The Bottom-Up Factor Inference Algorithm

Logan Swanson

OOKAMI
Stony Brook University
iACS INSTITUTE FOR ADVANCED COMPUTATIONAL SCIENCE

Outline

1. Representations and substructures for phonotactic learning
2. Structure of the constraint space
3. BUFIA: a phonotactic learning algorithm
4. Constraint selection strategies and their impact on learning

Phonotactic Grammars

Segmental

✔ papi
✔ tipa
✘ taap
✘ ptap

Grammar

*aa, *ai, *ii, *ia, *pp, *pt, *tp, *tt

Feature-Based

✔ papi
✔ tipa
✘ taap
✘ ptap

Grammar

*[+cons][-cons], *[-cons][+cons]

Autosegmental

```
✔ L H L H
   | |  | \
  σ σ σ σ
```

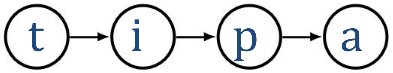
```
✘ L H L   H
   \ / /\   |
   σ σ σ σ
```

Grammar

```
*L H      *L
  \/      / \
  σ ,    σ σ
```

Phonotactic Grammars

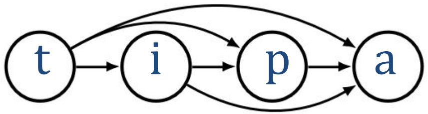
Successor



2-Factors

ti, ip, pa

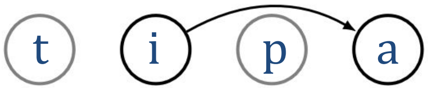
Precedence



2-Factors

ti, tp, ta, ip, ia, pa

Tier-Successor

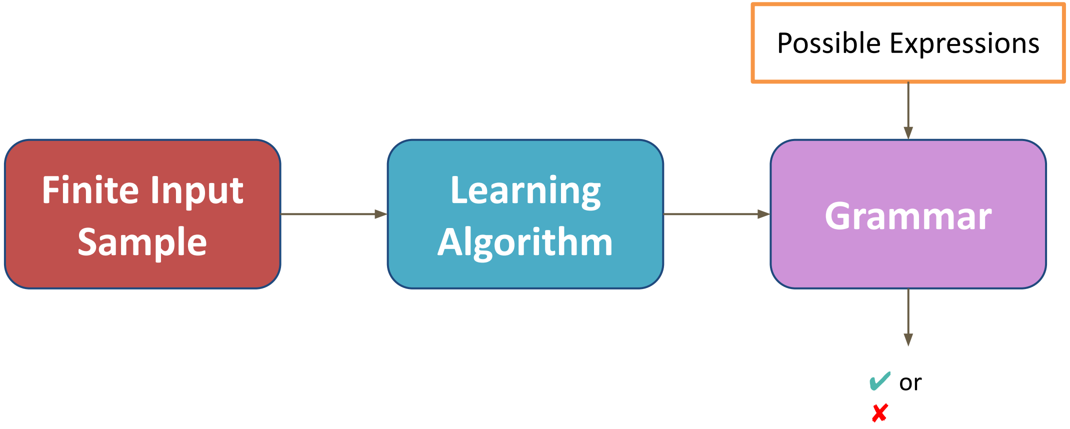


2-Factors

ia
Tier = Vowels

The Learning Problem

- Learning is about figuring out which substructures, or **factors**, belong in the grammar
- A learning algorithm is a function from input sample to grammar



Outline

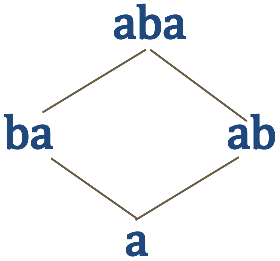
1. **Structure of the constraint space**
2. BUFIA: a phonotactic learning algorithm
3. Constraint selection strategies and their impact on learning
4. Case Study: Bolivian Quechua

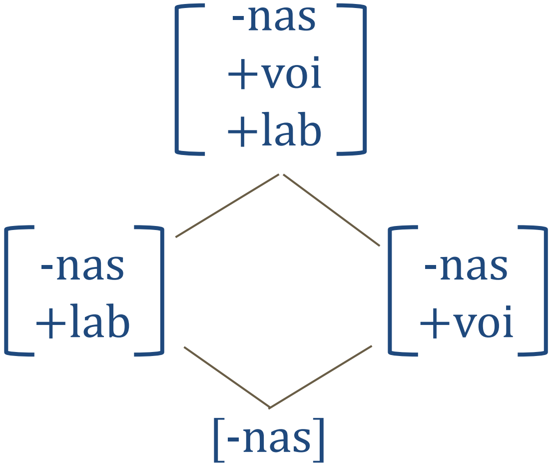
Phonotactic Grammars

- We can represent grammars as collections of substructures, or, **factors** which can together be used to determine whether a given form is licit or not
- Grammars can be **positive** or **negative**, and they can leverage many different kinds of substructures

Factor Entailments

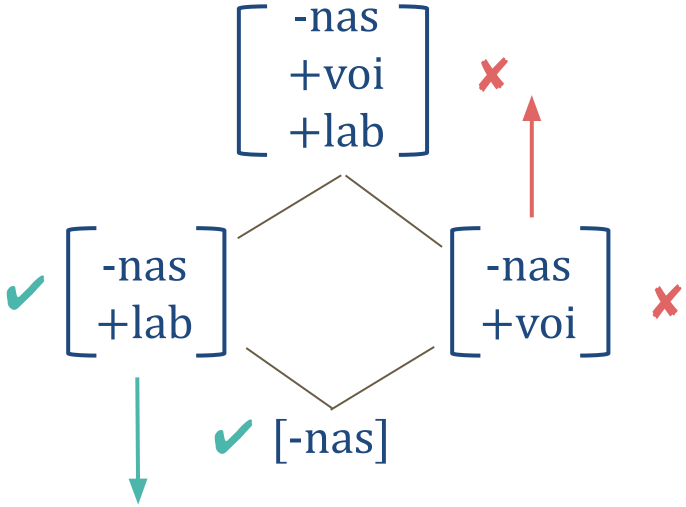
- Some factors contain others:





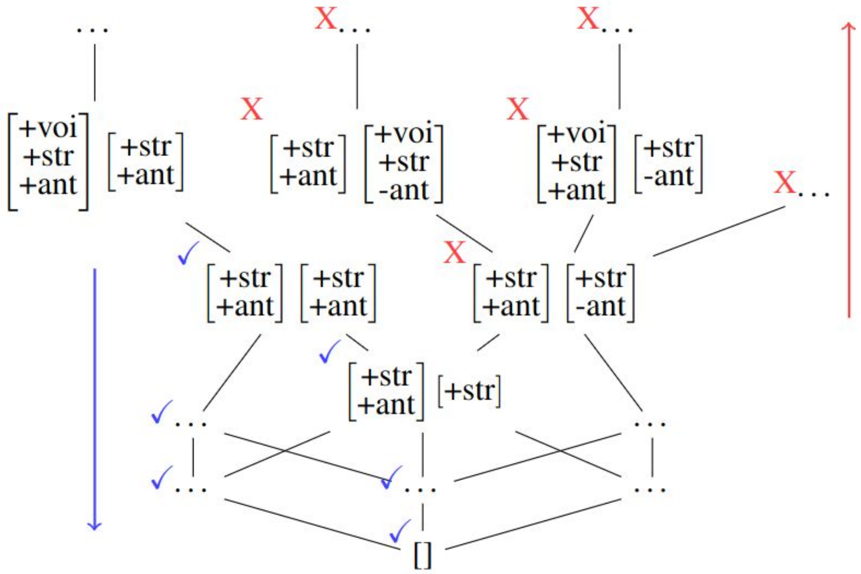
Factor Entailments

- If **S** is a subfactor of **T**, and **G** generates **T**, then **G** generates **S**
- If **T** is a superfactor of **S**, and **G** does *not* generate **S**, then **G** does not generate **T**



Factor Entailments

*sʃ



(Chandlee et al. 2019)

Leveraging Structure for Learning

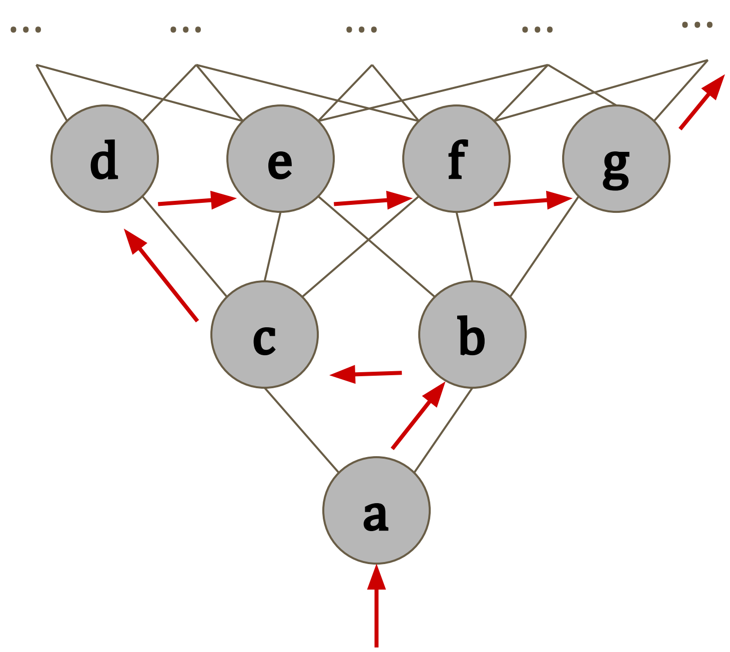
- These are large search spaces, but we have seen that they have a lot of internal structure
- How do we effectively leverage this structure for learning?
- The Bottom-Up Factor Inference Algorithm (Chandlee et al. 2019) does exactly this

Outline

1. Structure of the constraint space
2. **BUFIA: a phonotactic learning algorithm**
3. Constraint selection strategies and their impact on learning
4. Case Study: Bolivian Quechua

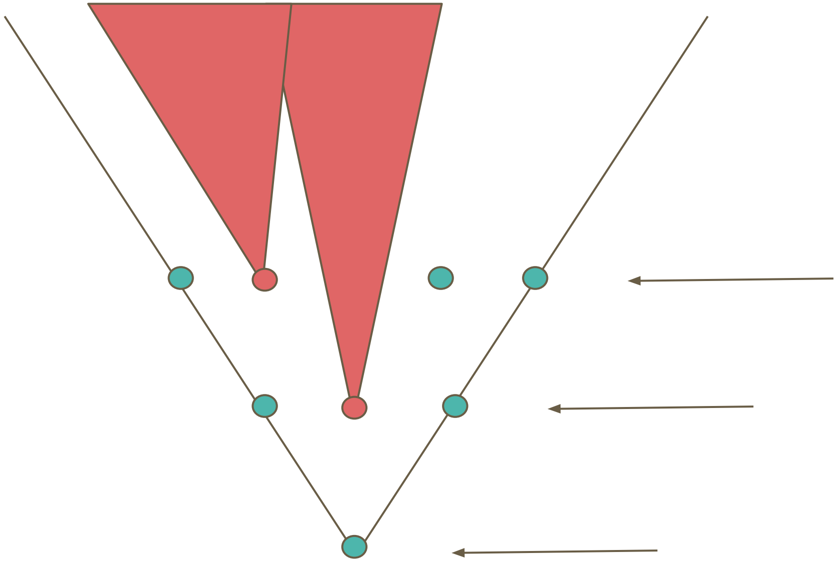
Bottom-Up Factor Inference Algorithm (BUFIA)

- Batch learner: all input data is provided upfront
- Start from the **bottom** of the partial order, and proceed upwards in a **breadth-first** manner
- Use the factor ordering relation to prune the hypothesis space along the way
    - This pruning crucially leverages the internal structure of the search space and sparsity in the input data to vastly reduce the area to be searched



BUFIA

- Start from the bottom of the partial order, and proceed upwards breadth-first
- For each factor:
  - **If it is present in the data, continue**
  - **If it is not present in the data, add it as a constraint and prune all its superfactors out of the search space**
- Stop when a cutoff condition is reached:
  - Typically this will be an upper bound on the size of the factors



Data: positive sample D, empty structure s_0, max constraint size k

Result: G, a set of constraints

$Q \leftarrow \{s_0\}$;

$G \leftarrow \emptyset$;

$V \leftarrow \emptyset$;

while $Q \neq \emptyset$ **do**

  $s \leftarrow Q$.dequeue();

  $V \leftarrow V \cup \{s\}$;

  if $\exists x \in D$ *such that* $s \sqsubseteq x$ **then**

    $S \leftarrow \texttt{NextSupFact}(s)$;

    $S' \leftarrow \{s \in S \mid |s| \leq k \wedge (\neg \exists g \in G)[g \sqsubseteq s] \wedge s \notin V\}$;

    Q.enqueue(S');

  end

  else

    $G \leftarrow G \cup \{s\}$;

  end

end

return G;

Properties of BUFIA

Given a finite positive data sample **D**, *BUFIA will find a grammar* **G** *of constraints for which the following are true:*

1. **G** is consistent with the data
2. **L(G)** is the smallest language in the relevant class which contains **D**
3. **G** includes the most general factors of any other grammars **G'** which satisfy both 1 and 2
   - ➢ No factor in any **G'** is more general than every factor in **G**

Demo: Parupa

Parupa is an artificial language created by Mayer (2020) consisting of a simple alphabet of 5 vowels (a, e, i, o, u) and 7 consonants (p, t, k, b, d, g, r) which follows some simple constraints:

Local Constraints:

- All syllables are CV
- Words must begin with /p/ or /b/
- /p t k/ must be followed by a high vowel or /a/
- /b d g/ must be followed by a mid vowel or /a/

Long-Distance Constraints:

- Words must contain only front or back vowels
- /a/ is transparent to harmony and may occur in either case

Outline

1. Structure of the constraint space
2. BUFIA: a phonotactic learning algorithm
3. **Constraint selection strategies and their impact on learning**
4. Case Study: Bolivian Quechua

Constraint Selection

- It is possible for many different sets of constraints to accomplish the same thing
- For example, if the sequence “*nt*” is absent from the data, there are many possible constraints that could account for this:

*[+nas][+cor]

*[+sonorant][-sonorant]

*[+consonant][-nas, -voi]

...

- How should the learner decide which constraints to select, when multiple hypotheses are empirically equivalent?

Abductive Principles

- **Abduction** refers to finding the *simplest* or *best* conclusion from a set of observations
- Abductive principles tell us which constraints to select, when multiple options may yield a result that is consistent with our data
- By virtue of proceeding bottom-up, BUFIA already employs one abductive principle–namely, that general constraints are better than specific ones
- However, without additional abductive principles in place, BUFIA will yield redundant grammars

Abductive Principles

- **Search Path:**
  - Feature ordering
  - What is considered a “layer”
- **Constraint Selection:**
  - Adds additional banned items
  - Adds exclusively novel banned items
  - Adds amount of novel banned items over some threshold
- Each of these choices has repercussions for learning behavior

Examples: Constraint Selection

```
[+cons][+cons]
[+cons][+wb]
[-cons][-cons]
[+wb][+son]
[+wb][-lab]
[+wb][+wb]
[-son,+voic][+high]
[-voic][-low,-high]
```

CV Syllables

Start with /p/ or /b/

/b d g/ must be followed by mid or /a/

/p t k/ must be followed by high or /a/

Parupa, only adding constraints which increase the number of banned ngrams

```
[+cons][+cons]
[+cons][-son]
[+cons][-lab]
[+cons][-dor]
[+cons][-voic]
[+cons][+lab]
[+cons][+dor]
[+cons][+wb]
[-son][+cons]
[-son][-son]
[-son][-lab]
[-son][-dor]
[-son][-voic]
[-son][+lab]
[-son][+dor]
```

...

121 total constraints

Parupa, adding all constraints

Examples: Feature Ordering

```
[+cons][+cons]
[+cons][+wb]
[-cons][-cons]
[+wb][+son]
[+wb][-lab]
[+wb][+wb]
[-son,+voic][+high]
[-voic][-low,-high]
```

Parupa, features ordered by extension size

CV Syllables

Start with /p/ or /b/

/b d g/ must be followed by mid or /a/

/p t k/ must be followed by high or /a/

```
[+back][+back]
[+back][-back]
[-back][+back]
[-back][-back]
[+cons][+cons]
[+cons][+wb]
[+wb][+back]
[+wb][-back]
[+wb][+son]
[+wb][+dor]
[+wb][-lab]
[+wb][+wb]
[-son,+voic][+high]
[-voic][+back,-high]
[-voic][-low,-high]
```

Parupa, features ordered by input feature matrix

Examples: Stopping Condition

```
[+cons][+cons]
[+cons][+wb]
[-cons][-cons]
[+wb][+son]
[+wb][-lab]
[+wb][+wb]
[-son,+voic][+high]
[-voic][-low,-high]
```

Parupa: max distance = 5

```
[+cons][+cons]
[+cons][+wb]
[-cons][-cons]
[+wb][+son]
[+wb][-lab]
[+wb][+wb]
[][+wb][]
[-son,+voic][+high]
[-voic][-low,-high]
[-back,-low][][+back]
[+back][][-back,-low]
[+son][+back,-high][+voic,+lab]
```

Parupa:
max factor width = 3,
max number of features = 3

```
[+cons][+cons]
[+cons][+wb]
[-cons][-cons]
[+wb][+son]
[+wb][-lab]
[+wb][+wb]
[][+wb][]
[-son,+voic][+high]
[-voic][-low,-high]
[-back,-low][][+back]
[+back][][-back,-low]
[+voic][+high][+son][+low]
[+son][+back,-high][+voic,+lab]
[+son][-back][+dor][+back]
[+dor][+high][+voic][+high]
[+high][+son][+low][+son]
[+back][+voic][+high][+son]
```

Parupa:
max distance = 8

Examples: Constraint Selection

Parupa: distance = 8

```
[+cons][+cons]
[+cons][+wb]
[-cons][-cons]
[+wb][+son]
[+wb][-lab]
[+wb][+wb]
[][+wb][]
[-son,+voic][+high]
[-voic][-low,-high]
[-back,-low][][+back]
[+back][][-back,-low]
```

New constraints must increase the number of banned ngrams by **10**

```
[+cons][+cons]
[+cons][+wb]
[-cons][-cons]
[+wb][+son]
[+wb][-lab]
[+wb][+wb]
[][+wb][]
[-son,+voic][+high]
[-voic][-low,-high]
[-back,-low][][+back]
[+back][][-back,-low]
[+voic][+high][+son][+low]
[+son][+back,-high][+voic,+lab]
[+son][-back][+dor][+back]
[+dor][+high][+voic][+high]
[+high][+son][+low][+son]
[+back][+voic][+high][+son]
```

New constraints must increase the number of banned ngrams by **1**

Outline

1. Structure of the constraint space
2. BUFIA: a phonotactic learning algorithm
3. Constraint selection strategies and their impact on learning
4. **Case Study: Bolivian Quechua**

Quechua Phonotactics

- /i u/ lower to [e o]:
  - Immediately **following** or **preceding** a uvular (/q q^h q'/)
  - Preceding a uvular **across an intervening coda**

(1) *Uvular contexts:* [e o] *[i u]

a.	**q'e**pij (*q'ipij)	'to carry'	**q'o**ɲi (*q'uɲi)	'hot'
b.	p**eq**aj (*piqaj)	'to grind'	n**oq**a (*nuqa)	'I'
c.	w**e**s**q'**aj (*wisq'aj)	'to close'	t**o**ʎ**q**a (*tuʎqa)	'son-in-law'

(2) *Elsewhere:* [i u] *[e o]

misi (*mese)	'cat'	kuʎku (*koʎko)	'type of bird'

Quechua Phonotactics

Table 1
Quechua tiers and phonotactic generalizations

Tier	Projected segments	Phonotactic generalizations
Dorsal	Dorsal consonants, vowels	High-mid vowel allophony
C-dorsal[a]	Dorsal consonants, +	*K . . . Q, *Q . . . K (within morphemes)
Laryngeal	Stops, affricates, h, ʔ	Laryngeal cooccurrence restrictions
Segmental	All, +	*VV, *CCC, *wu, *wo

[a] The C-dorsal tier contains a morpheme boundary symbol (+), allowing the model to represent the fact that the restriction on uvular and velar consonant cooccurrence holds within morphemes but not across them.

Accidental Gaps in Quechua (W & G)

- 2,966 legal trigrams, only 1,472 attested (49%)
- 32,971 illegal trigrams
- How should a learner distinguish principled gaps from accidental ones?
    - **[eq^{h}o]** vs **[k^{h}ek]**
- Wilson & Gallagher argue that two things are needed to do this:
    - Feature-based representations
    - **Statistical methods**

	Statistics	No Statistics
Segments	MaxEnt-Seg	(T)SL
Features	MaxEnt-Ftr	**???**

"What about a nonstatistical model that learns by memorizing feature sequences? ..."

	Statistics	No Statistics
Segments	MaxEnt-Seg	(T)SL
Features	MaxEnt-Ftr	**???**

*"Lacking a method for deciding **which** representations are relevant for assessing well-formedness – precisely the role played by statistics in Maxent-Ftr – **learning... is doomed**."*

– Wilson and Gallagher 2018

Experimental Setup

We followed the experimental setup used by W&G to add BUFIA to this paradigm:

- Training data: ~1,000 dictionary forms + 3 suffixes (with vowel lowering applied)
- Testing data: all possible CV(C)CV(C) sequences, sorted into
  - 150,000 “licit” forms
  - 400,000 “illicit” forms
  - Sorted according to the known phonotactic generalizations they describe

Experimental Setup

- Training data divided into 5 folds, with 20% of the dictionary forms held out in each group
- Testing data:
  - The held-out dictionary forms
  - The 150,000 synthetic licit forms
  - The 400,000 synthetic illicit forms
- We further divided this testing data into a tuning and eval set (random 50% of each category), and tuned our abductive parameters to optimize f1 over the tuning data
  - W&G also had parameter tuning, although it's not clear they had separate tuning and eval sets

Experiment 1 Results

	held-out forms (W&G)	legal nonce roots	illegal nonce roots
Features, Stats (MaxEnt-Ftr)	99.8%	82.2%	1.9%
Segments, Stats (MaxEnt-Seg)	99.7%	71.5%	45.4%
Segments, No-stats ((T)SL)	96.7%	18.8%	0.1%
Features, No-stats (BUFIA)	99.6%	94.1%	1.8%

Table 1: Experiment 1 Test Results: Percentage of forms accepted by evaluation category aggregated over the five folds. Results reported in rows 1-3 are from W&G.

Alternate Training Setup

- Possible issues with W&G experimental setup:
  - Roots are duplicated 4x in training set – but not controlled for in fold construction, so many roots will be present in both train and test sets
  - Uneven distribution of how many illicit forms violate each known constraint
    - In fact, a single tier can rule out 89% of illicit forms
  - Synthetic “licit” data is unverified by native speakers – the baked in assumption here is that the constraints identified by W&G are the only ones active in the grammar

Alternate Training Setup

- New 5-fold split of dataset with no roots duplicated across train and test sets
- New set of illicit data, consisting of 40 forms which uniquely violate each known constraint
- No synthetic “licit” data

Experiment 2 Results

	precision	recall	f1
MaxEnt-Ftr	0.786	0.951	0.861
BUFIA	0.946	0.943	0.945

Table 2: Experiment 2 Test Results: Scores are aggregated over the five folds.

Takeaways

- The substructures relevant for phonotactic learning form a **structured search space**, and this structure can be leveraged for learning
- **BUFIA** is a deterministic, non-statistical batch learner which leverages this structure to learn surface-true constraints over positive data
- BUFIA is competitive on phonotactic learning tasks with natural language data
- Choices about **abductive principles** are highly relevant to learning behavior
- BUFIA can be thought of as a **general-form phonotactic learner**, which can operate with different representations and abductive principles and lend insight into how these impact learning

Thanks!

Example: Malarky

[-approx][-voice]
[-sonorant][-sonorant]
[+vocalic][+vocalic]
[][+wb][]
[-vocalic][-vocalic][-vocalic]
[-vocalic][-vocalic][+wb]
[+wb][-vocalic][-vocalic]

*NT

No adjacent obstruents

No adjacent vowels

No complex onsets or codas

Malarky: a=100

[-approx][-voice]
[-sonorant][-sonorant]
[+vocalic][+vocalic]
[+wb][+wb]
[][+wb][]
[+dorsal][-anterior,-delrel]
[+wb][][+wb]
[-vocalic][-vocalic][-vocalic]
[-vocalic][-vocalic][+wb]
[+round][+dorsal][-delrel]
[+round][+rhotic][+rhotic]
[+round][+rhotic][+lateral]
[+back][+dorsal][-delrel]
[+back][+rhotic][+rhotic]
[+high][+dorsal][-delrel]
[+low][+rhotic][+rhotic]
[+low][+lateral][+dorsal]
[+wb][-vocalic][-vocalic]
[+rhotic][+rhotic][+back]
[+lateral][+lateral][+low]

Malarky: a=1

Layers by k vs d

```
[+cons][+cons]
[-cons][-cons]
[-son,+voic][+high]
[-voic][-low,-high]
[-back,-low][][+back]
[+back][][-back,-low]
```

Parupa, layered by distance, d=5

```
[+cons][+cons]
[-cons][-cons]
[-voic][-low,-high]
[+voic,-son][+high]
[+son][-high,+back][+voic,+lab]
[+back][][-low,-back]
[-low,-back][][+back]
```

Parupa, layered by factor width (k), k=3

Non-intersecting extension condition

```
[+cons][+cons]
[-cons][-cons]
[-voic][-low,-high]
[+voic,-son][+high]
[+son][-high,+back][+voic,+lab]
[+back][][-low,-back]
[-low,-back][][+back]
```

Parupa, new constraints must add **some** new ngram

```
[+cons][+cons]
[-cons][-cons]
[-cons][-voic][-low,-high]
[-cons][+voic,-son][+high]
[-voic][-low,-high][+cons]
[+back][-voic][-back,+high]
[+back][+cons,+son][-low,-back]
[+voic,-son][+high][+cons]
[+cons,+son][-high,+back][+voic,+lab]
[-low,-back][-voic][+back,+high]
[-low,-back][+voic,+cons][-high,+back]
[-low,-back][+cons,+son][+back,+high]
```

Parupa, new constraints must add **exclusively** new ngrams

What about MaxEnt?

- The grammar produced by MaxEnt is *gradient* rather than *categorical*, but fundamentally the MaxEnt learner is traversing the same kind of structured constraint space, just with different abductive principles about ordering and constraint selection
- Ratio of Observed:Expected occurrences of forms is used to choose which constraint to add next, and at every step constraints are re-weighted to maximize the likelihood of the observed data
- Although not explicitly discussed that way, rules about factor entailments are still highly relevant: if a factor is added to the grammar, the expected occurrences of all its superfactors will be lowered commensurately with the weight assigned to that constraint

MaxEnt

[+sonorant,+dorsal]
[+sonorant][]
[][+continuant]
[][+voice]
[][+strident]
[][-back]
[-continuant,+strident][]
[-continuant][-approximant]
[+continuant,+voice,-anterior]
[+continuant,+voice][]
[-strident][-approximant]
[-strident][+consonantal]
[-anterior][-approximant]
[+labial][-approximant]
[+labial][+labial]
[+spread][-approximant]
[][+coronal][+labial]

Constraint	Weight	Comment
1. *[+son, +dors]	5.64	*[ŋ]
2. *[+cont, +voice, −ant]	3.28	*[ʒ]
3. *[^ −voice, +ant, +strid][−approx]	5.91	Nasals and obstruents may only be preceded (within the onset) by [s].
4. *[ ][+cont]	5.17	Fricatives may not cluster with preceding C.
5. *[ ][+voice]	5.37	Voiced obstruents may not cluster with preceding C.
6. *[+son][ ]	6.66	Sonorants may only be onset-final.
7. *[−strid][+cons]	4.40	Nonstrident coronals may not precede nonglides.
8. *[ ][+strid]	1.31	Stridents must be initial in a cluster.
9. *[+lab][^ +approx, +cor]	4.96	The only consonants that may follow labials are [l] and [r].
10. *[−ant][^ +approx, −ant]	4.84	Only [r] may follow nonanterior coronals.
11. *[+cont, +voice][ ]	4.84	Voiced fricatives must be final in an onset.
12. *[−cont, −ant][ ]	3.17	[t͡ʃ] and [d͡ʒ] must be final in an onset.
13. *[ ][−back]	5.04	[j] may not cluster with a preceding C; see above for assumed syllabic parsing of [ju].

MaxEnt

```
[+sonorant,+dorsal]
[+sonorant][]
[][+continuant]
[][+voice]
[][+strident]
[][-back]
[-continuant,+strident][]
[-continuant][-approximant]
[+continuant,+voice,-anterior]
[+continuant,+voice][]
[-strident][-approximant]
[-strident][+consonantal]
[-anterior][-approximant]
[+labial][-approximant]
[+labial][+labial]
[+spread][-approximant]
[][+coronal][+labial]
```

Constraint	Weight	Comment
1. *[+son, +dors]	5.64	*[ŋ]
2. *[+cont, +voice, −ant]	3.28	*[ʒ]
3. *[^−voice, +ant, +strid][−approx]	5.91	Nasals and obstruents may only be preceded (within the onset) by [s].
4. *[ ][+cont]	5.17	Fricatives may not cluster with preceding C.
5. *[ ][+voice]	5.37	Voiced obstruents may not cluster with preceding C.
6. *[+son][ ]	6.66	Sonorants may only be onset-final.
7. *[−strid][+cons]	4.40	Nonstrident coronals may not precede nonglides.
8. *[ ][+strid]	1.31	Stridents must be initial in a cluster.
9. *[+lab][^+approx, +cor]	4.96	The only consonants that may follow labials are [l] and [r].
10. *[−ant][^+approx, −ant]	4.84	Only [r] may follow nonanterior coronals.
11. *[+cont, +voice][ ]	4.84	Voiced fricatives must be final in an onset.
12. *[−cont, −ant][ ]	3.17	[t͡ʃ] and [d͡ʒ] must be final in an onset.
13. *[ ][−back]	5.04	[j] may not cluster with a preceding C; see above for assumed syllabic parsing of [ju].

MaxEnt

```
[+sonorant,+dorsal]
[+sonorant][]
[][+continuant]
[][+voice]
[][+strident]
[][-back]
[-continuant,+strident][]
[-continuant][-approximant]
[+continuant,+voice,-anterior]
[+continuant,+voice][]
[-strident][-approximant]
[-strident][+consonantal]
[-anterior][-approximant]
[+labial][-approximant]
[+labial][+labial]
[+spread][-approximant]
[][+coronal][+labial]
```

14.	*[+ant, +strid][−ant]	2.80	Sibilants must agree in anteriority with a following [−anterior] consonant.
15.	*[+spread][^+back]	4.82	[h] may only cluster with [w] (dialect assumed has [hw] as legal).
16.	*[+cont, +voice, +cor]	2.69	Disprefer voiced coronal fricatives (violable).
17.	*[+voice][^+approx, +cor]	2.97	Voiced obstruents may only be followed by [l, r] (violable).
18.	*[+cont, −strid][^+approx, −ant]	2.06	[θ, ð] may only be followed by [r] (violable).
19.	*[ ][^−cont, −voice, +lab][+cons]	3.05	In effect: only [p], and not [k], may occur / s___l (violable).
20.	*[ ][+cor][^+approx, −ant]	2.06	In effect: only [r] may occur after [st].
21.	*[+cont, −strid]	1.84	[θ, ð] are rare (violable).
22.	*[+strid][−ant]	2.10	In effect: [ʃr] is rare (violable).
23.	*[−cont, −voice, +cor][^+approx, −ant]	1.70	In effect: [t] can only be followed by [r] (violable).

MaxEnt: English Onset Features

	p	t	tʃ	k	b	d	dʒ	g	f	θ	s	ʃ	h	v	ð	z	ʒ	m	n	ŋ	l	r	j	w
cons	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	−	−	−
approx	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	+	+	+
son	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	+	+	+	+	+	+
cont	−	−	−	−	−	−	−	−	+	+	+	+	+	+	+	+	+							
nas																		+	+	+				
voice	−	−	−	−	+	+	+	+	−	−	−	−	−	+	+	+	+							
spread													+											
lab	+				+				+					+				+						+
cor		+	+			+	+			+	+	+			+	+	+		+		+	+		
ant		+	−			+	−			+	+	−			+	+	−		+		+	−		
strid		−	+			−	+			−	+	+			−	+	+		−		−	−		
lat																					+			
dors				+				+												+				
high																							+	+
back																							−	+

References

- Jane Chandlee, Rémi Eyraud, Jeffrey Heinz, Adam Jardine, and Jonathan Rawski. 2019. “Learning with Partially Ordered Representations”. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 91--101. Association for Computational Linguistics.
- Connor Mayer. An algorithm for learning phonological classes from distributional similarity. Phonology 37 (2020) 91-131. doi:10.1017/S0952675720000056
- Hayes, Bruce, and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. Linguistic Inquiry 39:379–440.