No free lunch: Why computational learning theory matters for language acquisition

Adam Jardine
Rutgers University

April 21, 2023 · Tokyo University/Computational Psycholinguistics Tokyo

Basic questions

How do children

- acquire language...
- without explicit instruction...
- in such a uniform way...
- despite the variety of experience?

"[V]arious formal and substantive universals are intrinsic properties of the language-acquisition system, these providing a schema that is applied to data and that determines in a highly restricted way the general form and, in part, even the substantive features of the grammar that may emerge upon presentation of appropriate data."

(Chomsky, 1965)

"It made sense for researchers to explore the possibility of a universal grammar at the time it was proposed (Chomksy 1965), when an understanding of the power of statistical learning and induction were a long way off."

Goldberg (2009, p. 203)

Theoretical learning results refute Goldberg's claim:

- Gold (1967): No restrictions on data presentation $\implies$ no general learning algorithm from positive data
- Angluin (1988): "[T]he assumption of stochastically generated examples does not enlarge the class of learnable sets of languages." (p. 2)
- Wolpert and Macready (1997): "[I]f an algorithm performs well on a certain class of problems then it necessarily pays for that with degraded performance on the set of all remaining problems." (p. 67)

- A successful (language) learner **must assume** a restriction…
  - … on the possibilities it is willing to consider; or
  - … on how the data is being presented to it

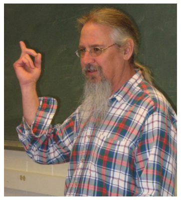
- Computational learning theory is a framework for...
  - clearly stating learning problems
  - ...and solutions!
  - developing restrictive, testable hypotheses about language learning

This talk:

- Basic results in comp. learning theory, starting from Gold (1967)
- Criticisms, extensions, alternatives
- Implications for theoretical linguistics, language acquisition
- Illustrations with applications/results in phonology (but transferable to syntax!)
- Further reading

- **Collaborators/Mentors**:

Jeff Heinz
(Stony Brook)



Jim Rogers
(Earlham)

Rémi Eyraud
(Jean Monnet)

Jane Chandlee
(Haverford)

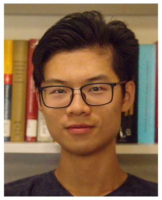
Kevin McMullin
(Ottowa)

...at Rutgers:

Tatevik Yolyan

Dine Mamadou

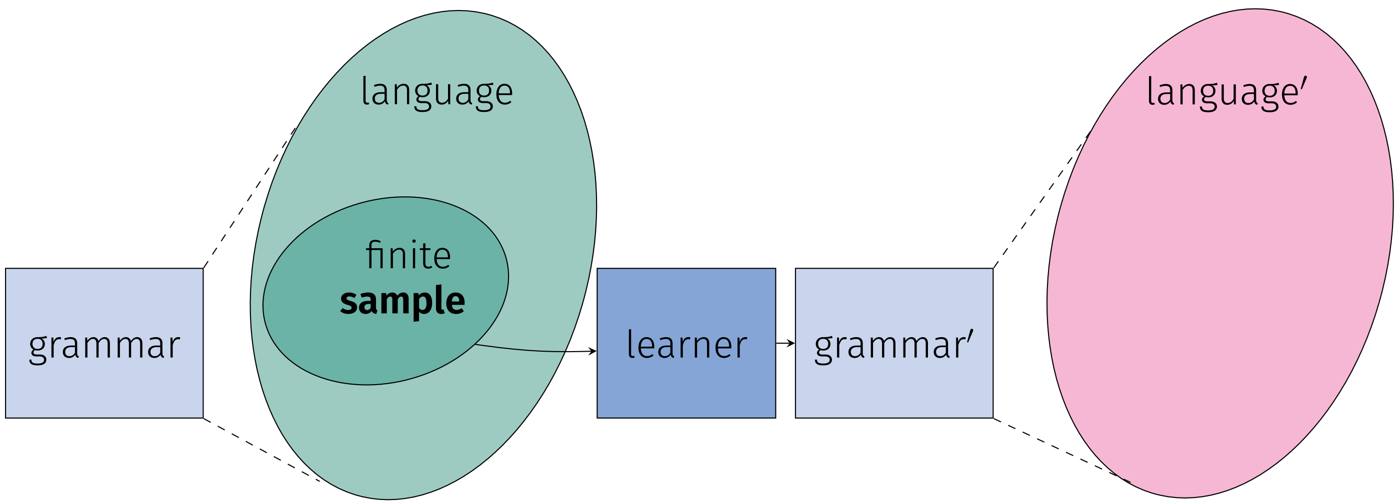
Wenyue Hua



Huteng Dai

What is learning?

What is (language) learning?



Languages and grammars

What is a pattern?

- Well-formedness patterns are **sets**

  ex. *CC

  well-formed: {V, CV, CVV, CVC, CVCV, CVCVC, ..., VVVVCVVV, ...}
  well-formed: {CC, CVCC, CCVC, ..., CVCVCCVCV, ..., CCCCCC, ...}

  ex. SVO word order (with C for complementizer)

  well-formed: $\{\mathrm{SV}, \mathrm{SVO}, \mathrm{SVCSVO}, \mathrm{SCSVVO}, \ldots\}$
  ill-formed: $\{\mathrm{VS}, \mathrm{SOV}, \mathrm{OSV}, \mathrm{SVCSOV}, \ldots\}$

Formal languages

- Sets of strings are **formal languages**
- An **alphabet** Σ is a finite set of symbols

$$\{0, 1\}$$

$$\{a, b, c\}$$

$$\{\text{a, b, c, ..., æ, β, ɔ, ..., z}\}$$

$$\{\text{N, V, ADJ, ..., C}\}$$

Formal languages

- A **string** w over Σ is some sequence $\sigma_1\sigma_2...\sigma_n$ of symbols in Σ.
- Σ^* is all strings over Σ

$$\Sigma = \{a, b, c\}$$

$$\begin{aligned}\Sigma^* = \{\ & \lambda, a, b, c, aa, ab, ac, \\ & ba, bb, bc, ca, cb, cc, \\ & aaa, aab, aac, ..., \\ & abbaaacccbabacb, ...\ \}\end{aligned}$$

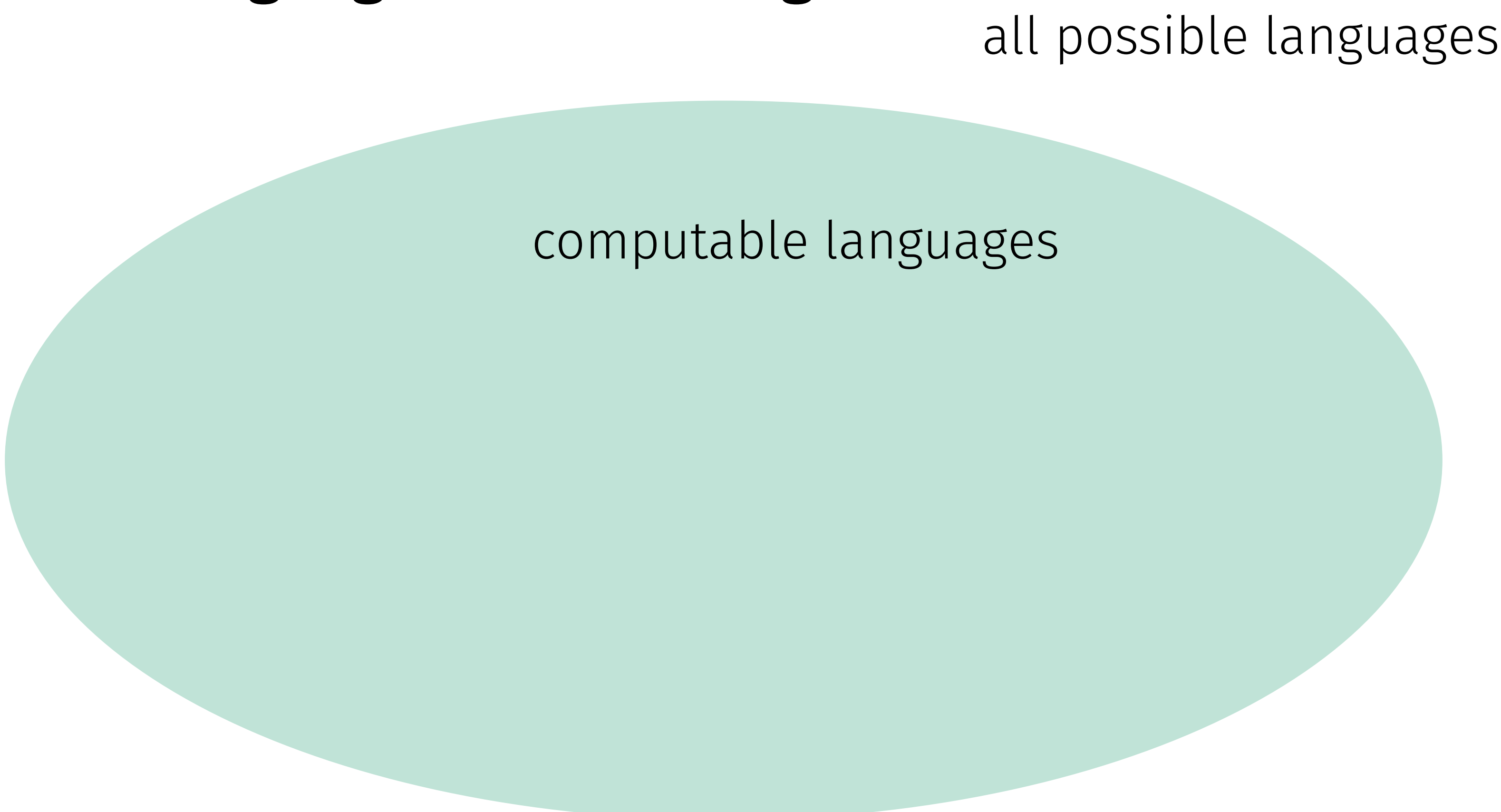
Formal languages

- A **(formal) language** some subset $L \subseteq \Sigma^*$
- Some formal languages for $\Sigma = \{a, b, c\}$:
  - $\{b\}$
  - $(ab)^n = \{\lambda, ab, abab, ababab, ...\}$
  - $a^n b^n = \{\lambda, ab, aabb, aaabbb, aaaabbbb, ...\}$
  - ...

Formal language classes and grammars

all possible languages

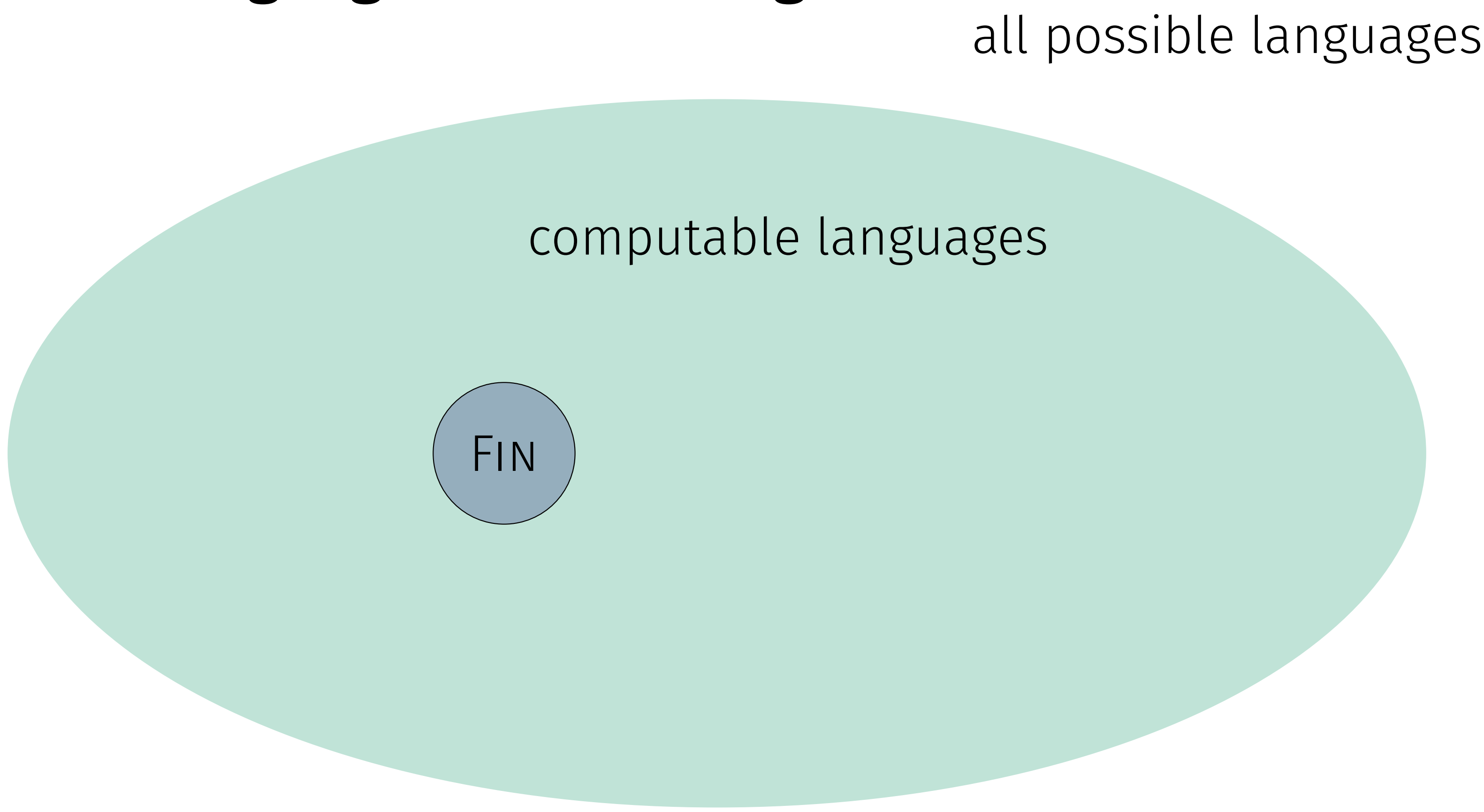
Formal language classes and grammars



Formal language classes and grammars

- **Finite languages (Fɪɴ)**
  - $\{b\}$
  - $\{ab, bab, aaa\}$
  - $\{a, aa, aaa, ..., aaaaaaaaaaaaaaaaaaaaaa\}$
  - ...
- A **grammar** is a finite description of a language
- A grammar for $L \in$ Fɪɴ is just L itself!

Formal language classes and grammars



Formal language classes and grammars

- How would you compute the *CC language?[1]

$$\{\text{V, CV, CVV, CVC, CVCV, CVCVC, ..., VVVVCVVV, ...}\}$$

[1] $\Sigma = \{\text{C, V}\}$

Formal language classes and grammars

- How would you compute the *CC language?[1]

$$\{V, CV, CVV, CVC, CVCV, CVCVC, ..., VVVVCVVV, ...\}$$

- Make sure the string doesn't contain CC sequences!

$$\{CC, CVCC, CCVC, ..., CVCVCCVCV, ..., CCCCCC, ...\}$$

[1] $\Sigma = \{C, V\}$

Formal language classes and grammars

- How would you compute the *CC language?[2]

$$\{\text{V, CV, CVV, CVC, CVCV, CVCVC, ..., VVVVCVVV, ...}\}$$

- Make sure the string doesn't contain CC sequences!

$$\{\text{CC, CVCC, CCVC, ..., CVCVCCVCV, ..., CCCCCC, ...}\}$$

- G for this language:

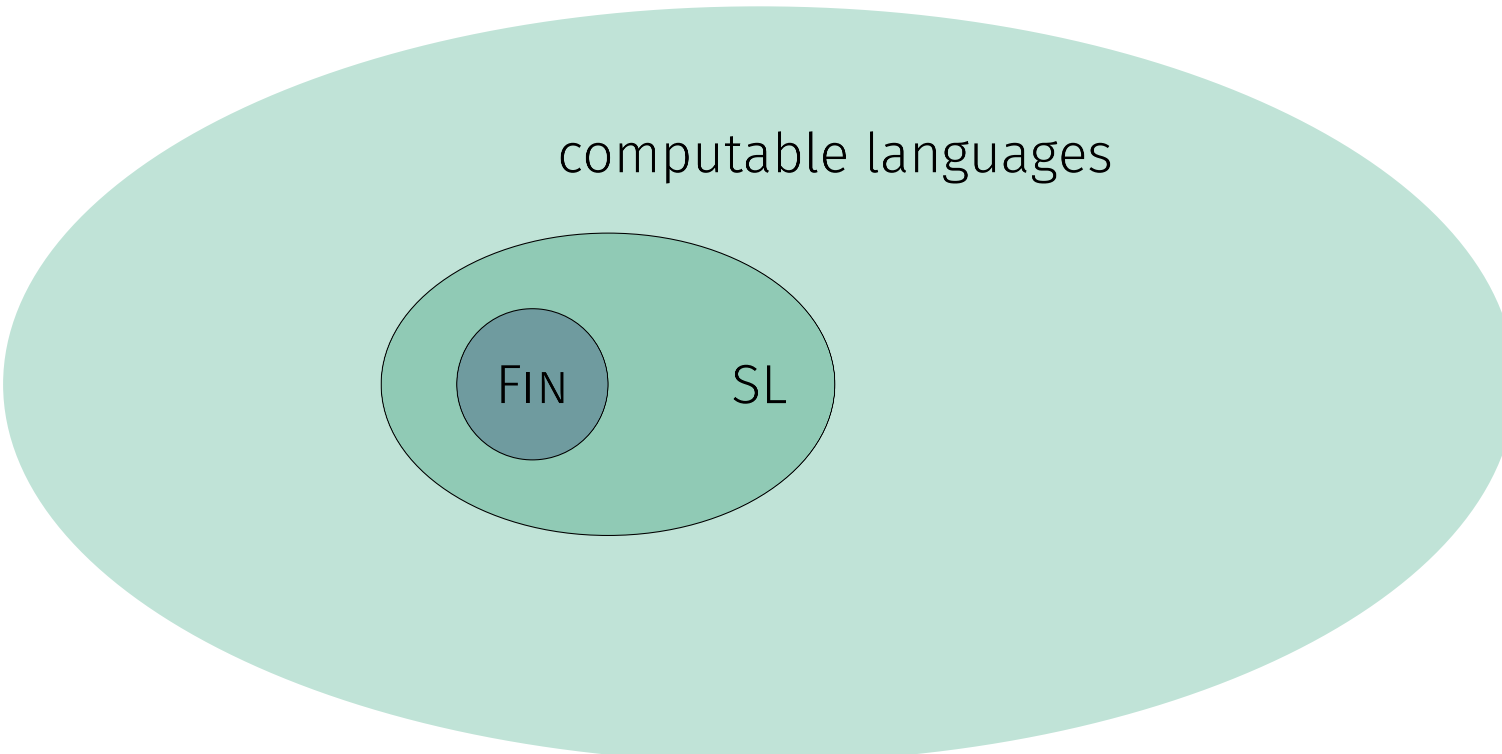
$$\{\text{CC}\}$$

[2] $\Sigma = \{\text{C, V}\}$

Formal language classes and grammars

- A language is **strictly local** iff it is described by a **forbidden substring grammar** (McNaughton and Papert, 1971; Rogers and Pullum, 2011)
- A good many phonotactics are SL (Heinz, 2010)

Formal language classes and grammars



Learning, formally defined

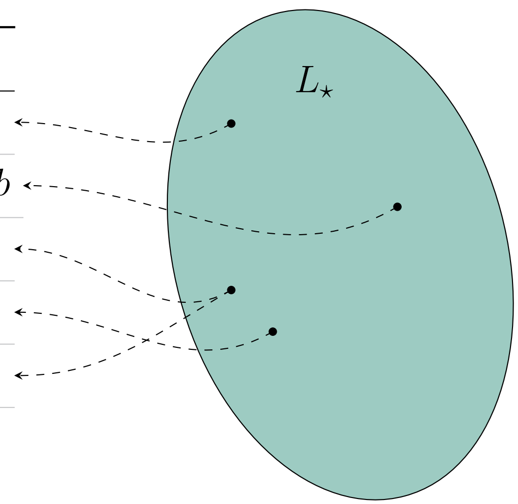
Identification in the limit from positive data (IILPD)

- Gold (1967)
  - on *any* infinite presentation of *positive* examples of target,
  - learner converges *exactly* to target after some *finite* number of examples
- Being (or not) IILPD-learnable is a property of *classes*, not languages

Identification in the limit from positive data (IILPD)

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
$\vdots$	$\vdots$



In the limit, every string in L_* appears in p

Identification in the limit from positive data (IILPD)

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 (IILPD)

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 (IILPD)

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 (IILPD)

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	$\{ab, bab\}$

Identification in the limit from positive data (IILPD)

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	

Identification in the limit from positive data (IILPD)

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	$\{ab, bab\}$

Identification in the limit from positive data (IILPD)

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 (IILPD)

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 (IILPD)

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	$\{ab, bab, aaa\}$
	...	
308	bab	$\{ab, bab, aaa\}$

Identification in the limit from positive data (IILPD)

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

t	$p(t)$	G_t
0	bab	G_0
1	ab	G_1
2	ab	G_2
$\vdots$	$\vdots$	$\vdots$
n	aaa	G_n
$n+1$	bab	G_n
$\vdots$	$\vdots$	$\vdots$
m	ab	G_n
$\vdots$	$\vdots$	$\vdots$

convergence

Identification in the limit from positive data (IILPD)

$\mathcal{A}_{\text{Fin}}$ converges on this p at $t = 3$

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

Note also that $G_t = L_* = \{ab, bab, aaa\}$

Identification in the limit from positive data (IILPD)

$\mathcal{A}_{\text{Fin}}$ converges on **any** p at some finite t

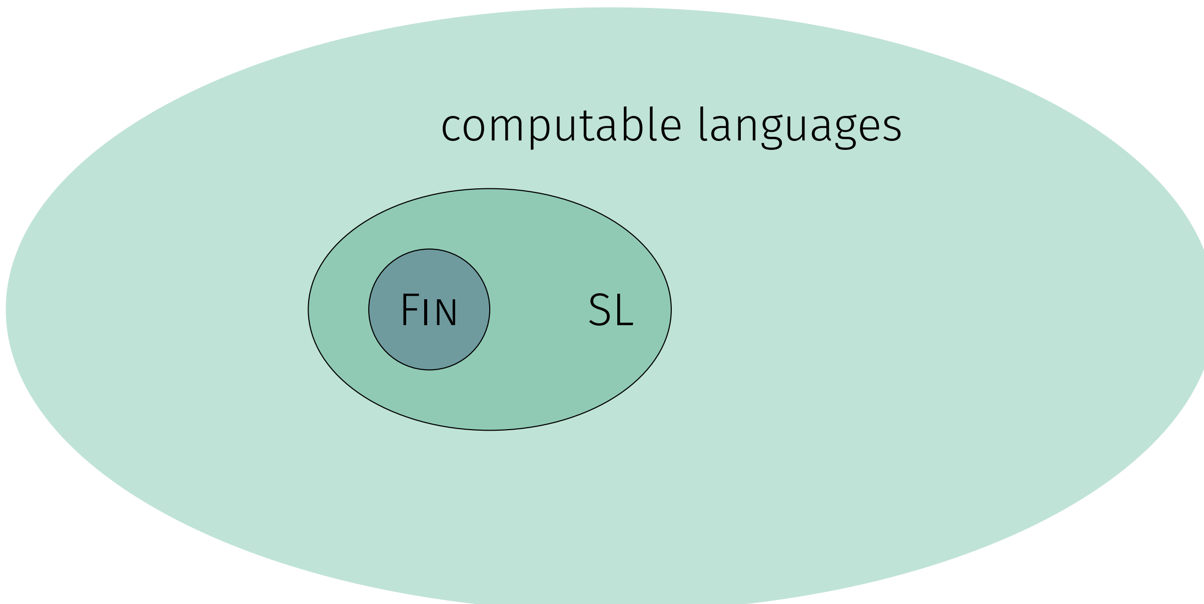
t	$p'(t)$	G_t
0	bab	$\{bab\}$
1	ab	$\{ab, bab\}$
2	ab	$\{ab, bab\}$
3	ab	$\{ab, bab\}$
4	ab	$\{ab, bab\}$
	...	$\{ab, bab\}$
1040	aaa	$\{ab, bab, aaa\}$
	...	$\{ab, bab, aaa\}$

t	$p''(t)$	G_t
0	aaa	$\{aaa\}$
1	aaa	$\{aaa\}$
2	aaa	$\{aaa\}$
3	...	$\{aaa\}$
45	bab	$\{aaa, bab\}$
	...	$\{aaa, bab\}$
23168	ab	$\{ab, bab, aaa\}$
	...	$\{ab, bab, aaa\}$

...

Because any p contains all and only strings in L_*, $G_t = L_*$ at some t

Identification in the limit from positive data (IILPD)



- $\mathcal{A}_{\mathrm{Fin}}$ only ever returns a language in Fin

Identification in the limit from positive data (IILPD)

IILPD-learnability

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

Identification in the limit from positive data (IILPD)

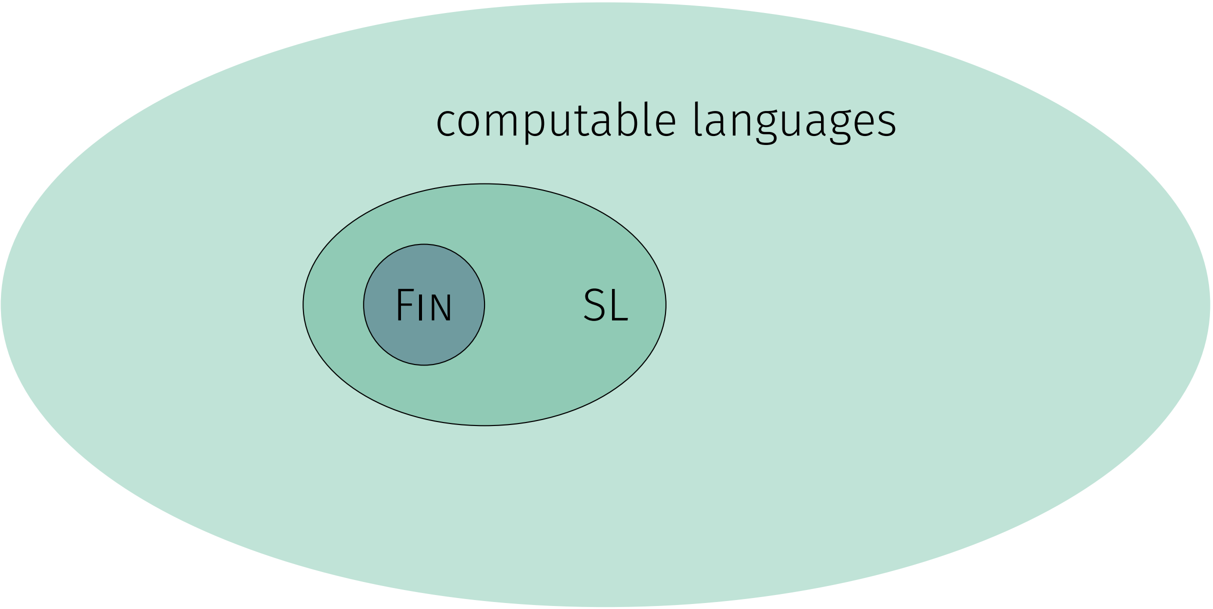
IILPD-learnability

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

Strengths

- Works on **any** presentation of L
- Works with positive data **only**
- Identifies target **exactly**

Identification in the limit from positive data (IILPD)



- Fixed size k of substrings $\Longrightarrow$ SL_k is IILPD-learnable

Identification in the limit from positive data (IILPD)

IILPD of SL$_k$ languages

$$G_\star = \{CC\}$$

t	datum	hypothesis $(k = 2)$
0	VC	
1	$CVCVC$	
2	$CVVCVCV$	
3	$VCVCV$	
$\vdots$		

Identification in the limit from positive data (IILPD)

IILPD of SL$_k$ languages

$$G_\star = \{CC\}$$

t	datum	hypothesis ($k = 2$)
		$\{CC, CV, VC, VV\}$
0	VC	
1	$CVCVC$	
2	$CVVCVCV$	
3	$VCVCV$	
$\vdots$		

Identification in the limit from positive data (IILPD)

IILPD of SL$_k$ languages

$$G_\star = \{CC\}$$

t	datum	hypothesis $(k=2)$
		$\{CC, CV, VC, VV\}$
0	VC	$\{CC, CV, VC, VV\}$
1	$CVCV$	
2	$CVVCVCV$	
3	$VCVCV$	
⋮		

Identification in the limit from positive data (IILPD)
IILPD of SL$_k$ languages

$$G_\star = \{CC\}$$

t	datum	hypothesis $(k=2)$
		$\{CC, CV, VC, VV\}$
0	VC	$\{CC, CV, VC, VV\}$
1	$CVCV$	$\{CC, CV, VC, VV\}$
2	$CVVCVCV$	
3	$VCVCV$	
$\vdots$		

Identification in the limit from positive data (IILPD)

IILPD of SL$_k$ languages

$$G_\star = \{CC\}$$

t	datum	hypothesis $(k = 2)$
		$\{CC, CV, VC, VV\}$
0	VC	$\{CC, CV, VC, VV\}$
1	$CVCV$	$\{CC, CV, VC, VV\}$
2	$CVVCVCV$	$\{CC, CV, VC, VV\}$
3	$VCVCV$	
$\vdots$		

Identification in the limit from positive data (IILPD)

IILPD of SL$_k$ languages

$$G_\star = \{CC\}$$

t	datum	hypothesis ($k = 2$)
		$\{CC, CV, VC, VV\}$
0	VC	$\{CC, CV, VC, VV\}$
1	$CVCV$	$\{CC, CV, VC, VV\}$
2	$CVVCVCV$	$\{CC, CV, VC, VV\}$
3	$VCVCV$	$\{CC, CV, VC, VV\}$

⋮

Identification in the limit from positive data (IIILPD)

IILPD of SL$_k$ languages

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

- Guaranteed to converge as soon as we see $\texttt{substrings}_k(L_\star)$
- The time it takes to calculate is directly proportional to the size of the data sample.

Identification in the limit from positive data (IILPD)

Gold (1967): Any class C containing all of FIN and at least one infinite language **is not** IILPD-learnable

- **Reason:** there are presentations p for which any $p[t]$ is consistent with some finite $L_{\text{fin}} \in \mathcal{C}$ and the infinite $L_{\text{inf}} \in \mathcal{C}$
- Most language classes are not IILPD-learnable!
  - SL when k is not fixed
  - Regular, Context-Free, etc.

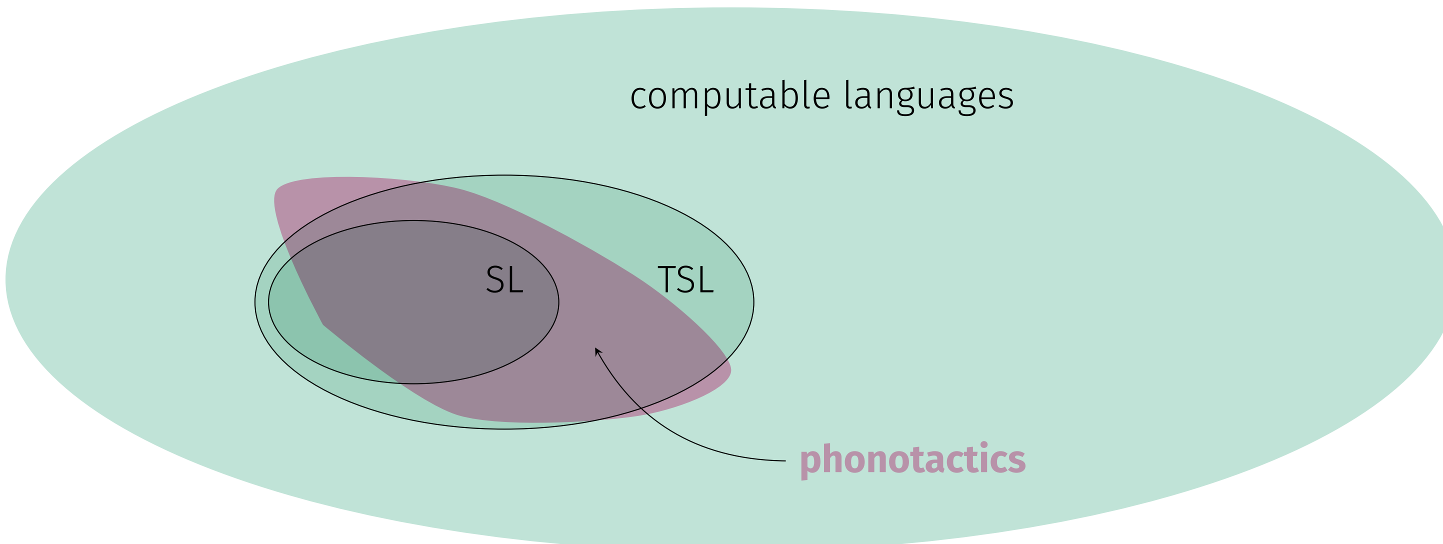
Identification in the limit from positive data (IILPD)

Gold (1967): Any class C containing all of FIN and at least one infinite language **is not** IILPD-learnable

- Learners must be restricted to some (non-superfinite) class to be successful IILPD (Angluin, 1982)
- This fact can be interpreted to give mathematical weight the poverty of the stimulus argument for UG

Identification in the limit from positive data (IILPD)

- Much (all?) of phonology lies in IILPD-learnable classes (Heinz, 2018)



  - TSL = **tier-based** strictly local (Heinz et al., 2011; Jardine and Heinz, 2016; McMullin and Hansson, 2016)

Other paradigms

Other paradigms

- Criticisms of IILPD as a model of human learning:
  - requires success on “adversarial” presentations
  - no “stochastic learning”

  - no considerations of feasibility
  - exact convergence is too hard
  - absence of noise is too easy

Other paradigms

IILPD from computable presentations

Gold (1967): The **entire class of computable languages** is learnable in the limit from **positive, computable** presentations.

- However, the learner is not **feasible**
- It is an enumerative learner that "guesses" the machine generating the presentation
- Is experience computable?

Other paradigms

IILPD with probability p

Angluin (1988): If we require learner to identify target with $p > 2/3$, then IILPD with probability p is same as IILPD

- In this paradigm, learners can behave randomly (e.g. flip coins)
- However, Angluin finds that "if the probability of identification is required to be above some threshold, randomization is no advantage" (p. 5)

Other paradigms

IIL from positive stochastic distributions

Angluin (1988): If we require learner to identify with $p > 2/3$, then IIL from positive stochastic distributions is same as IILPD

- In this paradigm, presentations are drawn from some stochastic distribution
- Learner must succeed on *any* distribution
- "[G]iven a presentation on which the normal nonprobabilistic learner fails, we can construct a corresponding distribution on which the probabilistic learner will fail." (Clark and Lappin, 2011, p. 110)

Other paradigms

IIL from restricted distributions

- Horning (1969): probabilistic context-free grammars can be learned from positive data with probability 1
- Osherson et al. (1986) extend this to all computable stochastic languages, given a fixed set of distributions

- Learning target is stochastic formal languages
- Results hold only for a restricted set of fixed distributions
- Distributions are *computable* (like in Gold 1967!)
- Similarly, learner is not feasible

Other paradigms

Summary

- Criticisms of IILPD as a model of human learning:
  - requires success on “adversarial” presentations
  - no “stochastic learning”

  - no considerations of feasibility
  - exact convergence is too hard
  - absence of noise is too easy

Other paradigms

Summary

- Gold (1967): no general learner for IILPD
- Naively adopting "stochastic learning" does not increase learning power
- Restricting distributions makes a difference (Horning, 1969; Osherson et al., 1986)
- So does restricting presentations! (Gold, 1967)
- For more see Heinz (2016)!

Other paradigms

Feasibility

- de la Higuera (2010): Identification in the limit in polynomial time and data

- This is based on sample *sets*, rather than *presentations*

Other paradigms

Inexact identification

- Osherson et al. (1986): IIDLP with finite number of errors
  - Makes learning easier, but not enough to learn all computable languages
- **Probably Approximately Correct (PAC)** learning (Valiant, 1984)
  - Probabilistic framework with explicit negative examples
  - Not even FIN is PAC-learnable!

Noise

Noise

- Naturalistic linguistic experience is not perfect
- **Noise** encapsulates errors and exceptions

Noise

Noisy presentation

For a language L, a presentation p is a **noisy presentation of** L iff it is a positive presentation of $L \cup X$ for some finite set X

IIL from noisy presentations (Osherson et al., 1986)

For a class $\mathcal{C}$ to be IIL from noisy presentations, for any $L_1, L_2 \in \mathcal{C}$, both $L_1 - L_2$ and $L_2 - L_1$ must be infinite.

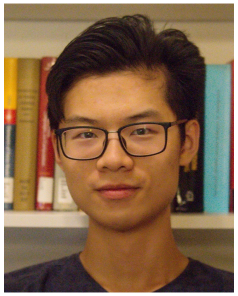
Noise

IIL from noisy presentations (Osherson et al., 1986)

For a class $\mathcal{C}$ to be IIL from noisy presentations, for any $L_1, L_2 \in \mathcal{C}$, both $L_1 - L_2$ and $L_2 - L_1$ must be infinite.

- Even with fixed substring size k, SL is not IIL from noisy presentations

Noise



- Dai (submitted)
  - SL learner ($k = 2$) for learning with noise
  - Empirical tests on English and Turkish
  - Works as well as MaxEnt (Hayes and Wilson, 2008)
- Probabilistic grammars not necessary to deal with noise
- Current work: what kind of presentations does Dai Algorithm work on?
- What kind of presentations are necessary for any algorithm to work?

Discussion

Discussion

- Computational learning theory investigates the logic of learning
- Necessarily, it makes idealizations (like IILPD)
- However, it motivates empirical investigations:
  - What classes do human language learners target?
  - What assumptions do human language learner make about the data presentation?

Thank you!

…and also thanks to Huteng Dai, Jeff Heinz, and the Rutgers Mathematical Linguistics Group

Reading list (in recommended reading order)

Jonathan Rawski and Jeffrey Heinz. 2019. No Free Lunch in Linguistics or Machine Learning: Response to Pater. *Language* , 95(1):e125–e135. (pdf)

Heinz, Jeffrey. 2016. Computational Theories of Learning and Developmental Psycholinguistics. In Jeffrey Lidz, et al., editors, *The Oxford Handbook of Developmental Linguistics*, chapter 27, pages 633–663. Oxford University Press. (pdf)

James Rogers and Geoffrey K. Pullum. 2011. Aural Pattern Recognition Experiments and the Subregular Hierarchy. *Journal of Logic, Language, and Information*, Vol. 20, No. 3. (pdf)

Clark, Alexander, and Shalom Lappin. 2011. *Linguistic Nativism and the Poverty of the Stimulus*. Wiley-Blackwell.

Partha Niyogi. 2006. *The Computational Nature of Language Learning and Evolution.* MIT Press.

Full references

Angluin, D. (1982). Inference of reversible languages. *J. ACM*, 29(3):741–765.

Angluin, D. (1988). Identifying languages from stochastic examples. Technical Report 615, Yale University, New Haven, CT.

Chomsky, N. (1965). *Aspects of Theory of Syntax*. Cambridge, Massachusetts: MIT Press.

Clark, A. and Lappin, S. (2011). *Linguistic Nativism and the Poverty of the Stimulus*. Wiley-Blackwell.

de la Higuera, C. (2010). *Grammatical Inference: Learning Automata Grammars*. Cambridge University Press.

Gold, M. E. (1967). Language identification in the limit. *Information and Control*, 10:447–474.

Goldberg, A. E. (2009). Constructions work. *Cognitive Linguistics*, 20:201–224.

Hayes, B. and Wilson, C. (2008). A maximum entropy model of phonotactics and phonotactic learning. *LI*, 39:379–440.

Heinz, J. (2010). Learning long-distance phonotactics. *LI*, 41:623–661.

Heinz, J. (2018). The computational nature of phonological generalizations. In Hyman, L. and Plank, F., editors, *Phonological Typology*, Phonetics and Phonology, chapter 5, pages 126–195. De Gruyter Mouton.

Heinz, J., Rawal, C., and Tanner, H. G. (2011). Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64, Portland, Oregon, USA. Association for Computational Linguistics.

Horning, J. J. (1969). *A Study of Grammatical Inference*. PhD thesis, Stanford University.

Jardine, A. and Heinz, J. (2016). Learning tier-based strictly 2-local languages. *Transactions of the Association for Computational Linguistics*, 4:87–98.

McMullin, K. and Hansson, G. O. (2016). Long-distance phonotactics as Tier-Based Strictly 2-Local languages. In *Proceedings of AMP 2015*.

McNaughton, R. and Papert, S. (1971). *Counter-Free Automata*. MIT Press.

Osherson, D. N., Stob, M., and Weinstein, S. (1986). *Systems That Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists*. Cambridge, MA: The MIT Press.

Rogers, J. and Pullum, G. (2011). Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*, 20:329–342.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.