From Formal Learnability Toward Phonological Acquisition

Jane Chandlee

Haverford College

SCiL 2023
Formal learnability and subregularity

- The well-known result that phonological grammars meeting certain conditions are regular relations (Johnson, 1972; Kaplan and Kay, 1994, etc.) provided a computational restriction on what is possible in phonology.
- But it also raised a question about formal learnability, since the regular classes are not learnable in the sense of Gold (1967).
Formal learnability and subregularity

<table>
<thead>
<tr>
<th></th>
<th>languages</th>
<th>mappings</th>
</tr>
</thead>
<tbody>
<tr>
<td>regular</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>subsequential</td>
<td>—</td>
<td>✓</td>
</tr>
<tr>
<td>tier-based local</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>strictly local</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Subregularity and phonological learning

• Proof of learnability:
  1. Characteristic or sufficient data sample
  2. Algorithm given that sample can learn any language/map in the target class

• The unrealistic nature of these data samples limits the viability of these learners as models of phonological acquisition.

• They were always intended as a first step—not a solution—but there has also been a shortage of work addressing that gap.
This talk

- Start with an algorithm used to prove formal learnability: \( SI_2DLA \) (Hua and Jardine, 2021), a function decomposition approach to the problem of learning underlying representations (URs) and a phonological (UR-to-SR) mapping.
- Propose a set of modifications to enable this learner to work on a realistic problem instance (Malay).
- The utility of subregularity for phonological learning is independent of the limitations assumed by the algorithms used to prove formal learnability.
Preliminaries

- The **lexicon** is a mapping from meanings to URs of morphemes.
  - \( \mathcal{L}(\text{CAT}) = /kæt/ \)
  - \( \mathcal{L}(\text{PI}) = /z/ \)

- The **morphology** is a mapping from combinations of meanings to URs.
  - \( \mathcal{M}(\text{CAT}, \text{PI}) = /kætz/ \)

- The **phonology** is a mapping from URs to surface representations (SRs).
  - \( \mathcal{P}(kætz) = [kæts] \)

(Adapted from Hua and Jardine (2021); see also Cotterell et al. (2015))
Phonological learning problem

Given $\mathcal{P} \circ \mathcal{M}$, identify $\mathcal{L}$ and $\mathcal{P}$.

$\mathcal{P} \circ \mathcal{M}($CAT, Pl$) = [\text{kæts}]$

$\mathcal{P}(\mathcal{M}($CAT, Pl$)) = \mathcal{P}(/\text{kætz}/) = [\text{kæts}]$
In the general case, function decomposition is an unsolvable problem. But, these are not arbitrary functions.

Hua and Jardine (2021)’s algorithm capitalizes on the assumption that the phonological function is a local function.

In addition, because it’s learning $\mathcal{L}$ instead of $\mathcal{M}$, this is more like function extraction than true decomposition.
Previous work on learning URs

- Maximum Likelihood Learning of OT Grammars (Jarosz, 2006a,b, 2009, 2015)
- GLA with lexical constraints (Apoussidou, 2007)
- Constraint Demotion using contrast pairs (Merchant, 2008; Merchant and Tesar, 2008) and output-driven maps (Tesar, 2014)
- MaxEnt (Eisenstat, 2009; Wang and Hayes, 2022), MaxEnt HG (O’Hara, 2017), MaxEnt with lexical/UR constraints (Pater et al., 2012; Nelson, 2019)
- Minimum Description Length (Rasin and Katzir, 2016; Rasin et al., 2020, 2021)
- Loopy belief propagation with WFSMs (Cotterell et al., 2015)
SI$_2$DLA (Hua and Jardine, 2021)

- Target lexicon: $A = /tat/$, $B = /tad/$, $C = /a/$, $1 = /ta/$, $2 = /da/$
- Target phonology: $t \rightarrow d$ / d __
- Data: {$(A, tat), (B, tad), (1, ta), (2, da), (AA, tattat), (AB, tattad), (A1, tatta), (A2, tatda), (BA, taddat), ...$}
Step 1: Morpheme segmentation

\[ \lambda \rightarrow A \rightarrow A1 \]

\[ B: \text{tattad} \]

\[ 1: \text{tatta} \]

\[ 2: \text{tatda} \]
Step 2: State merging

• Same method used by Oncina et al. (1993) for learning subsequential (i.e., deterministic) functions (Onward Subsequential Transducer Inference Algorithm).
• States are merged provided they have the same sets of **tails**:

\[
tails_f(x) = \{(y, v) | f(xy) = uv \land u = \text{lcp}(f(x\Sigma^*))\}
\]
• Any merge that creates irreparable non-determinism is rejected.
Step 2: State merging

\[ \lambda \]

\[ A : \text{tat} \]

\[ B : \text{tad} \]

\[ 1 : \text{ta} \]

\[ 2 : \text{da} \]

\[ C : \text{a} \]

\[ A : \text{tat} \]

\[ B : \text{tad} \]

\[ 1 : \text{ta} \]

\[ 2 : \text{da} \]

\[ A : \text{tat} \]

\[ A_1 \]

\[ A_2 \]

\[ C : \text{a} \]

\[ C_1 \]

\[ C_2 \]
Result of state merging

A: dat, B: dad, C: a, 1: da, 2: da

A: tat
C: a
2: da
1: ta
Result of state merging

- Morphemes that don’t alternate can be added to the lexicon.

A: dat, B: dad, C: a, 1: da, 2: da

```
0 --> 1
0 --> B: tad
B: tad --> 0
```

A: tat
C: a
2: da
1: ta
Result of state merging

• States represent the contextual information for morphemes that do alternate.
  • How do we extract that information?
  • Which variant is the UR?

A: dat, B: dad, 1: da

0

B: tad

A: tat

1: ta

1
Under the assumption that the phonology is a 2-local function, this contextual information can be reduced to the one-suffixes that reach each state.
Locality assumption

- Under the assumption that the phonology is a 2-local function, this contextual information can be reduced to the one-suffixes that reach each state.
Locality assumption

- Under the assumption that the phonology is a 2-local function, this contextual information can be reduced to the **one-suffixes** that reach each state.
Under the assumption that the phonology is a 2-local function, this contextual information can be reduced to the **one-suffixes** that reach each state.
Step 3: UR selection

- The larger set of one-suffixes is the 'default'; the URs are the outputs from that state.
Step 4: Constructing the phonology function

- Again because of locality, the alternation affects the first segment output from the context state.
Step 4: Constructing the phonology function
Step 4: Constructing the phonology function

\[ \lambda \quad t:d \quad d:d \]
Step 4: Constructing the phonology function

\[ \lambda \]

\[ t:t, a:a \]

\[ t:d, a:a \]

\[ d:d \]

\[ d:d \]
Limitations

\[ \mathcal{P} \] is assumed to be a **left-simplex 2-local function**: exactly one segment changes in exactly one context.
Limitations

• $\mathcal{P}$ is assumed to be a **left-simplex 2-local function**: exactly one segment changes in exactly one context.
• The sufficient sample used for the proof includes $M \leq 2$, where $M$ is the set of morphemes.
  • e.g., if $A = \text{CAT}$, and $1 = \text{Pl}$, this includes $(A, [kæt])$ and $(A1, [kæts])$, but also $(AA, [kætkæt]), (1, [z]), (1A, [zkæt])$, etc.
Limitations

- \( \mathcal{P} \) is assumed to be a **left-simplex 2-local function**: exactly one segment changes in exactly one context.
- A sufficient sample includes \( M^{\leq 2} \), where \( M \) is the set of morphemes.
  - e.g., if \( A = \text{CAT} \), and \( 1 = \text{Pl} \), this includes \((A, [kæt])\) and \((A1, [kæts])\), but also \((AA, [kætkæt]), (1, [z]), (1A, [zkæt])\), etc.
- Proposed modifications for a realistic problem instance:
  1. Use morphological domain information in state merging
  2. Natural class criterion to select the UR
  3. Canonical local FST to enact multiple phonological changes
Malay alternations

Johor Malay (Austronesian; Peninsular Malaysia; Onn, 1976) /pəŋ-/ (nominalizer) and /məŋ-/ (active voice) both have five variants.

- Before obstruents they are subject to nasal place assimilation:
  /pəŋ-boroŋ/ [pəmboroŋ] ‘wholesaler’
  /pəŋ-jahit/ [pəŋjahit] ‘tailor’
  /pəŋ-daki/ [pəndaki] ‘climber’
  /pəŋ-gali/ [pəŋgali] ‘digger’
  /pəŋ-arah/ [pəŋarah] ‘director’

- Before voiceless obstruents the obstruent also deletes (‘fusion’):
  /pəŋ-karaŋ/ [pəŋaraŋ] ‘author’
  /pəŋ-samun/ [pəŋamun] ‘robber’
  /pəŋ-tari/ [pəŋari] ‘dancer’
Malay alternations

- Before sonorant consonants the prefix nasal deletes:

  /pəŋ-/ [pəlayan] ‘waitress’
  /pəŋ-ŋaŋi/ [pəŋaŋi] ‘singer’
  /pəŋ-malu/ [pəmalu] ‘shame’
  /pəŋ-rayu/ [pərayu] ‘appeal’
Additional processes

- Final devoicing and regressive devoicing

  /jawab/  \([jawap]\)  ‘to answer’
  /pəŋ-jawab-an/  \([pəŋjawaban]\)  ‘the answering’
  /məŋ-jawab-kan/  \([məŋjawapkan]\)  ‘to cause to answer for’

- Velar stops become glottal in coda position

  /masak/  \([masaʔ]\)  ‘to cook’
  /pəŋ-masak-an/  \([pəmasakan]\)  ‘the cooking’
  /məŋ-masak-kan/  \([məmasaʔkan]\)  ‘to cause to cook for’
Additional processes

• /r/ deletes in coda position

/kisar/ [kisa] ‘revolve’
/kisar-an/ [kisaran] ‘revolution’
/kisar-kan/ [kisakan] ‘to cause to revolve for’

• Word-final /a/ reduces to schwa.

/bawa/ [bawə] ‘to carry’
/bawa-kan/ [bawakan] ‘to cause to carry for’
Data coding

<table>
<thead>
<tr>
<th>A</th>
<th>/ikat/</th>
<th>‘tie’</th>
<th>B</th>
<th>/asut/</th>
<th>‘instigate’</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>/gali/</td>
<td>‘dig’</td>
<td>E</td>
<td>/bayar/</td>
<td>‘pay’</td>
</tr>
<tr>
<td>F</td>
<td>/bawa/</td>
<td>‘carry’</td>
<td>I</td>
<td>/dakap/</td>
<td>‘embrace’</td>
</tr>
<tr>
<td>L</td>
<td>/ladaŋ/</td>
<td>‘farm’</td>
<td>M</td>
<td>/naik/</td>
<td>‘ascend’</td>
</tr>
<tr>
<td>N</td>
<td>/jawab/</td>
<td>‘answer’</td>
<td>P</td>
<td>/cërca/</td>
<td>‘revile’</td>
</tr>
<tr>
<td>S</td>
<td>/rompak/</td>
<td>‘rob’</td>
<td>T</td>
<td>/ŋəŋa/</td>
<td>‘open’</td>
</tr>
<tr>
<td>U</td>
<td>/main/</td>
<td>‘play’</td>
<td>V</td>
<td>/ŋəni/</td>
<td>‘sing’</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>/məŋ-/</th>
<th>active</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>/pəŋ-/</td>
<td>nominalizer</td>
</tr>
<tr>
<td>3</td>
<td>/-i/</td>
<td>causative</td>
</tr>
<tr>
<td>4</td>
<td>/-kan/</td>
<td>causative benefactive</td>
</tr>
<tr>
<td>5</td>
<td>/-an/</td>
<td>nominalizer</td>
</tr>
</tbody>
</table>

- Because the focus is on learning $\mathcal{P}$, not $\mathcal{M}$, morphemes are already ordered in the dataset.
Morpheme segmentation

- Attested sequences are sufficient for onwardness to segment morphemes.

/bayar/, ‘pay’
(1E, məmbaya, ‘to pay (active)’)  
(2E, pəmbaya, ‘payment’)

\[
\begin{array}{c}
1E \\
\text{1:məmbaya} \\
E: \lambda
\end{array}
\]
Morpheme segmentation

/bayar/, 'pay'
(1E, məmbaya, 'to pay (active)')
(2E, pəmbaya, 'payment')
Morpheme segmentation

/bayar/, ‘pay’
(1E, məmbaya, ‘to pay (active)’)
(2E, pəmbaya, ‘payment’)
(E, baya, ‘to pay’)

1E

1:məmbaya

#:baya

2:pəmbaya

2E

E

E

E:λ

λ
Morpheme segmentation

/ayar/, ‘pay’
(1E, məmbaya, ‘to pay (active)’)
(2E, pəmbaya, ‘payment’)
(E, baya, ‘to pay’)

1E
1:məm
#:λ
2E
2:pəm

E:baya

λ
Morpheme segmentation

/bayar/, ‘pay’
(2E5, pəmbayaran, ‘payment’)

\begin{center}
\begin{tikzpicture}
  \node[draw, circle] (2E5) {2E5 \nodepart{two} \text{pəmbayaran}};
  \node[draw, circle] (E5) [right of=2E5] {E5 \nodepart{two} \text{λ}};
  \node[draw, circle] (5) [right of=E5] {5 \nodepart{two} \text{λ}};
  \node[draw, circle] (lambda) [right of=5] {λ};
  \draw[->] (2E5) -- (E5) node[midway, above] {2: pəmbayaran};
  \draw[->] (E5) -- (5) node[midway, above] {E: λ};
  \draw[->] (5) -- (lambda) node[midway, above] {5: λ};
\end{tikzpicture}
\end{center}
Morpheme segmentation

/bayar/, ‘pay’
(2E5, pəmbayaran, ‘payment’)
Morpheme segmentation

/bayar/, ‘pay’
(2E5, pəmbayaran, ‘payment’)
(E5, bayaran, ‘payment’)

![Diagram showing morpheme segmentation]
Morpheme segmentation

/bayar/, ‘pay’
(2E5, pəmbayaran, ‘payment’)
(E5, bayaran, ‘payment’)

\[ \lambda \]

\[ \Rightarrow \]

2E5 \to 2:pəm \to E:bayaran \to 5 \to 5:λ \to \lambda

#:λ
Morpheme segmentation

/bayar/, ‘pay’
(2E5, pəmbayaran, ‘payment’)
(E5, bayaran, ‘payment’)

```
2E5 → 2:əm
E5 → E:bayaran
B5 → B:asutan
5 → 5:λ
λ
```
Morpheme segmentation

/bayar/, ‘pay’
(2E5, pəmbayaran, ‘payment’)
(E5, bayaran, ‘payment’)

![Diagram]

2E5 → 2:pəm
E5 ← E:bayar
5 ← 5:an
λ

B5

#:λ

B:asut
Sufficient sample (revised)

• Dataset includes all stems unaffixed, with each prefix, with each suffix, and with allowable prefix-suffix combinations.

/bayar/

baya  ‘to pay’
membaya  ‘to pay (active)’
bayari  ‘to cause to pay’
bayakan  ‘to cause to pay for’
membayari  ‘to cause to pay (active)’
membayakan  ‘to cause to pay for (active)’
pembaya  ‘payment’

• Still somewhat idealistic, but more on this later.
State merging using domain information

• Modification: incorporate morphological knowledge using domain information.
• Oncina et al. (1993)’s state merging algorithm by default targets a total function.
• An alternative version (Oncina and Varò, 1996) adds a condition: merged states must have the same tails and reach the same state in an FSA representation of the domain of the function (in this case $M$).
Domain FSA

0 → 1
prefix

0 → 2
stem

2 → 3
suffix
Prefix alternations only

N: jawab
F: bawa
L: landaŋ
I: dakap
A: ikat
D: gali
Collect one-suffixes

\{j, c\} → 1:məŋ
N:jawab

\{m, i, a, k, g\} → F:bawa

\{d\} → 1:məm

\{l, r, n, m, η, η\} → 1:mən
L:landan

1:məŋ
3:i
4:an
5:kan
A:ikat
D:gali
Modification: use natural classes

{j, c} → {m, i, a, k, g} → {b} → {d} → {l, r, n, m, η, η}

N: jawab
F: bawa
L: landaŋ
1: məŋ
1: məm
1: mən
1: məŋ
Select UR using natural classes

{i, a, k, g}
{d}
{j, c}

1: mən
F: bawa
N: jawab

{m, i, a, k, g}

1: məm
1: mən
L: landanə

1: məน

1: məν
3: i
4: an
5: kan
A: iKat
D: gali

{d}
{l, r, n, m, ɳ, ɲ}
Construct phonology function

\{m, i, a, k, g\}

\{b\}

F:bawa

1:məm

\lambda

b:b

η:m

1:məm

1:məm
Modification: use canonical local FST

- Separate states for all segments (=all potential contexts).
- Starts out as the identity function (‘faithfulness bias’) and is modified as patterns are discovered.
Additional processes

- Final devoicing and regressive devoicing
- \{k, g\} → ? in coda position
- /r/ deletes in coda position
- Word-final /a/-reduction
- All of these are regressive and 2-local
Output of state merging

1
\[ \lambda \]
2

bawa
məm
ikat, gali
suffixes
məη, ikat, gali
bawə
məm

3
Start state ambiguity

\[ \{ \text{m, i, g} \} \]

\[ \text{məŋ, iKat, gali} \]

\[ \text{bawə} \]

\[ \{ \# \} \]

\[ 2 \]
Interim summary

- With plausible modifications (e.g., morphological domain knowledge, natural classes, faithfulness bias), UR learning via function decomposition is viable using attested sequences that reflect multiple phonological changes in multiple contexts.

$\eta \rightarrow [\alpha \text{place}] / \_ [\text{–son}, \alpha \text{place}]$

$\eta \rightarrow \emptyset / \_ [\text{+son}, \text{–syl}]$

$[\text{–son}] \rightarrow [\text{–voice}] / \_ \{[\text{–voice}], \#\}$

$\{k, g\} \rightarrow ? / \_ \{C, \#\}$

$r \rightarrow \emptyset / \_ \{C, \#\}$

$a \rightarrow \text{ȃ} / \_ \#$
Opacity

- Counterfeeding: /r/-deletion and final /a/-reduction

/bakar/

/a/-reduction —
/r/-deletion [baka]
[baka]
‘to burn’

- Opacity is handled with ‘direct mapping’ of all changes to the UR (Kenstowicz and Kisseberth, 1977, 1979).
Going further: fusion

• /pəŋ-ˌtari/ \( \rightarrow \) [pənari], ‘dancer’
Going further: fusion

- /pən-tari/ → [pənari], ‘dancer’
Going further: fusion

- /pəŋ-tari/ → [pənari], ‘dancer’
Going further: epenthesis

• Glide epenthesis if first vowel is high:
  /bantu-an/  [bantuwan]  ‘aid, relief’
  /tari-an/  [tariyan]  ‘dance’

• Otherwise glottal epenthesis:
  /məŋ-gula-i/  [məŋgulaʔi]  ‘to cause to sweeten’
  /pəŋ-buka-an/  [pəmbukaʔan]  ‘opening’

• At prefix boundary, always glottal:
  /di-ambil/  [diʔambel]  ‘to take (passive)’

• Nonderived environment blocking:
  /main/  [main]  ‘to play’
  /naik/  [naik]  ‘to ascend’
Going further: epenthesis

- Adirectional and still 2-local, but requires a more detailed UR decision procedure

/ɕəɾca/, ‘to revile’

SRs: [ɕəɾcə] [ɕəɾca?] [ɕəɾca]

Contexts: _# _V _C
Going further: epentheses

/cərca\/  /cərca?/  /cərca/

ə → a / ___ C  a? → ə / ___ #  a → ə / ___ #
ə → a? / ___ V  ? → ø / ___ C  ø → ? / V ___ V

*[carcakan]*  ? isn’t phonemic
Idealistic → naturalistic data

• Less idealistic data coverage may be possible: under certain conditions state merging can compensate for initial segmentation errors
• Generalization using features (Markowska (in prep.))
Free variation

• /s/ optionally becomes [h] in coda position

/kipas/ [kipas] ∼ [kipah] ‘fan’
/kipas-kan/ [kipaskan] ∼ [kipahkan] ‘to cause to fan for’

• Optional deletion with some /c/-initial stems under prefixation.

/məŋ-cium/ [məŋciyum] ∼ [məniyum] ‘to kiss (active)’
/məŋ-cubit/ [məncubit] ∼ [mənubit] ‘to pinch (active)’
Idealistic $\rightarrow$ naturalistic data

- Semi-determinism for optionality (Beros and de la Higuera, 2016; Heinz, 2020)
  - $X:\{\text{tulis, tulih}\}$
Remaining limitations

- Clearly not all phonological patterns are regressive and 2-local.
- Computational locality as defined here comes in other flavors (larger $k$, input versus output, tier-based, etc.), all of which are tied to well-studied hierarchies of formal languages and function classes (Heinz, 2018).
- We know the structure of the search space, but how is it navigated?
  - e.g., ‘wrong direction’ is cued by non-determinism in the output of state merging
Conclusions

• The limitations assumed in proofs of formal learnability need not detract from the larger point that subregularity has utility for phonological learning.

• Testing these learners on actual problem instances clarifies those limits and points to methods for overcoming them.

• Studying phonological learning problems from a variety of perspectives will further our understanding of both the problem itself and the nature of the object being learned.
References


References


