

The Computational Complexity of Distinctive Feature Minimization in Phonology

Hubie Chen

Computer Science and Information Systems
Birkbeck, University of London
hubie@dcs.bbk.ac.uk

Mans Hulden

Department of Linguistics
University of Colorado
mans.hulden@colorado.edu

Abstract

We analyze the complexity of the problem of determining whether a set of phonemes forms a natural class and, if so, that of finding the minimal feature specification for the class. A standard assumption in phonology is that finding a minimal feature specification is an automatic part of acquisition and generalization. We find that the natural class decision problem is tractable (i.e. is in \mathcal{P}), while the minimization problem is not; the decision version of the problem which determines whether a natural class can be defined with k features or less is \mathcal{NP} -complete. We also show that, empirically, a greedy algorithm for finding minimal feature specifications will sometimes fail, and thus cannot be assumed to be the basis for human performance in solving the problem.

1 Introduction

The **distinctive feature** is held by many phonologists, independently of theoretical orientation, to be the fundamental unit of analysis of sound patterns in language. The underlying working assumption of most phonological approaches is that a single sound or a set of sounds is expressed through a combination of positive or negative features and that these features are in some sense universal across languages (Mielke, 2008a). The exact makeup of the feature set employed has varied over time, ranging from the limited, more acoustic-oriented features of Jakobson et al. (1951), to the richer model presented in Chomsky and Halle (1968), to more complex hierarchically organized features in Clements (1985). The concept of **natural class** is intimately tied to such feature systems and is taken to be any set of segments that share some number of distinctive features. Furthermore, phonological alternations that do not target natural classes are hypothesized not to occur.

Another assumption that is found in phonological literature—often less explicit—is that whenever a phonological process targets a group of sounds, that group is to be expressed nonredundantly by the *minimum number of features* required to do so. In general, one can find a multitude of ways in which a set of phonemes can be specified using positive or negative features. For example, using the feature system shown in Table 1, the set $\{m,n\}$ has the obvious minimal description $[+nas]$, since m and n are the only nasals. But that set could also be specified non-minimally as:

$$\begin{bmatrix} +cons \\ -cnt \\ -hi \\ -bk \\ +voi \\ +son \end{bmatrix} \quad (1)$$

The potential complexity of this problem, finding a minimal specification, is not addressed in the phonological literature. Yet, finding such a minimal specification is not trivial. Using more realistic feature systems such as the ones given in Hayes (2011), there are, for example, 208 distinct solutions to specifying the set $\{e,i\}$ in English, only four of which have the minimum length of 3 features. While such modern feature systems work with 25–30 features, one can actually assume more features are needed to cover more cross-linguistically exotic contrasts such as those produced by including click consonants (Miller, 2011). The P-base 3 resource (Mielke, 2008b) lists 398 features over 8 feature systems, covering 629 languages.

The general notion of **feature economy** has a long tradition in phonology (Jakobson, 1942; Martinet, 1955; Clements, 2003), mostly targeting entire feature systems in a language, i.e. advocating

that on the grammar level, individual languages make maximal use of the feature inventory available (Fant, 1966). Equally prominent is the assumption that any specification of phonological alternations be made with the minimal number of features necessary: “one should use the minimum number of features required to specify all and only the sounds in the class.” (Zsiga, 2012, p.282). Hayes (2011), among others, argues, following Ockham’s Razor, that this is how generalization can take place and that phonological hypotheses are made in precisely this way—witnessing alternations that target a set of sounds, learners find the minimal feature specification that is consistent with the alternating sounds, generalizing from there to other sounds that may enter the language. Halle (1962) also proposes a mechanism of “feature counting” as a methodology to rule out spurious generalizations one might propose—a process which implicitly includes the capability of feature minimization.

Similar arguments of feature minimization are used to perform an optimization of an entire phonological grammar. In Radical Underspecification, Archangeli (1984) refers to what is termed FEATURE MINIMIZATION PRINCIPLE: “A grammar is most highly valued when underlying representations include the minimal number of features necessary to make the different phonemes of the language” (Archangeli, 1984, p. 48).

Given such claims concerning acquisition and phonological analysis, it is of some interest to assess the actual computational complexity of feature minimization. This entails answering how difficult it is in the worst case to determine whether a set of segments represents a natural class, and also how to find the minimal feature specification.

2 Overview

We will assume a set of phonemes P —the phoneme inventory—and another set Q , our target set that we want to express through a combination of features, and a feature system \mathcal{F} such as the one shown in Table 1. The first problem we address is that of determining whether a set of phonemes Q forms a *natural class*, which we call the *feature description problem*. We show that this is decidable in polynomial time. Further, we will show that a minimization version of the problem, which we call the *feature minimization problem* is \mathcal{NP} -complete (Garey and Johnson, 1979; Sipser,

	cons	son	syl	voi	cnt	nas	lat	ant	cor	hi	bk	lo	rd
p	+	-	-	-	-	-	-	+	-	-	-	-	-
t	+	-	-	-	-	-	-	+	+	-	-	-	-
k	+	-	-	-	-	-	-	-	-	+	+	-	-
b	+	-	-	+	-	-	-	+	-	-	-	-	-
d	+	-	-	+	-	-	-	+	+	-	-	-	-
g	+	-	-	+	-	-	-	-	-	+	+	-	-
f	+	-	-	-	+	-	-	+	-	-	-	-	-
s	+	-	-	-	+	-	-	+	+	-	-	-	-
x	+	-	-	-	+	-	-	-	-	+	+	-	-
v	+	-	-	+	+	-	-	+	-	-	-	-	-
y	+	-	-	+	+	-	-	-	-	+	+	-	-
w	-	+	-	+	+	-	-	-	-	+	+	-	+
j	-	+	-	+	+	-	-	-	-	+	-	-	-
l	+	+	-	+	+	-	+	+	+	-	-	-	-
m	+	+	-	+	-	+	-	+	-	-	-	-	-
n	+	+	-	+	-	+	-	+	+	-	-	-	-
a	-	+	+	+	+	-	-	-	-	-	+	+	-
e	-	+	+	+	+	-	-	-	-	-	-	-	-
i	-	+	+	+	+	-	-	-	-	-	+	-	-
o	-	+	+	+	+	-	-	-	-	-	-	+	+
u	-	+	+	+	+	-	-	-	-	-	+	+	-
y	-	+	+	+	+	-	-	-	-	-	+	-	-

Table 1: An example typical feature system (truncated to 13 features).

2013). We will show this by reduction from the well known set covering problem (Karp, 1972).

Many phonologists espouse a combination of binary (equipollent) and privative (univalent) features (Trubetzkoy, 1969; Ewen and van der Hulst, 1985; Goldsmith, 1985). The rationale is that, for example, the feature $[\pm\text{labial}]$ has rarely, if ever, been found to play a role in a phonological system as $[-\text{labial}]$, i.e. phonological processes that target non-labials seem to be absent. Hence, many phonologists favor the use of a single possibility $[\text{LABIAL}]$ in a specification that includes the labials. For this reason, we analyze separately both the complexity of using only such privative features (positive features only), which we call the *positive feature description problem* (is Q a natural class if we only use positive features?) and the corresponding *positive feature minimization problem*.

3 Notation and terms

Throughout, when A is a set, we use $\wp(A)$ to denote the power set of A .

Relative to a set P of phonemes, we define a *feature system* to be a subset $\mathcal{F} \subseteq \wp(P)$. When $Q \subseteq P$, we define a *\mathcal{F} -description of Q* to be a sequence $G_1, \dots, G_m \subseteq P$ such that there exist pairwise distinct elements $F_1, \dots, F_m \in \mathcal{F}$ where:

- each G_i is equal to either F_i or $P \setminus F_i$, and
- $Q = G_1 \cap \cdots \cap G_m$.

We refer to m as the *size* of the description. We say that such a description is *positive* if each G_i is an element of \mathcal{F} .

We define the *feature description problem* as follows. An instance consists of a set P of phonemes, a feature system \mathcal{F} , and a non-empty subset $Q \subseteq P$; the problem is to decide whether or not there exists a \mathcal{F} -description of Q .

We define the *feature minimization problem* as follows. An instance consists of a tuple (P, \mathcal{F}, Q, k) where P , \mathcal{F} , and Q are an instance of the feature description problem, and $k \geq 1$ is a natural number. The problem is to decide whether or not there exists a \mathcal{F} -description of Q having size less than or equal to k .

We define the *positive feature description problem* to have the same instances as the *feature description problem*, but where the problem is to decide whether or not there exists a positive \mathcal{F} -description of Q . Analogously, we define the *positive feature minimization problem* to have the same instances as the *feature minimization problem*, but where the problem is to decide whether or not there exists a positive \mathcal{F} -description of Q obeying the size restriction.

3.1 The feature description problems

We will first show that the feature description problems are decidable in polynomial time.

Proposition 1. *The feature description problem and the positive feature description problem are each polynomial-time decidable.*

Proof. The algorithm for the feature description problem is as follows. Given an instance (P, \mathcal{F}, Q) , compute a set $\mathcal{C} \subseteq \wp(P)$ as follows. For each $F \in \mathcal{F}$, place F in \mathcal{C} if $Q \subseteq F$, and place $P \setminus F$ in \mathcal{C} if $Q \subseteq P \setminus F$. Then, check if $\bigcap \mathcal{C} = Q$; if so, accept, otherwise, reject. It suffices to argue that if there exists a \mathcal{F} -description of Q , then the elements of \mathcal{C} constitute such a description. If there exists a \mathcal{F} -description of Q , say, G_1, \dots, G_m , we have $Q \subseteq G_i$ for each i ; thus, $G_1, \dots, G_m \in \mathcal{C}$ and we have

$$Q \subseteq \bigcap \mathcal{C} \subseteq G_1 \cap \cdots \cap G_m = Q,$$

implying that the elements of \mathcal{C} provide a \mathcal{F} -description of Q .

For the positive feature description problem, the algorithm computes \mathcal{C}^+ to contain each $F \in \mathcal{F}$ such that $Q \subseteq F$, and accepts if and only if $\bigcap \mathcal{C}^+ = Q$. The proof of correctness is similar to that given for the general feature description problem. \square

3.2 The minimization problems

An instance (U, \mathcal{S}, k) of the set cover problem consists of a non-empty set U , a subset $\mathcal{S} \subseteq \wp(U)$, and a natural number $k \geq 1$. A *set cover* S_1, \dots, S_m is a sequence of sets from \mathcal{S} such that $S_1 \cup \cdots \cup S_m = U$; m is said to be its size. The problem is to decide whether or not there exists a set cover of size less than or equal to k (Karp, 1972). We prove that both the feature minimization problem and the positive feature minimization problem are \mathcal{NP} -complete, by reducing from set cover.

The reduction is the same for both of these problems, and is as follows. Given an instance (U, \mathcal{S}, k) of set cover, let x be a fresh element not in U .

Define

$$\begin{aligned} P &= U \cup \{x\} \\ \mathcal{F} &= \{(U \setminus S) \cup \{x\} \mid S \in \mathcal{S}\} \\ Q &= \{x\} \end{aligned}$$

The resulting instance is (P, \mathcal{F}, Q, k) .

The following establishes the correctness of this reduction.

Proposition 2. *The following are equivalent:*

1. There exists a size m set cover of U .
2. The set $Q = \{x\}$ has a positive \mathcal{F} -description of size m .
3. The set $Q = \{x\}$ has a \mathcal{F} -description of size m .

Proof. $1 \Rightarrow 2$: Let (S_i) be such a set cover, so that $S_1 \cup \cdots \cup S_m = U$. Then, by DeMorgan's laws, $(U \setminus S_1) \cap \cdots \cap (U \setminus S_m) = \emptyset$, and so $((U \setminus S_1) \cup \{x\}) \cap \cdots \cap ((U \setminus S_m) \cup \{x\}) = \{x\}$.

$2 \Rightarrow 3$ is immediate.

$3 \Rightarrow 1$: Suppose that G_1, \dots, G_m is such a description. Since $x \in G_i$ for each i , it follows that each G_i is an element of \mathcal{F} . Define S_1, \dots, S_m to be the sets such that $G_i = (U \setminus S_i) \cup \{x\}$. By reversing the argumentation in the $1 \Rightarrow 2$ case, we obtain that S_1, \dots, S_m is a set cover of U . \square

In the just-given reduction, the parameter k is not changed. We show that there in fact exists a reduction in the other direction, from set cover to each of the minimization problems, that likewise does not change the parameter k . This indicates a tight relationship between these minimization problems and the set cover problem.

Proposition 3. *Let (P, \mathcal{F}, Q, k) be an instance of the feature minimization problem; let \mathcal{C} be the set (from Proposition 1). The mapping which, upon being given this instance, returns (U, \mathcal{S}, k) where $U = P \setminus Q$ and $\mathcal{S} = \{G \setminus Q \mid G \in \mathcal{C}\}$, is a reduction to the set cover problem; here, the complement is with respect to U . Likewise, one obtains a reduction from the positive feature minimization problem to the set cover problem, by using \mathcal{C}^+ in place of \mathcal{C} .*

Proof. We argue this for the feature minimization problem as follows (the positive minimization case is analogous). It was seen in the proof of Proposition 1 that (P, \mathcal{F}, Q, k) is a ‘yes’ instance if and only if there exist $G_1, \dots, G_m \in \mathcal{C}$, with $m \leq k$, such that $G_1 \cap \dots \cap G_m = Q$. This equality holds if and only if $(G_1 \setminus Q) \cap \dots \cap (G_m \setminus Q) = \emptyset$, which holds if and only if $(G_1 \setminus Q) \cup \dots \cup (G_m \setminus Q) = U$. \square

Parametrized complexity The set cover problem, with the value k taken to be parameter, is known to be $W[2]$ -complete in parameterized complexity theory (Cygan et al., 2015, Theorem 13.21). As we have given polynomial-time reductions between set cover and each of the minimization problems that do not change the value k , we obtain the following.

Proposition 4. *Both the feature minimization problem and the positive feature minimization problem are $W[2]$ -complete, when viewed as parameterized problems with k taken to be the parameter.*

4 Empirical concerns

Many \mathcal{NP} -hard problems can in practice often be solved by either a greedy algorithm (Chvatal, 1979) or a branch-and-bound algorithm (Land and Doig, 1960) that recursively explores the search space, terminating search branches as soon as some defined limit is exceeded.

Greedy Search We have implemented a greedy search strategy that starts with no features, and

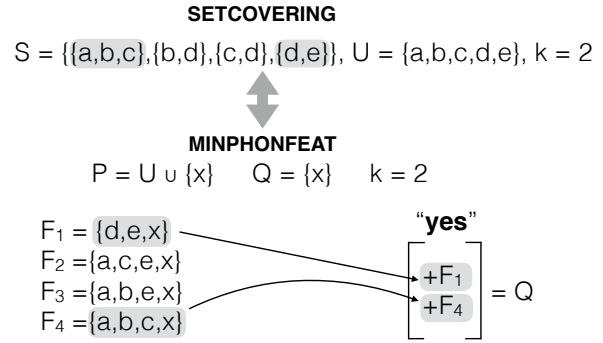


Figure 1: Illustration of the reduction from SETCOVERING: we can solve an instance of the set cover problem (SETCOVERING) by a call to a solver for the feature minimization problem (MINPHONFEAT). There are reductions both from and to the set cover problem.

then picks a single (\pm) feature from the known, possibly non-minimal description \mathcal{C} discussed in §3.1 in such a way as to rule out the majority of phonemes not in the desired set Q . Features are added to the description until only the set Q is described. For example, in describing the set $\{m\}$ using the example in Table 1, the greedy approach would first pick the feature $[+nas]$ since this cuts down the number of corresponding phonemes to just two (m and n), less than any other feature choice. This is essentially an analogue to the well-known greedy approximation algorithm for set cover (Chvatal, 1979).

This greedy algorithm, however, in many practical cases fails to find the correct minimal specification, and is therefore not a viable candidate for efficient feature minimization. A simple example is finding the featural specification for the set $\{\emptyset\}$ in English under the fairly standard featural system provided in Hayes (2011) and van Vugt and Hayes (2012): the algorithm recovers the features $[-tense, -back, -front, -coronal]$ while the minimal specification is $[-back, -front, -coronal]$.

Branch & Bound We also implemented a branch-and-bound algorithm with a recursive component that explores, exhaustively, all combinations of features in the full description \mathcal{C} , bounding the search whenever the current search tree contains more features than in the shortest solution found so far. The branch-and-bound algorithm is efficient in practice in finding minimal feature specifications, but still needs to explore a reasonably large search space (see Table 2).

Set	$\{h\}$	$\{i, e\}$	$\{r\}$	$\{j\}$	$\{m, n\}$	$\{p, t, k\}$
BF	32,768	262,144	131,072	524,288	2048	4,096
B&B	3,432	11,482	3,444	4,790	1,450	1,315

Table 2: The number of search nodes explored during a brute-force search (BF) for finding a minimal solution for various sets of phonemes, or a branch-and-bound search (B&B). The feature inventory is one used for English (van Vugt and Hayes, 2012; Hayes, 2011).

5 Discussion

Humans have been noted to outperform simple, low-polynomial time heuristic algorithms for some intractable problems such as the traveling salesman problem (MacGregor and Ormerod, 1996) using intuition and visual inspection of the problem structure. It is, however, unclear if such performance carries over to other problems of a different structure, such as MINPHONFEAT, or whether the hypothesis that a phonological acquisition process should include this type of minimization is too strong.

An important point to address in the complexity analysis is whether we are operating in a bounded domain. With a fixed, finite set of features to choose from and a fixed finite phoneme inventory, the problem can be considered static and solvable in constant time by memorizing or pre-calculating all the possible patterns of feature combinations and their corresponding phoneme sets. Such counter-arguments have been leveled (Kornai, 2006) at other \mathcal{NP} -completeness analyses, such as those that have shown that Optimality Theory is potentially intractable (Eisner, 2000; Idsardi, 2006). In the case at hand, however, such an argument has less traction since the domain in question is rather large (potentially hundreds of features) and it seems inevitable that some search method must be used by speakers (or phonologists) to discover the minimum number of features required. The fact that simple greedy algorithms do not always find the minimum specification, and that branch-and-bound algorithms, while efficient for practical computational use, still need to explore a large search space, prompts the question whether some alternative strategy would work particularly well with phonological feature structures proposed in the literature. This could address the problem that this intractability poses to phonological learning.

6 Conclusion

We have shown that one of the commonly assumed subtasks of acquisition of phonological generalizations—distinctive feature minimization—is computationally intractable. The decision version of the minimization problem is \mathcal{NP} -complete, and it follows that the optimization version \mathcal{NP} -hard. This is true even if one limits oneself to using only positive features, i.e. a privative feature system. Furthermore, a simple greedy strategy for solving the problem can not be attributed to purported human performance in finding minimal feature specifications since such a strategy will sometimes fail to find a minimal description with commonly proposed feature systems and phonological inventories. The problem of simply deciding whether a set of phonemes constitutes a natural class—the feature description problem—is solvable in polynomial time.

Reproducibility

Our feature systems data and code for the feature description and minimization problems is available at <https://github.com/mhulden/minphonfeat>.

Acknowledgements

Chen was partially supported by the Spanish Project TIN2017-86727-C2-2-R. We wish to thank Miikka Silfverberg for comments on an earlier version on the paper. Thanks also to participants in LING 5410 (Phonology) and LING 7410 (Phonological Theory) at The University of Colorado, whose work and feedback on various feature minimization problems prompted part of this research.

References

- Diana Bennett Archangeli. 1984. *Underspecification in Yawelmani phonology and morphology*. Ph.D. thesis, Massachusetts Institute of Technology.
- Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. Harper & Row.
- Vasek Chvatal. 1979. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235.
- George N. Clements. 1985. The geometry of phonological features. *Phonology*, 2(1):225–252.
- George N. Clements. 2003. Feature economy in sound systems. *Phonology*, 20(03):287–333.
- Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms*. Springer.
- Jason Eisner. 2000. Easy and hard constraint ranking in optimality theory. In *Proceedings of the 5th Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON)*, pages 22–33.
- Colin J. Ewen and Harry van der Hulst. 1985. Single-valued features and the nonlinear analysis of vowel harmony. *Linguistics in the Netherlands*, pages 39–48.
- Gunnar Fant. 1966. The nature of distinctive features. *STL-QPSR*, 4(1966):1.
- Michael R. Garey and David S. Johnson. 1979. Computers and intractability: A guide to the theory of NP-completeness.
- John Goldsmith. 1985. Vowel harmony in Khalkha Mongolian, Yaka, Finnish and Hungarian. *Phonology*, 2(1):253–275.
- Morris Halle. 1962. Phonology in generative grammar. *Word*, 18(1-3):54–72.
- Bruce Hayes. 2011. *Introductory Phonology*. John Wiley & Sons.
- William J. Idsardi. 2006. A simple proof that optimality theory is computationally intractable. *Linguistic Inquiry*, 37(2):271–275.
- Roman Jakobson. 1942. The concept of phoneme. In Linda R. Waugh and Monique Merville-Burston, editors, *On Language*, pages 218–241. Harvard Univ. Press, Cambridge, MA.
- Roman Jakobson, C. Gunnar M. Fant, and Morris Halle. 1951. *Preliminaries to Speech Analysis. The distinctive features and their correlates*. MIT Press, Cambridge, MA.
- Richard M. Karp. 1972. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer.
- András Kornai. 1993. The generative power of feature geometry. *Annals of Mathematics and Artificial Intelligence*, 8(1-2):37.
- András Kornai. 2006. Is OT NP-hard? *Rutgers Optimality Archive (ROA)*.
- Ailsa H. Land and Alison G. Doig. 1960. An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, pages 497–520.
- James N. MacGregor and Tom Ormerod. 1996. Human performance on the traveling salesman problem. *Attention, Perception, & Psychophysics*, 58(4):527–539.
- André Martinet. 1955. *Économie des changements phonétiques*. A. Francke, Bern, Switzerland.
- Jeff Mielke. 2008a. *The Emergence of Distinctive Features*. Oxford University Press.
- Jeff Mielke. 2008b. P-base 3. <http://pbase.phon.chass.ncsu.edu/>. Accessed: 2017-10-01.
- Amanda Miller. 2011. The representation of clicks. In Mark van Oostendorp, Colin J. Ewen, Elizabeth Hume, and Keren Rice, editors, *The Blackwell Companion to Phonology*, volume 1, pages 416–439. Blackwell.
- Michael Sipser. 2013. *Introduction to the Theory of Computation*, 2nd edition. Cengage Learning.
- Nikolai Sergeevich Trubetzkoy. 1969. *Principles of phonology*. Univ. of California Press, Berkeley.
- Florian van Vugt and Bruce Hayes. 2012. *Pheatures spreadsheet: User's manual*. Technical report, UCLA.
- Elizabeth C. Zsiga. 2012. *The sounds of language: An introduction to phonetics and phonology*. John Wiley & Sons.