Computation, learning, and typology

Class 3: More formal limits

Adam Albright and Eric Baković

CreteLing 2023 — July 2023

creteling2023.phonology.party

Some other invocations of phonological simplicity

Simplicity in the face of phonetic motivation

Gordon (2002):51

The author proposes that syllable weight is driven by considerations of phonetic effectiveness and **phonological simplicity**, and that the phonetically best distinctions are those which divide syllables into groups that are phonetically most distinct from each other. **Phonologically complex distinctions are those which exceed an upper threshold in the number of phonological predicates to which they refer.** It is claimed that languages adopt weight distinctions that are phonetically most effective **without being overly complex phonologically**. Syllable weight thus reflects **a compromise between phonetic and phonological factors**.

Human learners and learning algorithms

- Moreton & Pater (2012); Pater & Moreton (2012): "... make predictions about skews toward simplicity in typology, in conjunction with a model of iterated learning."
- Durvasula & Liter (2020): "There is a simplicity bias when generalising from ambiguous data."
- Albright & Hayes (2002): Minimum Generalization Learner
- Barke et al. (2019): SyPhon: separating out inferences
- Ellis et al. (2022): Bayesian Program Induction

Consistency of rule interactions

A fragment of the phonological grammar of Yokuts

	/mut+t/	/ʔuːt+ka/	/paxaːt+t/	/ʔuːt+hn/
$\varnothing \longrightarrow \text{i} \,/\, \text{C} \_\_ \text{C} \left\{ \begin{matrix} \text{C} \\ \# \end{matrix} \right\}$	|mutit|	|ʔuːtka|	|paxaːtit|	|ʔuːthin|
$[\alpha\text{high}] \longrightarrow [\beta\text{round}] \,/\, \begin{bmatrix} \alpha\text{high} \\ \beta\text{round} \end{bmatrix} \text{C}_0 \_\_$	|mutut|	|ʔuːtka|	|paxaːtit|	|ʔuːthun|
$[+\text{long}] \longrightarrow [-\text{high}]$	|mutut|	|ʔoːtka|	|paxaːtit|	|ʔoːthun|
$\text{V} \longrightarrow [-\text{long}] \,/\, \_\_ \text{C} \left\{ \begin{matrix} \text{C} \\ \# \end{matrix} \right\}$	|mutut|	|ʔotka|	|paxaːtit|	|ʔothun|
	[mutut]	[ʔotka]	[paxaːtit]	[ʔothun]
	‘swear’ (aor.pass.)	‘steal’ (fut.pass.)	‘mourn’ (aor.pass.)	‘steal’ (aorist)

epenthesis feeds harmony	lowering counterxeeds harmony; shortening counterbleeds lowering	epenthesis bleeds shortening	*everything everywhere all at once*

Different strokes for different folks

- Anderson (1974): local ordering
  - A $<$ B, B $<$ C, but C $<$ A
  - interaction markedness: feeding $\succ$ neutral $\succ$ bleeding
- Kiparsky (1985): cyclicity
  - A $<$ B $<$C within a cycle, but C $<$ A across cycles
  - must be independently motivated by the morphology

Consistency of rankings

A possible rule-based grammar

- Stop+liquid cluster simplification
  - liquid $\rightarrow \varnothing$ / stop ___ #
- r+sonorant epenthesis
  - $\varnothing \rightarrow$ ə / r ___ sonorant
- Assume lexicon with a range of structures

UR	/mat/	/kab/	/patr/	/tabl/	/parl/	/tarm/
TR simpl.	—	—	pat	tab	—	—
rN epenth.	—	—	—	—	parəl	tarəm
SR	[mat]	[kab]	[pat]	[tab]	[parəl]	[tarəm]

OT translation: Stop+liquid cluster simplification

- Constraints
  - Markedness: $*\begin{bmatrix} -\text{son} \\ -\text{cont} \end{bmatrix} \begin{bmatrix} +\text{cons} \\ +\text{son} \\ +\text{cont} \end{bmatrix} = \text{*TR}$
  - Faithfulness 1: Max(C)
  - Faithfulness 2: Dep(V)
- Ranking: *TR, Dep(V) $\gg$ Max(C)

/tabl/	$\mathbb{M}$:*TR	$\mathbb{F}$:Dep(V)	$\mathbb{F}$:Max(C)
☞ [tab]			*
[tabl]	*W		L
[tabəl]		*W	L

OT translation: r+sonorant epenthesis

- Constraints
  - Markedness: $*\text{r}\begin{bmatrix} -\text{syl} \\ +\text{son} \end{bmatrix} = *\text{rN}$
  - Faithfulness 1: Max(C)
  - Faithfulness 2: Dep(V)

- Ranking: *rN, Max(C) $\gg$ Dep(V)

/parl/	$\mathbb{M}$:*rN	$\mathbb{F}$:Max(C)	$\mathbb{F}$:Dep(V)
☞ [parəl]			*
[parl]	*W		L
[par]		*W	L

A ranking contradiction!

- Violations of both *TR and *rN could be repaired by either epenthesis or deletion
- Epenthesis requires MAX $\gg$ DEP
- Deletion requires DEP $\gg$ MAX
- Without further stipulations, expect the same repair for both configurations

A typological consequence of OT

- Consistency of ranking makes predictions about what *combinations of processes* a grammar contains
- Surprisingly, this prediction has not been tested extensively
- Perhaps promising that even with this limitation, OT has been as successful as it has been so far?

A caveat

- Example assumes a single relevant MAX(C) constraint, and a single relevant DEP(V) constraint
- Several maneuvers available to tackle different repairs for different configurations
  - Generally preferred repair violates different $\mathbb{M}$
  - Different $\mathbb{F}$ constraints, sensitive to context
- Example chosen to involve C_2=/l/ in both cases, similar contexts, etc., but even so one might imagine augmenting the constraint set to accommodate it
- Nonetheless, expect typological bias for unified repairs

“Simplicity” of rankings

Evaluation metrics in learning

The use of the evaluation metric in SPE

- Given two grammars that are consistent with the data
- Choose the one that is ‘valued’ more highly by the evaluation metric

Is there an equivalent of this for OT constraint rankings?

Simplicity of rankings

One intuitive idea

- Grammars with fewer crucial rankings are simpler than grammars with more crucial rankings
  - A, B, C
  - A, B $\gg$ C
  - A $\gg$ B $\gg$ C

But: recall that we assume that all OT grammars are total rankings Tesar & Smolensky (2000)

Reinterpretation in terms of learning

- Learners use data to establish rankings, by adjusting the “strength” of constraints
  - Crucial rankings demanded by the data
  - Perhaps additional rankings that happen to emerge during the course of learning, due to the way in which constraint strengths are adjusted
- Rankings between other constraints are arbitrary, and could differ from speaker to speaker, since data doesn’t require/guarantee them
- Question: if the data is ambiguous between two grammars, does the learning procedure end up favoring one interpretation?
  - If so, this bias could shape the typology

An example: Stanton (2016)

- Data about stress placement may be ambiguous
  - Two analyses agree for short words, but make different predictions for (unseen) longer words
- Learning procedure may prefer one analysis over the other
- This analysis also corresponds to the typologically preferred pattern

Assigning stress without feet

- Constraints on position of stresses and intervals between them

ALIGN-L, ALIGN-R	Assign one * for each σ separating stress from the L/R edge of the word
NONFINALITY	Assign one * for stress on the final σ
*CLASH	Assign one * for each sequence of two stressed σ's
*LAPSE	Assign one * for each sequence of two stressless σ's
*LAPSE-L/R	Assign one * if neither of the initial/final two σ's is stressed
*EXTENDLAPSE	Assign one * for each sequence of three stressless σ's
*EXTLAPSE-L/R	Assign one * if none of the initial/final three σ's is stressed

Assigning stress without feet (*cont.*)

- E.g., antepenultimate stress

/σσσσσσ/	*ExtLapse(R)	*ExtLapse(L)	Align(L)	Align(R)
a. όσσσσσσ	*! W		L	****** W
b. σόσσσσσ	*! W		* L	***** W
c. σσόσσσσ	*! W		** L	**** W
d. σσσόσσσ	*! W	*	*** L	*** W
☞ e. σσσσόσσ		*	****	**
f. σσσσσόσ		*	*****! W	* L
g. σσσσσσό		*	*****!* W	L

- The insight behind the analysis
  - Stress as far left as possible (Align(L) ≫ Align(R))
  - But not more than 3σ from the end (ExtLapse(R) ≫ Align(L))

The midpoint pathology (Kager, 2012; Stanton, 2016)

- Short words: can satisfy both *(Extended)Lapse(L) and *(Extended)Lapse(R), by keeping stress towards the middle of the word

/σσσσσ/			*ExtLapse(L)	*ExtLapse(R)
☞	a.	σσόσσ		
	b.	σσσσσ́	*! W	
	c.	όσσσσ		*! W

- Longer words: can't satisfy both, so satisfy the higher-ranked one with stress at the relevant edge

/σσσσσσσ/			*ExtLapse(L)	*ExtLapse(R)
	a.	σσσόσσσ	*! W	*
☞	b.	όσσσσσσ		*
	c.	σσσσσσσ́	*! W	L

Example: a ‘midpoint-stress’ language

*ExtLapse(L) ≫ *ExtLapse(R) ≫ Align(L) ≫ Align(R)

2 syl	σ́σ
3 syl	σ́σσ
4 syl	σσ́σσ
5 syl	σσσ́σσ
6 syl	σ́σσσσσ
7 syl	σ́σσσσσσ
8 syl	σ́σσσσσσσ

- *ExtLapse(L/R) ≫ Align(L/R): stress can move inside word to avoid extended lapse
- *ExtLapse(L) ≫ *ExtLapse(R): when too long to satisfy both, it moves to the left side of the word
- Align(L) ≫ Align(R): when on the left side of the word, it falls on the very first syllable

Comparing the grammars

- Antepenultimate (AP) stress
  - *ExtLapse(R) $\gg$ Align(L) $\gg$ Align(R)
  - *ExtLapse(R) $\gg$ *ExtLapse(L), but ranking of *ExtLapse(L) w.r.t. other constraints does not matter
- AP midpoint stress
  - *ExtLapse(L) $\gg$ *ExtLapse(R) $\gg$ Align(L) $\gg$ Align(R)
  - Midpoint grammar requires additional crucial rankings
  - Question: if learners receive ambiguous data, would they favor a ranking consistent with AP stress?

Interpreting ambiguous data

AP stress

2 syl	όσ
3 syl	όσσ
4 syl	σόσσ
5 syl	σσόσσ
6 syl	σσσόσσ
7 syl	σσσσόσσ

“AP” midpoint

2 syl	όσ
3 syl	όσσ
4 syl	σόσσ
5 syl	σσόσσ
6 syl	όσσσσσ
7 syl	όσσσσσσ

Interpreting ambiguous data

AP stress	
2 syl	σ́σ
3 syl	σ́σσ
4 syl	σσ́σσ
5 syl	σσσ́σσ

“AP” midpoint	
2 syl	σ́σ
3 syl	σ́σσ
4 syl	σσ́σσ
5 syl	σσσ́σσ

- The two patterns are ambiguous in words <6σ
- Question: what grammar would a learning algorithm prefer, given ambiguous data of this sort?
- We try first with Recursive Constraint Demotion
  - All constraints start out ranked equally
  - At each step, demote constraints that favor losers (L’s)

Learning from short words: constraint demotion

/σσ/	*ExtLapse(L)	*ExtLapse(R)	Align(L)	Align(R)
☞ a. όσ				*
b. σό			* W	L

/σσσ/	*ExtLapse(L)	*ExtLapse(R)	Align(L)	Align(R)
☞ a. όσσ				**
b. σόσ			* W	* L
c. σσό			** W	L

/σσσσ/	*ExtLapse(L)	*ExtLapse(R)	Align(L)	Align(R)
a. όσσσ		* W	L	*** W
☞ b. σόσσ			*	**
c. σσόσ			** W	* L
d. σσσό	* W		*** W	L

/σσσσσ/	*ExtLapse(L)	*ExtLapse(R)	Align(L)	Align(R)
a. όσσσσ		* W	L	**** W
b. σόσσσ		* W	* L	*** W
☞ c. σσόσσ			**	**
d. σσσόσ	* W		*** W	* L
e. σσσσό	* W		**** W	L

Learning from short words: constraint demotion

/σσ/	*ExtLapse(L)	*ExtLapse(R)	Align(L)	Align(R)
☞ a. όσ				*
b. σό			* W	L

/σσσ/	*ExtLapse(L)	*ExtLapse(R)	Align(L)	Align(R)
☞ a. όσσ				**
b. σόσ			* W	* L
c. σσό			** W	L

/σσσσ/	*ExtLapse(L)	*ExtLapse(R)	Align(L)	Align(R)
a. όσσσ		* W	L	*** W
☞ b. σόσσ			*	**
c. σσόσ			** W	* L
d. σσσό	* W		*** W	L

/σσσσσ/	*ExtLapse(L)	*ExtLapse(R)	Align(L)	Align(R)
a. όσσσσ		* W	L	**** W
b. σόσσσ		* W	* L	*** W
☞ c. σσόσσ			**	**
d. σσσόσ	* W		*** W	* L
e. σσσσό	* W		**** W	L

Learning from short words: constraint demotion

/σσ/	*ExtLapse(L)	*ExtLapse(R)	Align(L)	Align(R)
☞ a. όσ				*
b. σό			* W	L

/σσσ/	*ExtLapse(L)	*ExtLapse(R)	Align(L)	Align(R)
☞ a. όσσ				**
b. σόσ			* W	* L
c. σσό			** W	L

/σσσσ/	*ExtLapse(L)	*ExtLapse(R)	Align(L)	Align(R)
a. όσσσ		* W	L	*** W
☞ b. σόσσ			*	**
c. σσόσ			** W	* L
d. σσσό	* W		*** W	L

/σσσσσ/	*ExtLapse(L)	*ExtLapse(R)	Align(L)	Align(R)
a. όσσσσ		* W	L	**** W
b. σόσσσ		* W	* L	*** W
☞ c. σσόσσ			**	**
d. σσσόσ	* W		*** W	* L
e. σσσσό	* W		**** W	L

Two possible refinements

*ExtLapse(R) ≫ *ExtLapse(L)

/σσσσσσ/	*ExtLapse(R)	*ExtLapse(L)	Align(L)	Align(R)
a. όσσσσσσ	*! W		L	****** W
b. σόσσσσσ	*! W		* L	***** W
c. σσόσσσσ	*! W		** L	**** W
d. σσσόσσσ	*! W	*	*** L	*** W
☞ e. σσσσόσσ		*	****	**
f. σσσσσόσ		*	*****! W	* L
g. σσσσσσό		*	*****!* W	L

/σσσσσσσ/	*ExtLapse(R)	*ExtLapse(L)	Align(L)	Align(R)
a. όσσσσσσσ	*! W			******* W
b. σόσσσσσσ	*! W		* L	****** W
c. σσόσσσσσ	*! W		** L	***** W
d. σσσόσσσσ	*! W	*	*** L	**** W
e. σσσσόσσσ	*! W	*	**** L	*** W
☞ f. σσσσσόσσ		*	*****	**
g. σσσσσσόσ		*	******! W	* L
h. σσσσσσσό		*	******!* W	L

Antepenultimate stress

2 syl	όσ	5 syl	σσόσσ
3 syl	όσσ	6 syl	σσσόσσ
4 syl	σόσσ	7 syl	σσσσόσσ

*ExtLapse(L) ≫ *ExtLapse(R)

/σσσσσσ/	*ExtLapse(L)	*ExtLapse(R)	Align(L)	Align(R)
☞ a. όσσσσσσ		*		******
b. σόσσσσσ		*	* W	***** L
c. σσόσσσσ		*	** W	**** L
d. σσσόσσσ	*! W	*	*** W	*** L
e. σσσσόσσ	*! W	L	**** W	** L
f. σσσσσόσ	*! W	L	***** W	* L
g. σσσσσσό	*! W	L	****** W	L

/σσσσσσσ/	*ExtLapse(L)	*ExtLapse(R)	Align(L)	Align(R)
☞ a. όσσσσσσσ		*		*******
b. σόσσσσσσ		*	* W	****** L
c. σσόσσσσσ		*	** W	***** L
d. σσσόσσσσ	*! W	*	*** W	**** L
e. σσσσόσσσ	*! W	*	**** W	*** L
f. σσσσσόσσ	*! W	L	***** W	** L
g. σσσσσσόσ	*! W	L	****** W	* L
h. σσσσσσσό	*! W	L	******* W	L

Midpoint system

2 syl	όσ	5 syl	σσόσσ
3 syl	όσσ	6 syl	όσσσσσ
4 syl	σόσσ	7 syl	όσσσσσσ

Ambiguity in short words

- Using RCD, learners exposed to ambiguous ($\leq 5\sigma$) data have no reason to prefer either consistent AP or AP midpoint stress
- Where does the antepenultimate bias come from?

The learning algorithm matters

- RCD doesn't explain antepenultimate bias, because in short words, *ExtLapse(L) and *ExtLapse(R) are 'W-only' constraints $\Rightarrow$ remain highly ranked
- Stanton's conjecture: human learners actually use a ranking algorithm that doesn't just demote L's, but also promotes W's (**??**)

The learning algorithm matters (*cont.*)

- Why this will help:
  - Short words give lots of evidence for Align(L) $\gg$ Align(R)
  - If the learner demotes Align(R) and *promotes* Align(L), then Align(L) will end up above other markedness constraints
  - Similarly, 4–5σ provide evidence for *ExtLapse(R) $\gg$ Align(L), causing it to be promoted
  - Consequence: *ExtLapse(L) is ‘left in the dust’ (not promoted unless you get 6$+$ syllable words)

Learning from short words: promotion and demotion

/σσ/	*ExtLapse(L)	*ExtLapse(R)	←Align(L)	Align(R)→
☞ a. όσ				*
b. σό			* W	L

/σσσ/	*ExtLapse(L)	*ExtLapse(R)	←Align(L)	Align(R)→
☞ a. όσσ				**
b. σόσ			* W	* L
c. σσό			** W	L

/σσσσ/	*ExtLapse(L)	*ExtLapse(R)	←Align(L)	Align(R)→
a. όσσσ		* W	L	*** W
☞ b. σόσσ			*	**
c. σσόσ			** W	* L
d. σσσό	* W		*** W	L

/σσσσσ/	*ExtLapse(L)	*ExtLapse(R)	←Align(L)	Align(R)→
a. όσσσσ		* W	L	**** W
b. σόσσσ		* W	* L	*** W
☞ c. σσόσσ			**	**
d. σσσόσ	* W		*** W	* L
e. σσσσό	* W		**** W	L

Learning from short words: promotion and demotion

/σσ/	ALIGN(L)←	*EXTLAPSE(L)	*EXTLAPSE(R)	→ALIGN(R)
☞ a. σ́σ				*
b. σσ́	* W			L

/σσσ/	ALIGN(L)←	*EXTLAPSE(L)	*EXTLAPSE(R)	→ALIGN(R)
☞ a. σ́σσ				**
b. σσ́σ	* W			* L
c. σσσ́	** W			L

/σσσσ/	ALIGN(L)←	*EXTLAPSE(L)	*EXTLAPSE(R)	→ALIGN(R)
a. σ́σσσ	L		* W	*** W
☞ b. σσ́σσ	*			**
c. σσσ́σ	** W			* L
d. σσσσ́	*** W	* W		L

/σσσσσ/	ALIGN(L)←	*EXTLAPSE(L)	*EXTLAPSE(R)	→ALIGN(R)
a. σ́σσσσ	L		* W	**** W
b. σσ́σσσ	* L		* W	*** W
☞ c. σσσ́σσ	**			**
d. σσσσ́σ	*** W	* W		* L
e. σσσσσ́	**** W	* W		L

Learning from short words: promotion and demotion

/σσ/	←Align(L)→	*ExtLapse(L)	←*ExtLapse(R)	←Align(R)→
☞ a. σ́σ				*
b. σσ́	* W			L

/σσσ/	←Align(L)→	*ExtLapse(L)	←*ExtLapse(R)	←Align(R)→
☞ a. σ́σσ				**
b. σσ́σ	* W			* L
c. σσσ́	** W			L

/σσσσ/	←Align(L)→	*ExtLapse(L)	←*ExtLapse(R)	←Align(R)→
a. σ́σσσ	L		* W	*** W
☞ b. σσ́σσ	*			**
c. σσσ́σ	** W			* L
d. σσσσ́	*** W	* W		L

/σσσσσ/	←Align(L)→	*ExtLapse(L)	←*ExtLapse(R)	←Align(R)→
a. σ́σσσσ	L		* W	**** W
b. σσ́σσσ	* L		* W	*** W
☞ c. σσσ́σσ	**			**
d. σσσσ́σ	*** W	* W		* L
e. σσσσσ́	**** W	* W		L

Learning from short words: promotion and demotion

/σσ/	*ExtLapse(R)←	←Align(L)→	*ExtLapse(L)	←Align(R)→
☞ a. όσ				*
b. σό		* W		L

/σσσ/	*ExtLapse(R)←	←Align(L)→	*ExtLapse(L)	←Align(R)→
☞ a. όσσ				**
b. σόσ		* W		* L
c. σσό		** W		L

/σσσσ/	*ExtLapse(R)←	←Align(L)→	*ExtLapse(L)	←Align(R)→
a. όσσσ	* W	L		*** W
☞ b. σόσσ		*		**
c. σσόσ		** W		* L
d. σσσό		*** W	* W	L

/σσσσσ/	*ExtLapse(R)←	←Align(L)→	*ExtLapse(L)	←Align(R)→
a. όσσσσ	* W	L		**** W
b. σόσσσ	* W	* L		*** W
☞ c. σσόσσ		**		**
d. σσσόσ		*** W	* W	* L
e. σσσσό		**** W	* W	L

Stepping back: the approach, more generally

- Some unattested systems may be possible to capture grammatically, but are difficult to learn
- Goal: theory of grammatical learning that predicts that learners, when exposed to typical input from a ‘difficult’ pattern, systematically misacquire it as a different, more commonly attested pattern
- Potential to explain not only unattested systems, but also rare systems (which we can’t exclude as impossible grammars, anyway)
- Converging evidence: acquisition data, learning in the lab?

References

Albright, Adam, and Bruce Hayes. 2002. Modeling English past tense intuitions with minimal generalization. *Sigphon 6: Proceedings of the sixth meeting of the acl special interest group in computational phonology*, 58–69. ACL.

Anderson, Stephen R. 1974. *The Organization of Phonology*. Academic Press.

Barke, Shraddha; Rose Kunkel; Nadia Polikarpova; Eric Meinhardt; Eric Bakovic; and Leon Bergen. 2019. Constraint-based learning of phonological processes. *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)*,

References (*cont.*)

6176–6186. Hong Kong, China: Association for Computational Linguistics. Online: https://aclanthology.org/D19-1639.

Durvasula, Karthik, and Adam Liter. 2020. There is a simplicity bias when generalising from ambiguous data. *Phonology* 37.177–213.

Ellis, Kevin; Adam Albright; Armando Solar-Lezama; Joshua B. Tenenbaum; and Timothy J. O'Donnell. 2022. Synthesizing theories of human language with Bayesian program induction. *Nature Communications* 13.5024.

Gordon, Matthew J. 2002. A phonetically driven account of syllable weight. *Language* 78.51–80.

References (*cont.*)

Kager, René. 2012. Stress in windows: Language typology and factorial typology. *Lingua* 122.1454–1493.

Kiparsky, Paul. 1985. Some consequences of Lexical Phonology. *Phonology Yearbook* 2.85–138.

Moreton, Elliott, and Joe Pater. 2012. Structure and substance in artificial-phonology learning, part i: Structure. *Language and Linguistics Compass* 6.686–701.

Pater, Joe, and Elliott Moreton. 2012. Structurally biased phonology: Complexity in learning and typology. *The EFL Journal* 3.1–44.

Stanton, Juliet. 2016. Learnability shapes typology: the case of the midpoint pathology. *Language* 92.753–791.

References (*cont.*)

Tesar, Bruce, and Paul Smolensky. 2000. *Learnability in Optimality Theory*. Cambridge, MA: MIT Press.