Talk slides:

`adamjardine.net/files/jardine-msutalk2023.pdf`

Why computational learning theory matters for language learning

Adam Jardine
Rutgers University

September 28, 2023 · Montclair State Brown Bag Series

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 (and others') claim:

- Gold (1967): No restrictions on data presentation $\Longrightarrow$ 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
  - (or both!)

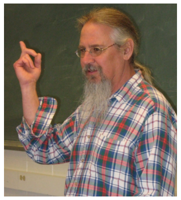
- Computational learning theory is a framework for...
  - rigorously studying the logic of learning problems
  - ...and solutions!
  - developing restrictive, testable hypotheses about language learning

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)

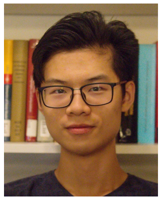

Jane Chandlee
(Haverford)


Kevin McMullin
(Ottowa)

...at Rutgers:


Tatevik Yolyan


Wenyue Hua


Huteng Dai

Empirical vs. theoretical learning models

Empirical vs. theoretical learning models

- ***Empirical*** - running models on corpora

- ***Theoretical*** - proving conditions under which a learning algorithm succeeds

(see Niyogi (2006); Heinz et al. (2016); Clark (2017))

Empirical vs. theoretical learning models

Why theoretical?

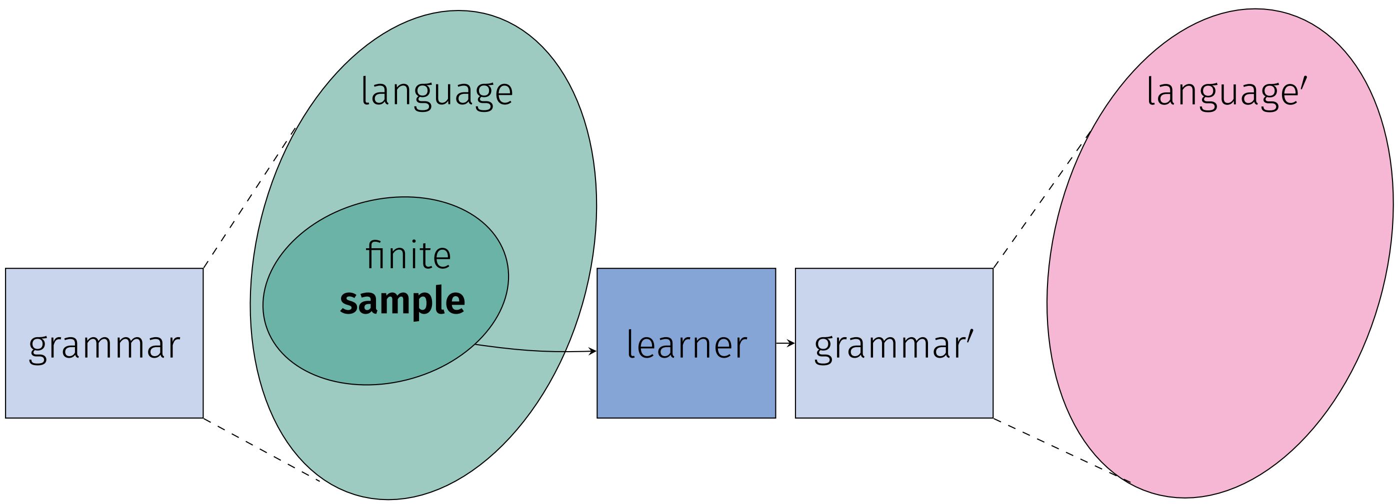
- Requires some idealization and assumption
- But, "when [empirical] algorithms do work, we do not know why they work or what properties of the languages they rely on...
  ...[T]he method of mathematical proof will give us the strongest possible guarantees. Moreover, we will often have a precise understanding of the properties of the grammars and languages that allow them to be learned..." (Clark, 2017, p.109)

What is (language) learning?

What is (language) learning?

- What is the learning *target*?
- What is the *nature of the input* to the learner?
- What are the *conditions of success*?

What is (language) learning?



What is a language?

- Grammaticality patterns are ***formal languages***

  ex. SVO word order (with C for complementizer)

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

  ex. *CC, *VV

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

What is a grammar?

- A ***grammar*** (G) is a finite description of a formal language

$$
\begin{aligned}
I &\rightarrow XY \\
X &\rightarrow \mathrm{S} \\
X &\rightarrow \mathrm{SC}I \\
Y &\rightarrow \mathrm{V} \\
Y &\rightarrow \mathrm{VO} \\
Y &\rightarrow \mathrm{VC}I
\end{aligned}
$$

G for SVO word order

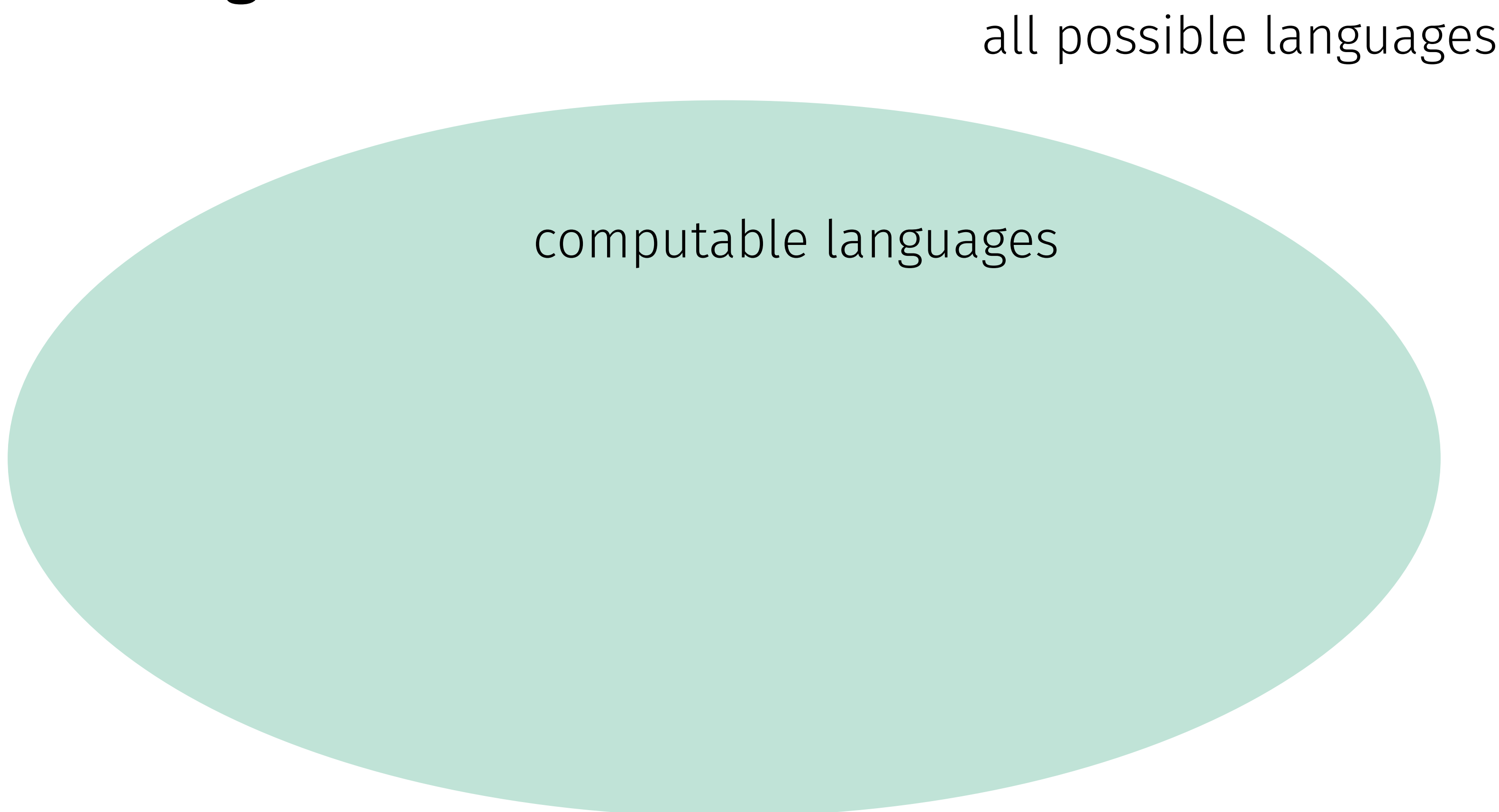
*CC, *VV

G for *CC, *VV language

What is a grammar?

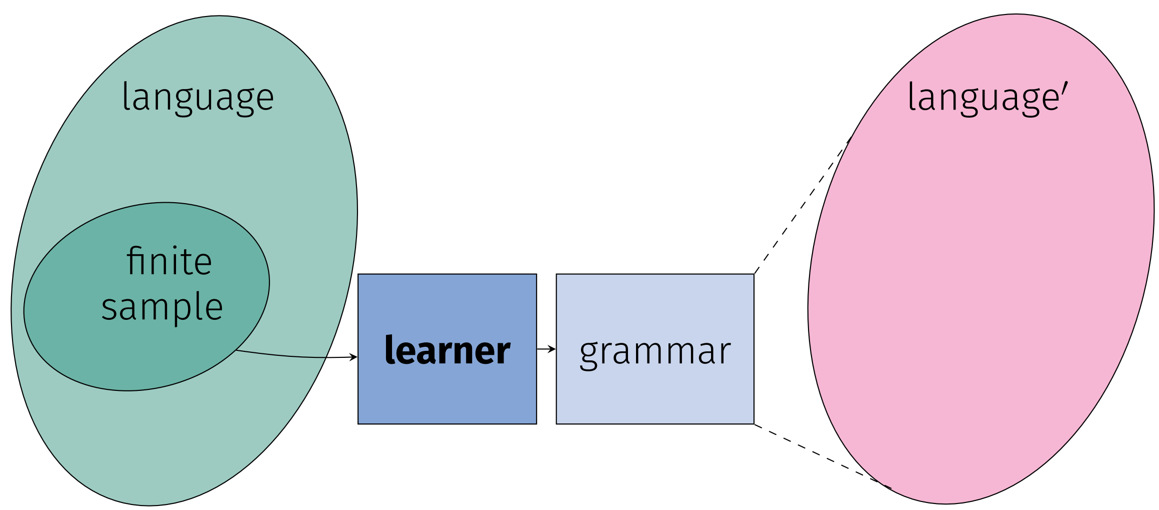
all possible languages

What is a grammar?



What is a learner?

- ***Learner***: a function that takes a finite sample of data and outputs a grammar



- **Question:** Is there a learner for the computable languages?

Learning, formally defined

Identification in the limit from positive data (IILPD)

- Gold (1967): first to formalize learning, in several ways
- Computable languages are *not* learnable from positive examples[1]

[1]Unless there are restrictions on how the examples are presented.

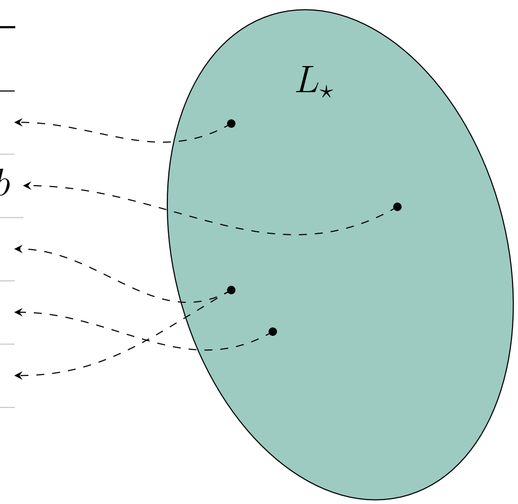
Identification in the limit from positive data (IILPD)

- Gold (1967)
  - for *any* target in a class,
  - on *any* infinite presentation of *positive* examples of that target,
  - learner *converges* to target *exactly* after some *finite* number of examples

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	ab
4	$bbab$
$\vdots$	$\vdots$



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 initial segment* of p and outputs a grammar

t	$p(t)$
0	$abab$
1	$ababab$
2	ab
3	ab
$\vdots$	$\vdots$
i	$ababab$
$i+1$	bab
$i+2$	$ababab$
$\vdots$	$\vdots$

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

Identification in the limit from positive data (IILPD)

A learner for *XY constraints

- Assume all *XY constraints
- If you see XY in presentation, remove *XY from guess

Identification in the limit from positive data (IILPD)

$L_\star =$ the *CC, *VV language

t	$p(t)$	hypothesis
0	VC	
1	$CVCVC$	
2	$CVCVCV$	
3	$VCVCV$	
$\vdots$		

Identification in the limit from positive data (IILPD)

$L_\star =$ the *CC, *VV language

t	$p(t)$	hypothesis
		$\{CC, CV, VC, VV\}$
0	VC	
1	$CVCVC$	
2	$CVCVCV$	
3	$VCVCV$	
$\vdots$		

Identification in the limit from positive data (IILPD)

$L_\star =$ the *CC, *VV language

t	$p(t)$	hypothesis
		$\{CC, CV, VC, VV\}$
0	VC	$\{CC, CV, VC, VV\}$
1	$CVCV$	
2	$CVCVCV$	
3	$VCVCV$	
$\vdots$		

Identification in the limit from positive data (IILPD)

$L_\star =$ the *CC, *VV language

t	$p(t)$	hypothesis
		$\{CC, CV, VC, VV\}$
0	VC	$\{CC, CV, VC, VV\}$
1	$CVCV$	$\{CC, CV, VC, VV\}$
2	$CVCVCV$	
3	$VCVCV$	
$\vdots$		

Identification in the limit from positive data (IILPD)

$L_\star =$ the *CC, *VV language

t	$p(t)$	hypothesis
		$\{CC, CV, VC, VV\}$
0	VC	$\{CC, CV, VC, VV\}$
1	$CVCV$	$\{CC, CV, VC, VV\}$
2	$CVCVCV$	$\{CC, CV, VC, VV\}$
3	$VCVCV$	
$\vdots$		

Identification in the limit from positive data (IILPD)

$L_\star =$ the *CC, *VV language

t	$p(t)$	hypothesis
		$\{CC, CV, VC, VV\}$
0	VC	$\{CC, CV, VC, VV\}$
1	$CVCV$	$\{CC, CV, VC, VV\}$
2	$CVCVCV$	$\{CC, CV, VC, VV\}$
3	$VCVCV$	$\{CC, CV, VC, VV\}$

$\vdots$

Identification in the limit from positive data (IILPD)

A different presentation...

$$L_\star = \text{the *CC, *VV language}$$

t	$p(t)$	hypothesis
		$\{CC, CV, VC, VV\}$
0	$CVCVCV$	$\{CC, CV, VC, VV\}$
1	CV	$\{CC, CV, VC, VV\}$
$\vdots$		
3	$VCVC$	$\{CC, CV, VC, VV\}$
$\vdots$		

Identification in the limit from positive data (IILPD)

A different language...

$$L_\star = \text{the *CC language}$$

t	$p(t)$	hypothesis
		$\{CC, CV, VC, VV\}$
0	VC	$\{CC, CV, VC, VV\}$
1	$CVCV$	$\{CC, CV, VC, VV\}$
2	$CVCVCV$	$\{CC, CV, VC, VV\}$
$\vdots$		
57	$VCVVCV$	$\{CC, CV, VC, VV\}$
$\vdots$		

Identification in the limit from positive data (IILPD)

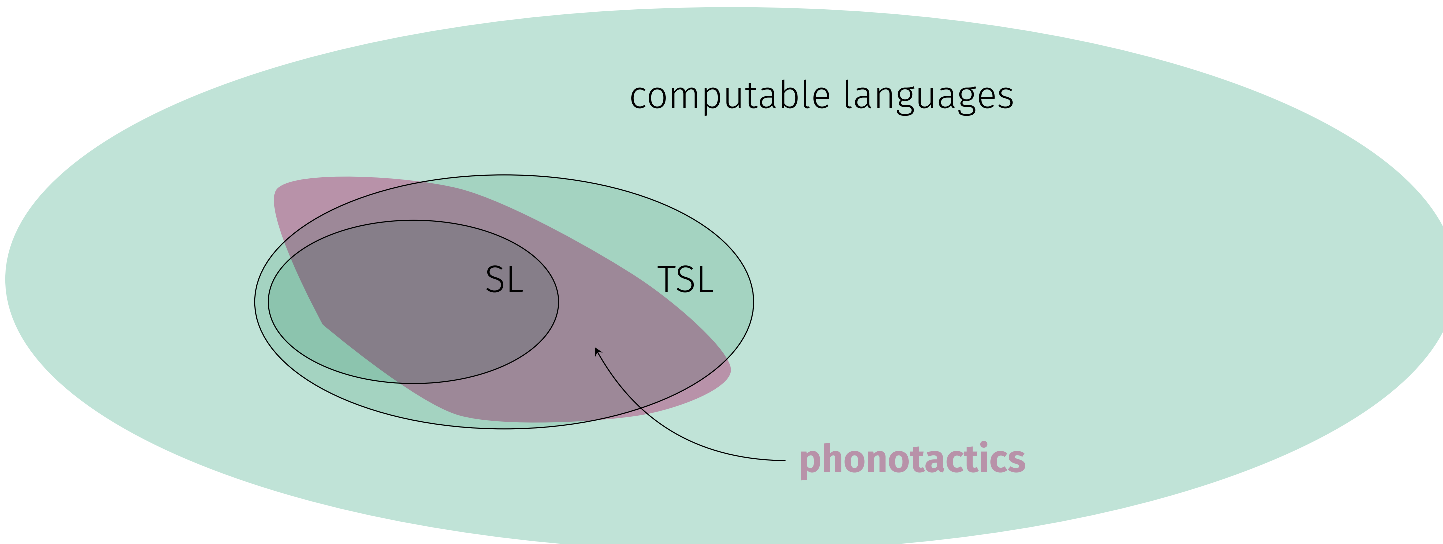
Definition: A class is *IILPD-learnable* iff there is an algorithm $\mathcal{A}$ such that for *any* language L in the class, for *any* presentation p of that language, $\mathcal{A}$ converges to a grammar for L on p at some finite i.

Identification in the limit from positive data (IILPD)

- A language is **strictly local (SL)** iff it is described by a **forbidden substring grammar** (McNaughton and Papert, 1971; Rogers and Pullum, 2011)
- For any fixed length k, SL_k is IILDP-learnable

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)

Identification in the limit from positive data (IILPD)

Strengths

- An IILPD-provable learner works on **any** presentation of L
- Works with positive data **only**
- Identifies target **exactly**

Abstracts away from...

- gaps or noise
- feasibility (time or data required)

Identification in the limit from positive data (IILPD)

Gold (1967): The entire computable class **is not** IILPD-learnable

- **Reason:** for *any* finite presentation, there are *at least two computable languages* consistent with that presentation
- 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): The entire computable class **is not** IILPD-learnable

- Learners must be restricted to some class to be successful IILPD (Angluin, 1982)

- This fact can be interpreted to give mathematical weight the poverty of the stimulus argument for UG

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

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

- 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)!

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

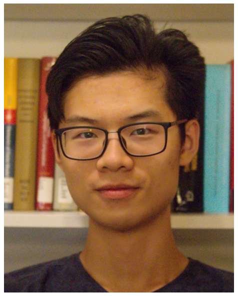
IIL from noisy presentations (Osherson et al., 1986)

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)

Jordan Kodner, Sarah Payne, and Jeffrey Heinz. 2023. Why Linguistics Will Thrive in the 21st Century: A Reply to Piantadosi. LingBuzz 007485. (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. (2017). Computational learning of syntax. *Annual Review of Linguistics*, 3:107–123.

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

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. (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., de la Higuera, C., and van Zaanen, M. (2016). *Grammatical Inference for Computational Linguistics*. Number 28 in Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

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.

Niyogi, P. (2006). *The Computational Nature of Language Learning and Evolution*. Cambridge, MA: 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.

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