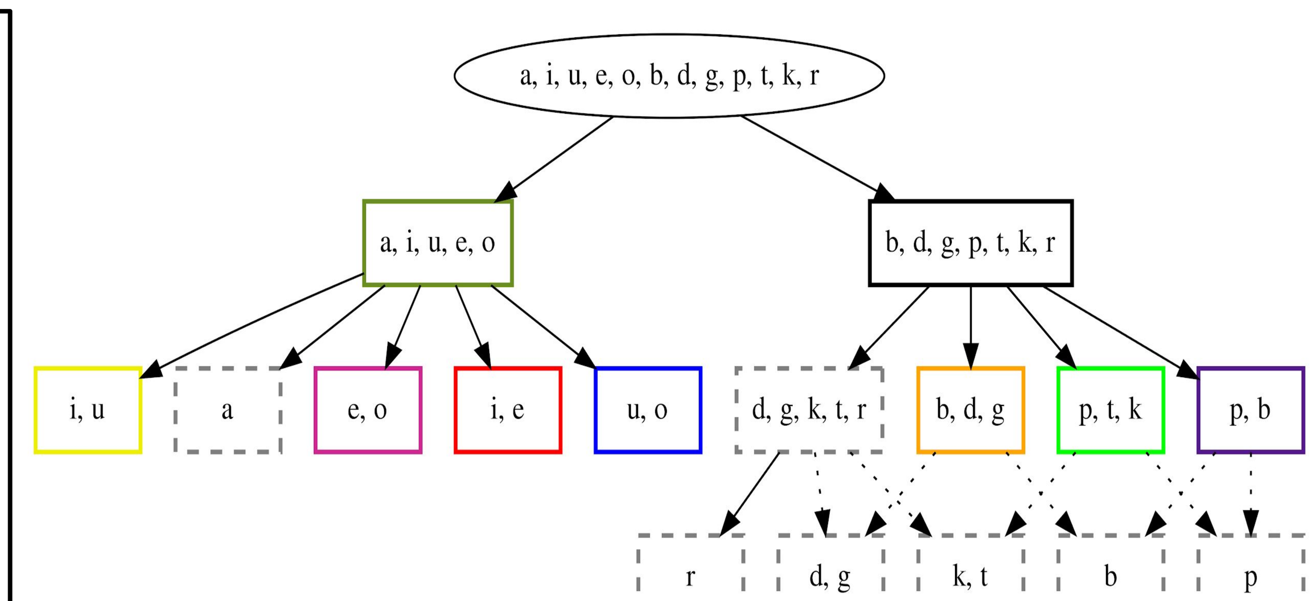
Run on corpus of ~18k word types

Finds **all expected classes**!

- Other classes found are *derivable* from expected classes (complements, intersections)

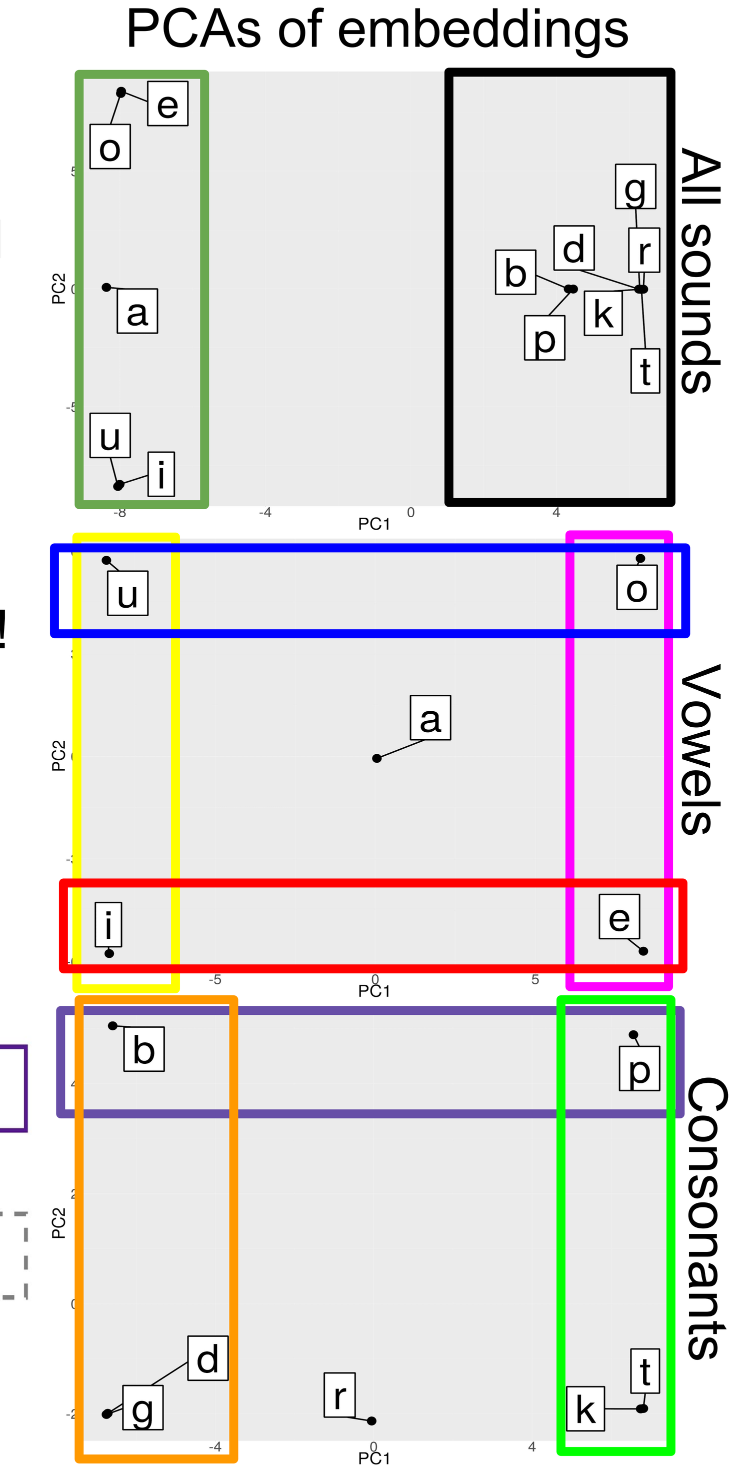
Performs well **even as noise is added**

- Allow some percentage of words to violate phonotactic constraints (except CV)
- Most expected classes found in up to 90% noise!




Discovered classes displayed hierarchically. Colors match those in the inventory diagram in section 3. Dashed boxes indicate "unexpected" classes.

PCAs of embeddings




## 6. Results on natural languages

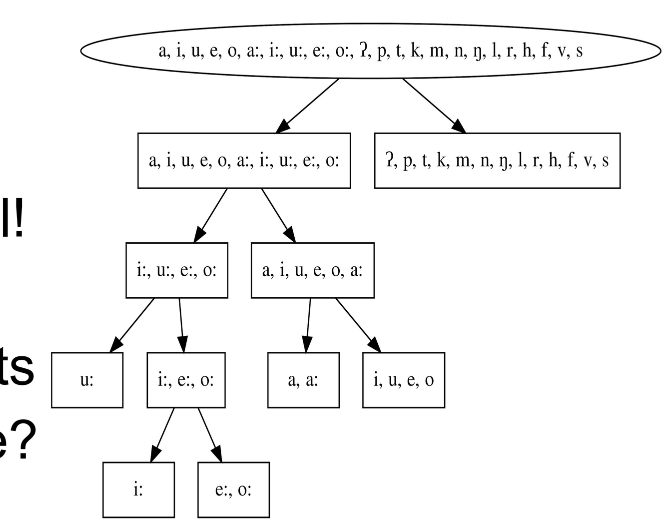
Successfully distinguishes C vs. V in four languages, and finds classes suggesting...

- **English**
  - tense vs. lax vowels
  - coronals & nasals vs. other consonants
- **French**
  - liquids vs. glides vs. other consonants
  - nasal & marked rounding vs. other vowels
- **Samoan**
  - long vs. short vowels
- **Finnish**
  - front vs. back vs. neutral vowels

## 7. A short example: Samoan vowels

Vowel are split into long vs. short, except for /a:/

- VV sequences are common
- V:V, VV:, V:V: are not
  - In 76% of these, V: is /a:/
- /a:/ patterns like a short vowel!




No structure found in consonants

- Due to strict (C)(V)V structure?
- Trigram window too small?

## 8. Discussion/Future Directions

This algorithm learns nested/overlapping classes from a phonological corpus with no phonetic information

**Why is this interesting?**

- Provides insight into which classes are distributionally salient in a language
- Improves on performance of past attempts (e.g. Goldsmith & Xanthos, 2009)
- May be combined with other sources (e.g. phonetics) for more realistic learnability models
- Can be used as input for testing productivity of distributionally salient classes using artificial grammar learning tasks

Next step: Moving beyond trigram counts…

**See paper on my website for *much* more detail!**

## Selected References


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

Chomsky, N., & Halle, M. (1968). *The Sound Pattern of English*. New York Harper & Row.

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

Mielke, J. (2008). *The Emergence of Distinctive Features*. Oxford: Oxford Press.



Acknowledgements: I would like to thank Bruce Hayes, Kie Zuraw, Yizhou Sun, Tim Hunter, Pat Keating, and Robert Daland for their guidance and support throughout this project. Thanks also to the attendees of the UCLA phonology seminar for their valuable questions and insights.