Grammatical Inference on Learning Underlying Forms and Phonological Grammars

Wenyue Hua (Rutgers), Huteng Dai (Rutgers), Adam Jardine (Rutgers)

Introduction. The simultaneous inference of underlying representations (URs) and a phonological grammar from alternating surface representations (SRs) in a morphological paradigm is a core problem in phonological learning. This problem only recently has seen progress (Tesar 2014, Berger et al. 2018). We propose a solution based on the hypothesis that phonology is subregular (Heinz, 2018). We give a procedure that, given sequences of morphemes paired with SRs, learns URs and a phonological grammar that is an **Input Strictly Local** (ISL; Chandlee 2014, Chandlee & Heinz 2018) function. ISL functions make changes in the output with respect to the local information in the input. The upshot is that restrictive computational principles, combined with major principles in phonological analysis, allow for significant progress in understanding how phonological grammar and URs are learned. This method is general enough to be extended to other classes of functions that capture iterative processes, long-distance processes, and featural representations.

Learning Problem. The learner has two targets: a **lexicon function** ℓ which maps sequences of morphemes to their URs, and a **phonology function** φ which maps UR to SR. Assuming that morphological analysis has already been completed, the learner knows which sequences of morphemes are associated with which SRs. The learner's input is a finite set of SRs and the corresponding morphological concatenations, drawn from the *composition* of ℓ and φ. Above is an example of progressive assimilation: Table 1a shows the target lexicon (roots r_1, r_2, r_3 suffixes s_1, s_2) and phonology functions (t $\rightarrow$ d / d __). The learning problem is to identify ℓ and φ from finite input samples, such as in Table 1b.

a.

Lexicon func.	Phonology func.
$r_1 \overset{\ell}{\mapsto}$ /tat/	/tat-ta/ $\overset{\varphi}{\mapsto}$ [tatta]
$r_2 \overset{\ell}{\mapsto}$ /tad/	/tad-ta/ $\overset{\varphi}{\mapsto}$ [tadda]
$r_3 \overset{\ell}{\mapsto}$ /a/	/tad-da/ $\overset{\varphi}{\mapsto}$ [tadda]
$s_1 \overset{\ell}{\mapsto}$ /ta/	/a-ta/ $\overset{\varphi}{\mapsto}$ [ata]
$s_2 \overset{\ell}{\mapsto}$ /da/	... $\overset{\varphi}{\mapsto}$ ...

b.

Input data	
$r_1 s_1$	[tatta]
$r_1 s_2$	[tatda]
$r_2 s_1$	[tadda]
$r_2 s_2$	[tadda]
$r_3 s_1$	[ata]
$r_3 s_2$	[ada]

Table 1: Target functions (a) and input data (b)

Learning Procedure. We represent the learner's hypotheses for ℓ and φ with two *finite-state transducers* (FSTs; Mohri 1997) that directly express the computational properties the learner uses. The first, T_ℓ, encodes the morphemes and their phonological representations. The second, T_φ, represents the phonological environments in which segments are changed by the grammar. Since the two hypotheses must be inferred jointly, our procedure makes changes in T_ℓ until each morpheme has exactly one UR, making the opposite change in T_φ.

The learner begins by building T_ℓ as a *prefix tree transducer* (Oncina et al., 1993), a FST that represents a finite map like that in Table 1b. Morphemes and their allomorphs are parsed by identifying the longest shared initial sequences of the outputs; e.g., the longest sequence shared by the outputs of $r_1 s_1$ and $r_1 s_2$ is [tat], therefore we assign an initial guess of /tat/ for r_1. In this way, it is straightforward to discover allomorphs of each morpheme. Abbreviating the FST representation, the initial state for T_ℓ given the data in Table 1b is shown in Fig. 1a.

The learner's initial hypothesis for T_φ is the completely faithful mapping between URs and SRs, represented with an ISL transducer. The states in an ISL transducer keep track of local input environments of some size $k-1$; here we use $k=2$, i.e. each state represents the last element seen in the input. This is shown in Fig. 1b.

We assume the one-to-one correspondence between a morpheme and its underlying form. If a morpheme has more than one SRs (**inconsistency**), the learner must decide on the UR and posit a corresponding phonological process is proposed to map the proposed UR to different SRs in different environments.

Step 1: **(Inconsistency detection)** In a phonological process, if a morpheme is mapped to multiple SRs, the learner detects this inconsistency and makes modifications in the following steps. As illustrated in Figure 1, the learner detects that only s_1 has two output {ta, da} in SR.

Step 2: **(Environment)** Given several SRs, the SR that occurs in most phonological environments is the assumed UR (i.e., it is the "elsewhere" environment). Again, the environments are represented by the various states in T_φ. In our example, the allomorph [da] for s_1 only occurs after [d] (in r_2s_1 [tadda]) but [ta] occurs after both [t] and [d] (in r_1s_1 and r_3s_1, respectively). Since [ta] occurs in more environments, it is the UR.

$r_1 \overset{\ell}{\mapsto}$ tat
$r_2 \overset{\ell}{\mapsto}$ tad
$r_3 \overset{\ell}{\mapsto}$ a
$s_1 \overset{\ell}{\mapsto}$ {ta, $\overset{\text{t}}{\cancel{\text{d}}}$a }
$s_2 \overset{\ell}{\mapsto}$ da

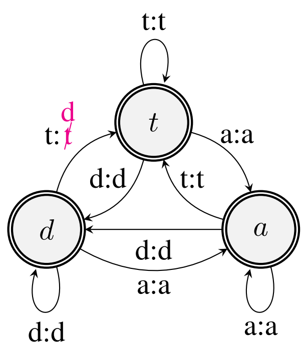

Figure 1: Hypotheses for ℓ and φ

Step 3: (**Modification**) After determining s_1's UR, /ta/, and allomorphs {[ta], [da]}, we need to modify T_ℓ and T_φ. Since the allomorph [da] differs form the UR /ta/ on the first segment, in T_ℓ, the allomorph is erased by replacing [d] with /t/. This change is illustrated in red in Figure 1a. Phonology is then proposed to map /ta/ to [da] when /ta/ comes after states of voiced consonant in T_φ. This is done by an opposite operation changing /t/'s output from [t] to [d] when following a /d/ (i.e., the /t/ transition from the /d/ state). This change is shown in Figure 1b.

Final Result: Based on the processes illustrated above, we have inferred the URs and the correspondent phonological process: the set of URs is $\{r_1 = \text{tat}, r_2 = \text{tad}, r_3 = \text{a}, s_1 = \text{ta}, s_2 = \text{da}\}$, the phonology is "/t/ assimilated to [d] after input /d/". Note that the learner has converged to a full grammar T_φ that can apply the assimilation rule to any arbitrary input.

Primary Result and Future Work The learner induces UR and phonological grammar from a range of ISL_2 functions (ISL functions for $k = 2$), including assimilation/dissimilation, deletion, epenthesis, metathesis, and opacity. There are multiple extensions of this work: (1) extending to a provably-correct algorithm able to learn all ISL_2 functions (2) extending to learn these phonology processes in ISL_k transducer for any k (3) replacing symbols with features and design the learner based on feature system. Probabilistic FSTs can be used to study learning with noisy data.

Contributions. The current proposal provides a method for induction of a lexicon and phonology from morphologically unanalyzed surface strings. This goes beyond much of the previous literature on learning phonological grammars, in which the URs are given as input to the learner (Gildea & Jurafsky 1995, Tesar 1994, *et seq.*). This is based directly on restrictive computational properties of phonological functions; however, it has considerable potential empirical coverage: 94% of phonology patterns in P-Base database (Mielke, 2004) are ISL (Chandlee & Heinz 2018). Additionally, opaque interactions are also ISL (Chandlee et al. 2018), so the procedure here can learn opaque interactions. Two recent comparable results are those of Tesar (2014) and Berger et al. (2018). However, unlike Tesar (2014), this procedure can learn opaque interactions; likewise, unlike Berger et al. (2018), it is not tied to a particular encoding and grammatical framework.