# An algorithm to assign features to a set of natural classes

Mayer, Connor Joseph
connor.joseph.mayer@gmail.com

Daland, Robert
r.daland@gmail.com

December 5, 2017

**Abstract**

This squib describes a dynamic programming algorithm which assigns features to a set of natural classes. The input consists of a set of classes, each containing one or more segments; in other words, a subset of the powerset of a segmental alphabet $\Sigma$. If a class can be generated as the union of existing features ( = intersection of already-processed classes), those features are propagated to every segment in the class. Otherwise, a new feature/value is assigned. The algorithm comes in 4 flavors, which differ with respect to complementation and how negative values are assigned. We show that these variants yield *privative specification*, *contrastive underspecification*, *contrastive specification*, and *full specification*, respectively. The main text sets out necessary background, and illustrates each variant of the algorithm. The Appendix formally proves that each algorithm is sound.

## 1 Introduction

merge what Connor wrote

## 2 Definitions and notation

Let $\Sigma$ denote an alphabet of segments. We will use the term *class* to mean a subset of $\Sigma$.

### 2.1 Definition and example of natural class system

A *natural class system* $\mathcal{C}$ is a set of classes over $\Sigma$, $\mathcal{C} = \{C_i\}_{i=1}^{N}$, which includes $\Sigma$ itself, and the empty set (i.e. $\varnothing, \Sigma \in \mathcal{C}$).

Readers who are familiar with the notion of *lattice* will note that every natural class system forms a lattice under the subset relation. To illustrate, a vowel harmony lattice is shown below (the empty set is suppressed):

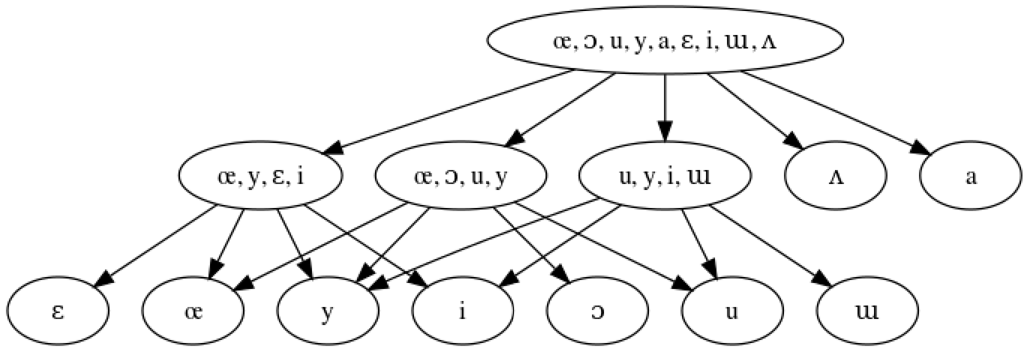


Figure 1: Vowel harmony lattice

## 2.2 Definition and example of a feature system

A *feature system* is a tuple $(\mathcal{F}, \Sigma, \mathcal{V})$ where

- $\Sigma$ is a segmental alphabet,
- $\mathcal{V}$ is a set of values, and
- $\mathcal{F} = \{f_j : \Sigma \to \mathcal{V}\}_{j=1}^{M}$ is a set of feature functions mapping segments to feature values

We say that a feature system has *privative specification* if $\mathcal{V} = \{+, 0\}$, *full specification* if $\mathcal{V} = \{+, -\}$, and *contrastive specification* if $\mathcal{V} = \{+, -, 0\}$. We do not consider other value sets here.

A (fully specified) feature system for the vowel harmony lattice shown in Fig. 1 is shown below:

| $\sigma$ | front | back | low | high | round |
|---|---|---|---|---|---|
| i | + | − | − | + | − |
| y | + | − | − | + | + |
| ɯ | − | + | − | + | − |
| u | − | + | − | + | + |
| ɛ | + | − | − | − | − |
| œ | + | − | − | − | + |
| ʌ | − | + | − | − | − |
| ɔ | − | + | − | − | + |
| a | − | + | + | − | − |

Table 1: Example of a (fully specified) featurization.

### 2.3 Featural descriptors

Let $(\mathcal{F}, \Sigma, \mathcal{V})$ be a feature system. The following definitions will prove useful:

- We will refer to the set of feature functions $\mathcal{F} = \{f_j\}_{j=1}^{M}$ as a *featurization* (of $\Sigma$).
- A *featural descriptor* $\mathbf{e}$ is a subset of $(V \backslash \{0\}) \times \mathcal{F}$, in other words a set of feature/value pairs where the value cannot be 0
  - featural descriptors can be written in the form $[\alpha_k f_k]_{k \in K}$ for some index set $K$
  - an example is [+front, −low]
- The natural class described by a featural descriptor $\mathbf{e}$, written $\langle \mathbf{e} \rangle$, consists of every segment which has *at least* the feature/value pairs in $\mathbf{e}$
  - $\mathbf{e} = [\alpha_k f_k]_{k \in K} \leftrightarrow \langle \mathbf{e} \rangle = \{x \in \Sigma \,|\, \forall k \in K\, [f_k(x) = \alpha_k]\}$
  - for the feature system in Table 1, the natural class described by [+front, −low] is { i, y, ɛ, œ }
- Let $\mathcal{V}^{\mathcal{F}}$ denote the set of all licit featural descriptors over $(\mathcal{F}, \Sigma, \mathcal{V})$. Let $\langle \mathcal{V}^{\mathcal{F}} \rangle$ indicate the set of natural classes described by this set. We say that the feature system $(\mathcal{F}, \Sigma, \mathcal{V})$ generates $\langle \mathcal{V}^{\mathcal{F}} \rangle$.

Note that while every featural descriptor in $\mathcal{V}^{\mathcal{F}}$ picks out a class in $\langle \mathcal{V}^{\mathcal{F}} \rangle$, the two are not in 1-1 correspondence. This is because the same class can often be described by multiple featural descriptors. For example, under the the vowel feature system shown in Table 1, $[+front, -front]$ and $[+high, +low]$ both pick out the empty set.

### 2.4 Properties of feature systems

Let $(\mathcal{F}, \Sigma, \mathcal{V})$ be a feature system with featurization $\mathcal{F} = \{f_j\}_{j=1}^{M}$.

- The *feature vector* of a segment $x$ is the tuple $F(x) = (f_j(x))_{j=1}^{k}$.
- Two segments $x, y$ are *featurally distinct* if and only if $F(x) \neq F(y)$; in other words, if they do not match on at least feature.
- The feature system is *well-formed* if every pair of segments in $\Sigma$ is featurally distinct.[1]

[1] It is always possible to make ill-formed systems become well-formed. For example, suppose that [ptk] are not given distinct place features. One way to make the system well-formed is to add place features. Another way is to replace instances of [ptk] with a meta-symbol [T] in $\Sigma$, yielding a new segmental alphabet $\Sigma' = \Sigma \setminus \{p, t, k\} \cup \{T\}$.

- Let $\mathcal{C} = \{C_i\}_{i=1}^N$ be a set of natural classes. A featurization is *adequate* to distinguish $\mathcal{C}$ if and only if $\mathcal{C} \subset \langle \mathcal{V}^{\mathcal{F}} \rangle$, in other words if it provides a way to pick out every class in $\mathcal{C}$.
- A feature $f_j$ is *redundant* if $\mathcal{F}' = \mathcal{F} \setminus \{f_j\}$ is well-formed.
- A featurization is *efficient* if it contains no redundant features.
- A featurization is *minimal* (for the value set $\mathcal{V}$) if there is no well-formed featurization that contains a smaller number of feature functions than $\mathcal{F}$ does (mapping to the same value set).

It is straightforward to show that if $\mathcal{F}(\Sigma)$ is a well-formed featurization, then it generates a natural class system. Our goal in the remainder of this paper is the following:

- Suppose that the learner has evidence (based on phonetic similarity, and/or phonological patterning) for a set of natural classes $\mathcal{C} = \{C_i\}_{i=1}^N$
- Describe an algorithm which returns a well-formed feature system that is *adequate* for $\mathcal{C}$, *efficient*, and ideally *minimal*.

# 3 A dynamic programming algorithm for computing intersectional closure

Let $\mathcal{C} = \{C_i\}_{i=1}^n$ be a natural class system. The *intersective closure* of $\mathcal{C}$, denoted $\mathcal{C}_\cap$, is the natural class system consisting of every class that can be generated by the intersection of finitely many classes in $\mathcal{C}$ (e.g. every class in $\mathcal{C}$, as well as any class that can be generated by the intersection of two or more classes in $\mathcal{C}$). Formally,

# 4 Privative specification

achieved by assigning a new feature [+f] only, to every segment in $X$

# 5 Contrastive underspecification

achieved by assigning a new feature [+f] to every segment in $X$, and if $Y \setminus X$ (the complement of $X$ with respect to $Y$) is in the input, then [-f] is assigned to every segment in $Y \setminus X$

# 6 Contrastive specification

achieved by assigning a new feature [+f] to every segment in $X$, and [-f] to every segment in $Y \setminus X$ (even if $Y \setminus X$ was not in the input)

# 7 Full specification

achieved by assigning a new feature [+f] to every segment in $X$, and [-f] to every segment in $\Sigma \setminus X$

# A Formal proof of the algorithm

## A.1 Privative underspecification

## A.2 Contrastive underspecification

## A.3 Contrastive specification

## A.4 Full specification