Grammatical inference and subregular phonology

Adam Jardine
Rutgers University

December 10, 2019 · Tel Aviv University

Review

Outline of course

- **Day 1:** Learning, languages, and grammars
- **Day 2:** Learning strictly local grammars
- **Day 3:** Automata and input strictly local functions
- **Day 4:** Learning functions and stochastic patterns, other open questions

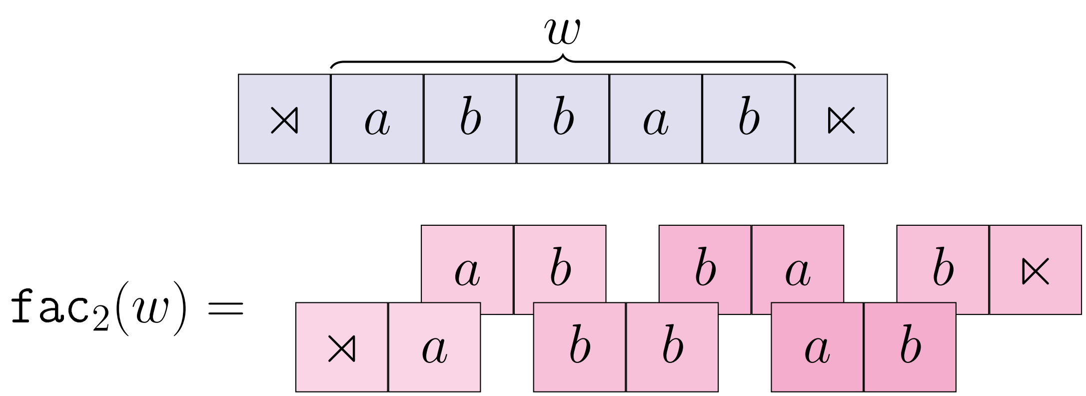
Review of day 1

- Phonological patterns are governed by restrictive computational universals
- **Grammatical inference** connects these universals to solutions to the **learning problem**:

> **Problem**
>
> Given a **positive** sample of a language, return a grammar that describes that language **exactly**

Review of day 1

- **Strictly local languages** are patterns computed solely by k**-factors** in a string

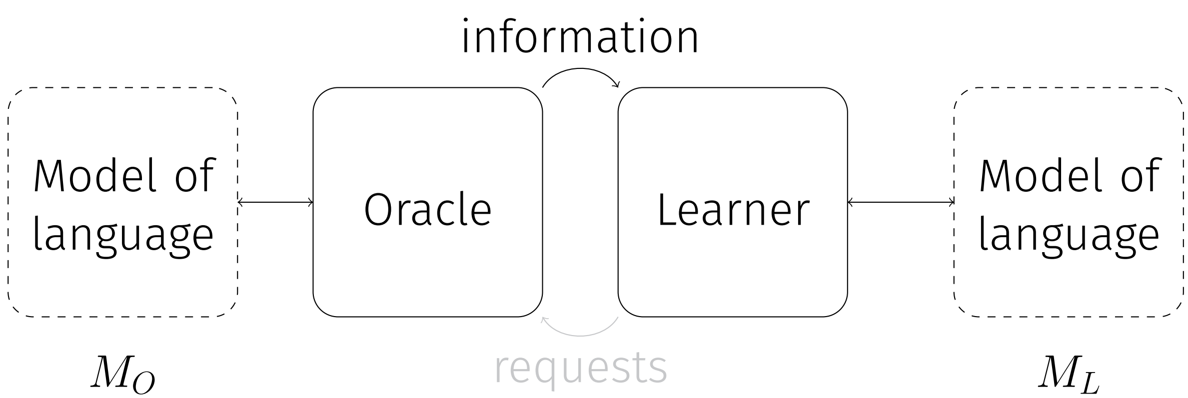


Today

- A provably correct method for learning SL$_k$ languages
- The paradigm of **identification in the limit from positive data** (Gold, 1967; de la Higuera, 2010)
- Why learners target **classes** (not specific languages, or all possible languages)

Learning paradigm

Learning paradigm



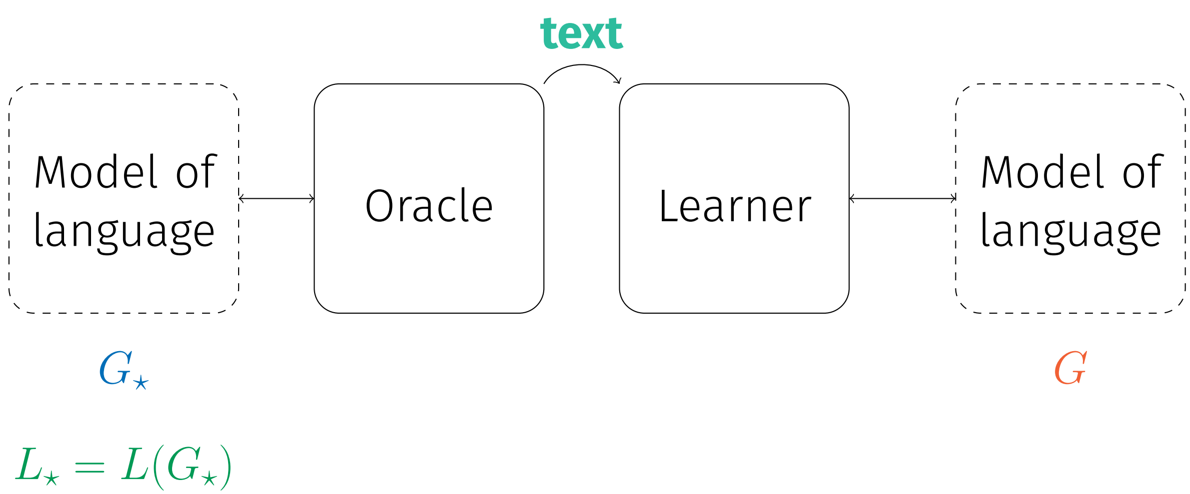
(from Heinz et al., 2016)

Problem

Given a **positive** sample of a language, return a grammar that describes that language **exactly**

- This is **(exact) identification in the limit from positive data** (**ILPD;** Gold, 1967)

Identification in the limit from positive data (ILPD)

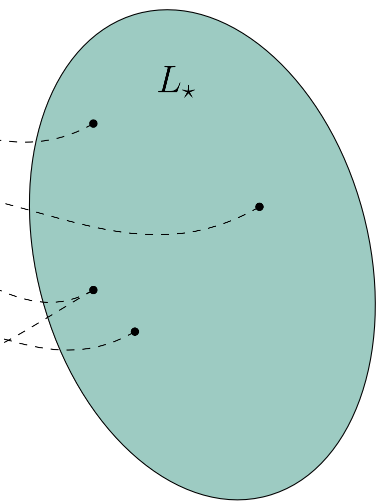


- A **text** of $L_\star$ is some sample of positive examples of $L_\star$

Identification in the limit from positive data (ILPD)

A **presentation** of $L_\star$ is a *sequence* p of examples drawn from $L_\star$

t	$p(t)$
0	$abab$
1	$ababab$
2	ab
3	λ
4	ab
$\vdots$	$\vdots$



(this is the ‘in the limit’ part)

Identification in the limit from positive data (ILPD)

A learner $\mathcal{A}$ takes a finite sequence and outputs a grammar

t	$p(t)$
0	$abab$
1	$ababab$
2	ab
3	λ
4	ab
$\vdots$	$\vdots$
n	$ababab$
$\vdots$	$\vdots$

$$p[i] \longrightarrow \mathcal{A} \longrightarrow G_i$$

Identification in the limit from positive data (ILPD)

Let’s take the learner $\mathcal{A}_{\mathrm{Fin}}$:

$$\mathcal{A}_{\mathrm{Fin}}(p[n]) = \{w \mid w = p(i) \text{ for some } i \leq n\}$$

Identification in the limit from positive data (ILPD)

Let's take the learner $\mathcal{A}_{\mathrm{Fin}}$:

$$\mathcal{A}_{\mathrm{Fin}}(p[n]) = \{w \mid w = p(i) \text{ for some } i \leq n\}$$

Let's set $L_\star = \{ab, bab, aaa\}$

Identification in the limit from positive data (ILPD)

Let's take the learner $\mathcal{A}_{\mathrm{Fin}}$:

$$\mathcal{A}_{\mathrm{Fin}}(p[n]) = \{w \mid w = p(i) \text{ for some } i \leq n\}$$

Let's set $L_\star = \{ab, bab, aaa\}$

t	$p(t)$	G_t
0	bab	

Identification in the limit from positive data (ILPD)

Let's take the learner $\mathcal{A}_{\mathrm{Fin}}$:

$$\mathcal{A}_{\mathrm{Fin}}(p[n]) = \{w \mid w = p(i) \text{ for some } i \leq n\}$$

Let's set $L_\star = \{ab, bab, aaa\}$

t	$p(t)$	G_t
0	bab	$\{bab\}$

Identification in the limit from positive data (ILPD)

Let's take the learner $\mathcal{A}_{\mathrm{Fin}}$:

$$\mathcal{A}_{\mathrm{Fin}}(p[n]) = \{w \mid w = p(i) \text{ for some } i \leq n\}$$

Let's set $L_\star = \{ab, bab, aaa\}$

t	$p(t)$	G_t
0	bab	$\{bab\}$
1	ab	

Identification in the limit from positive data (ILPD)

Let's take the learner $\mathcal{A}_{\text{Fin}}$:

$$\mathcal{A}_{\text{Fin}}(p[n]) = \{w \mid w = p(i) \text{ for some } i \leq n\}$$

Let's set $L_\star = \{ab, bab, aaa\}$

t	$p(t)$	G_t
0	bab	$\{bab\}$
1	ab	$\{ab, bab\}$

Identification in the limit from positive data (ILPD)

Let's take the learner $\mathcal{A}_{\text{Fin}}$:

$$\mathcal{A}_{\text{Fin}}(p[n]) = \{w \mid w = p(i) \text{ for some } i \leq n\}$$

Let's set $L_\star = \{ab, bab, aaa\}$

t	$p(t)$	G_t
0	bab	$\{bab\}$
1	ab	$\{ab, bab\}$
2	bab	

Identification in the limit from positive data (ILPD)

Let's take the learner $\mathcal{A}_{\text{Fin}}$:

$$\mathcal{A}_{\text{Fin}}(p[n]) = \{w \mid w = p(i) \text{ for some } i \leq n\}$$

Let's set $L_\star = \{ab, bab, aaa\}$

t	$p(t)$	G_t
0	bab	$\{bab\}$
1	ab	$\{ab, bab\}$
2	bab	$\{ab, bab\}$

Identification in the limit from positive data (ILPD)

Let's take the learner $\mathcal{A}_{\mathrm{Fin}}$:

$$\mathcal{A}_{\mathrm{Fin}}(p[n]) = \{w \mid w = p(i) \text{ for some } i \leq n\}$$

Let's set $L_\star = \{ab, bab, aaa\}$

t	$p(t)$	G_t
0	bab	$\{bab\}$
1	ab	$\{ab, bab\}$
2	bab	$\{ab, bab\}$
3	aaa	$\{ab, bab, aaa\}$

Identification in the limit from positive data (ILPD)

Let's take the learner $\mathcal{A}_{\mathrm{Fin}}$:

$$\mathcal{A}_{\mathrm{Fin}}(p[n]) = \{w \mid w = p(i) \text{ for some } i \leq n\}$$

Let's set $L_\star = \{ab, bab, aaa\}$

t	$p(t)$	G_t
0	bab	$\{bab\}$
1	ab	$\{ab, bab\}$
2	bab	$\{ab, bab\}$
3	aaa	$\{ab, bab, aaa\}$
4	ab	$\{ab, bab, aaa\}$

Identification in the limit from positive data (ILPD)

Let's take the learner $\mathcal{A}_{\mathrm{Fin}}$:

$$\mathcal{A}_{\mathrm{Fin}}(p[n]) = \{w \mid w = p(i) \text{ for some } i \leq n\}$$

Let's set $L_\star = \{ab, bab, aaa\}$

t	$p(t)$	G_t
0	bab	$\{bab\}$
1	ab	$\{ab, bab\}$
2	bab	$\{ab, bab\}$
3	aaa	$\{ab, bab, aaa\}$
4	ab	$\{ab, bab, aaa\}$
	$\ldots$	
308	bab	$\{ab, bab, aaa\}$

Identification in the limit from positive data (ILPD)

$\mathcal{A}$ **converges** at point n if $G_m = G_n$ for *any* $m > n$

t	$p(t)$	G_t
0	$abab$	G_0
1	$ababab$	G_1
2	ab	G_2
$\vdots$	$\vdots$	$\vdots$
n	$ababab$	G_n
$n+1$	$abababab$	G_n
$\vdots$	$\vdots$	$\vdots$
m	λ	G_n
$\vdots$	$\vdots$	$\vdots$

convergence

Identification in the limit from positive data (ILPD)

ILPD-learnability

A class $\mathcal{C}$ is **ILPD-learnable** if there is some algorithm $\mathcal{A}_{\mathcal{C}}$ such that for *any* stringset $L \in \mathcal{C}$, given *any* positive presentation p of L, $\mathcal{A}_{\mathcal{C}}$ converges to a grammar G such that $L(G) = L$.

- How is ILPD learning an idealization?

- What are the advantages of using ILPD as a criterion for learning?

Learning strictly local languages

Learning SL languages

- Given any k, the class SL_k is IDLP-learnable.
- Using $\mathcal{A}_{\mathrm{Fin}}$ as an example, how might we learn a SL_k language?

Learning SL languages

$$G_\star = \{CC, C\ltimes\}$$

t	datum	hypothesis
0	V	
1	$CVCV$	
2	$CVVCVCV$	
3	$VCVCV$	
$\vdots$		

Learning SL languages

$$G_\star = \{CC, C\ltimes\}$$

t	datum	hypothesis
0	V	$\{\rtimes C, \rtimes V, CC, CV, C\ltimes, VC, VV, V\ltimes\}$
1	$CVCV$	
2	$CVVCVCV$	
3	$VCVCV$	

$\vdots$

Learning SL languages

$$G_\star = \{CC, C\ltimes\}$$

t	datum	hypothesis
0	V	$\{\rtimes C, \rtimes V, CC, CV, C\ltimes, VC, VV, V\ltimes\}$
1	$CVCV$	$\{\rtimes C, \rtimes V, CC, CV, C\ltimes, VC, VV, V\ltimes\}$
2	$CVVCVCV$	
3	$VCVCV$	
$\vdots$		

Learning SL languages

$$G_\star = \{CC, C\ltimes\}$$

t	datum	hypothesis
0	V	$\{\rtimes C, \rtimes V, CC, CV, C\ltimes, VC, VV, V\ltimes\}$
1	$CVCV$	$\{\rtimes C, \rtimes V, CC, CV, C\ltimes, VC, VV, V\ltimes\}$
2	$CVVCVCV$	$\{\rtimes C, \rtimes V, CC, CV, C\ltimes, VC, VV, V\ltimes\}$
3	$VCVCV$	
$\vdots$		

Learning SL languages

$$G_\star = \{CC, C\ltimes\}$$

t	datum	hypothesis
0	V	$\{\rtimes C, \rtimes V, CC, CV, C\ltimes, VC, VV, V\ltimes\}$
1	$CVCV$	$\{\rtimes C, \rtimes V, CC, CV, C\ltimes, VC, VV, V\ltimes\}$
2	$CVVCVCV$	$\{\rtimes C, \rtimes V, CC, CV, C\ltimes, VC, VV, V\ltimes\}$
3	$VCVCV$	$\{\rtimes C, \rtimes V, CC, CV, C\ltimes, VC, VV, V\ltimes\}$
$\vdots$		

Learning SL languages

$$\mathcal{A}_{\mathrm{SL}_k}(p[i]) = \texttt{fac}_k(\Sigma^*) - \texttt{fac}_k\{p(0), p(1), ..., p(i)\}$$

Learning SL languages

$$\mathcal{A}_{\mathrm{SL}_k}(p[i]) = \texttt{fac}_k(\Sigma^*) - \texttt{fac}_k\{p(0), p(1), ..., p(i)\}$$

- The **characteristic sample** is ...

Learning SL languages

$$\mathcal{A}_{\mathrm{SL}_k}(p[i]) = \texttt{fac}_k(\Sigma^*) - \texttt{fac}_k\{p(0), p(1), ..., p(i)\}$$

- The **characteristic sample** is $\texttt{fac}_k(L_\star)$

Learning SL languages

$$\mathcal{A}_{\mathrm{SL}_k}(p[i]) = \mathtt{fac}_k(\Sigma^*) - \mathtt{fac}_k\{p(0), p(1), ..., p(i)\}$$

- The **characteristic sample** is $\mathtt{fac}_k(L_\star)$
- The **time complexity** is **linear**—the time it takes to calculate is directly proportional to the size of the data sample.

Learning SL languages

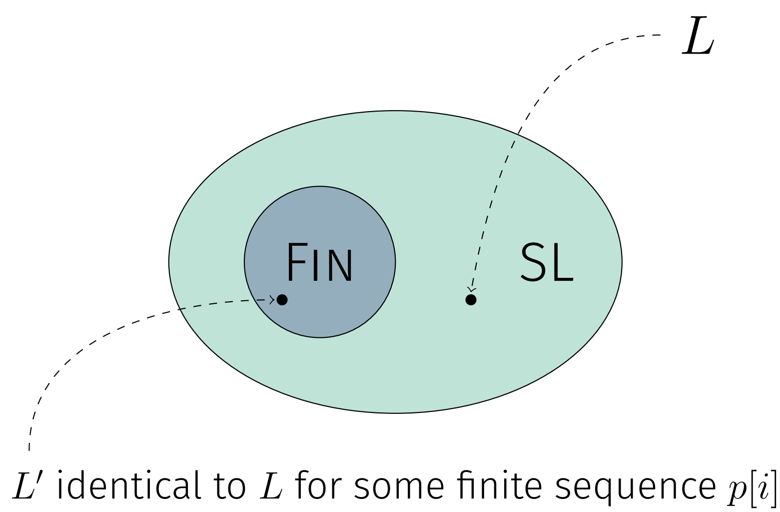
Let's learn Pintupi. Note that $k = 3$. What is the initial hypothesis? At what point do we converge?

t	datum	hypothesis
0	$\acute{\sigma}$	
1	$\acute{\sigma}\sigma$	
2	$\acute{\sigma}\sigma\sigma$	
3	$\acute{\sigma}\sigma\acute{\sigma}\sigma$	
4	$\acute{\sigma}\sigma\acute{\sigma}\sigma\sigma$	
5	$\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma$	
$\vdots$		

The limits of SL learning

The limits of SL learning

- We must know k in advance



- Gold (1967): any class $\mathcal{C}$ such that $\mathrm{Fin} \subsetneq \mathcal{C}$ is not learnable from positive examples

The limits of SL learning

- Consider this pattern from Inseño Chumash:

ʃ-api-tʃʰol-it	‘I have a stroke of good luck’
s-api-tsʰol-us	‘he has a stroke of good luck’
ʃ-api-tʃʰol-uʃ-waʃ	‘he had a stroke of good luck’
ha-ʃxintila-waʃ	‘his former Indian name’
s-is-tisi-jep-us	‘they (two) show him’
k-ʃu-ʃojin	‘I darken it’

- What phonotactic constraints are active here?

The limits of SL learning

- Consider this pattern from Inseño Chumash:

ʃ-api-tʃ^{h}ol-it	'I have a stroke of good luck'
s-api-ts^{h}ol-us	'he has a stroke of good luck'
ʃ-api-tʃ^{h}ol-uʃ-waʃ	'he had a stroke of good luck'
ha-ʃxintila-waʃ	'his former Indian name'
s-is-tisi-jep-us	'they (two) show him'
k-ʃu-ʃojin	'I darken it'

- What phonotactic constraints are active here?

*ʃ…s, *s…ʃ

The limits of SL learning

- Let's assume $L_\star = L_C$ for $\Sigma = \{\text{s,o,t,ʃ}\}$ as given below

$$L_C = \{\text{so, ss, ..., sos, ʃoʃ, ʃoʃoʃ, sosos, ʃototoʃ, sototos, ...}\}$$

t	datum	hypothesis
0	sos	{ss, so, sʃ, ..., ʃs, ʃt, ʃʃ}
1	sotoss	{ss, so, sʃ, ..., ʃs, ʃt, ʃʃ}
2	ʃoʃtoʃʃ	{ss, so, sʃ, ..., ʃs, ʃt, ʃʃ}
⋮		

The limits of SL learning

- Let's assume $L_\star = L_C$ for $\Sigma = \{\text{s,o,t,ʃ}\}$ as given below

$$L_C = \{\text{so, ss, ..., sos, ʃoʃ, ʃoʃoʃ, sosos, ʃototoʃ, sototos, ...}\}$$

t	datum	hypothesis
0	sos	{ss, so, sʃ, ..., ʃs, ʃt, ʃʃ}
1	sotoss	{ss, so, sʃ, ..., ʃs, ʃt, ʃʃ}
2	ʃoʃtoʃʃ	{ss, so, sʃ, ..., ʃs, ʃt, ʃʃ}
⋮		

- Learner will never see sʃ or ʃs, so in the limit $G = \{\text{sʃ,ʃs}\}$.

The limits of SL learning

$$L_C = \{\text{so, ss, ..., sos, ʃoʃ, ʃoʃoʃ, sosos, ʃototoʃ, sototos, ...}\}$$

$$G = ?\{\text{sʃ,ʃs}\}$$

✓ sosos $\in L_C$
✓ soʃs $\notin L_C$
✗ soʃos $\notin L_C$

The limits of SL learning

$$L_C = \{\text{so, ss, ..., sos, ʃoʃ, ʃoʃoʃ, sosos, ʃototoʃ, sototos, ...}\}$$

$$G_{k=3} = ?\{\text{soʃ, ssʃ, stʃ, sʃʃ, ... ʃos, ʃss, ʃts, ʃʃs}\}$$

✓ sosos $\in L_C$
✓ soʃs $\notin L_C$
✓ soʃos $\notin L_C$
✗ ʃotos $\notin L_C$

The limits of SL learning

$$L_C = \{\text{so, ss, ..., sos, ʃoʃ, ʃoʃoʃ, sosos, ʃototoʃ, sototos, ...}\}$$

- There is no k such that $\mathcal{A}_{\mathsf{SL}_k}$ learns a grammar for L_C
- This is because there is **no SL grammar** for L_C!

The limits of SL learning

- $\mathcal{A}_{\mathrm{SL}_k}$ **only learns SL$_k$ languages**
- This is the advantage of studying learning with formal grammatical inference:
  - we what patterns it **can learn**,
  - what patterns it **cannot learn**,
  - on **exactly what data**

Review

- As a hypothesis of phonotactic learning, $\mathcal{A}_{\mathrm{SL}_k}$
  - makes **restrictive** predictions about what patterns can and cannot be learned
  - suggests phonological learning is **modular** (Heinz, 2010)
  - directly connects computational typological generalizations with a theory of learning

Review

Problem

Given a **positive** sample of a language, return a grammar that describes that language **exactly**

- We have formalized this problem as **identification in the limit from positive data**
- We have solved this problem for any SL_k class
- We'll find another solution with automata, and extend that to learn processes