Learning parametric stress without domain-specific mechanisms

Aleksei Nazarov (Harvard University)

Gaja Jarosz (University of Massachusetts Amherst)

Introduction: Parameters

- Chomsky (1981): Principles and Parameters
  - UG encodes fixed universals and finite number of choices languages can make
    - Language learner's task: find settings of parameters
- Applied to stress systems: Dresher and Kaye (1990), Hayes (1995)
    - L-to-R or R-to-L? Trochee or Iamb?
      (σ ˌσ)(σ ˈσ) σ vs. σ (ˌσ σ)(ˈσ σ)
    - QS or QI?
      (kà.maa$_{\sigma}$)(kà$_{\sigma}$.ta$_{\sigma}$)(mák$_{\sigma}$.ma$_{\sigma}$) vs. (kà$_{L}$)(màa$_{H}$.ka$_{L}$)(tà$_{L}$)(mák$_{H}$.ma$_{L}$)

Introduction: Previous proposals

- For stress parameters: domain-specific learning mechanisms argued to be necessary, e.g., Dresher and Kaye (1990):
  - Parameters set in particular innately specified order
  - Each parameter innately associated with a "cue": configuration in data that triggers marked value
    - E.g., QS starts out set to Off. If corpus contains two words of same length with different stress, set QS to On.
- See similar work on "triggering" in learning syntax (Gibson and Wexler 1994, Berwick and Niyogi 1996, Lightfoot 1999)

Introduction: Previous proposals

- Statistical learning of parameters: argued to be insufficient for stress
- Naïve Parameter Learner (NPL; Yang 2002) domain-general learner for parameters (syntax or phonology)
- Pearl (2007, 2011) argues: NPL must be supplemented with domain-specific mechanisms to learn stress
  - Parameter ordering
  - Cues (Dresher and Kaye) or parsing method (Fodor 1998, Sakas and Fodor 2001) for disambiguation

Introduction: Current proposal

- Proposal: Slightly richer statistical learning model, no domain-specific mechanisms
  - Expectation Driven Parameter Learner (EDPL; based on Jarosz submitted)
    - Parameter update sensitive to ambiguity between parameter settings compatible with a data point
- EDPL and NPL (no domain-specific mechanisms) tested on languages predicted by Dresher and Kaye (1990)
  - First typologically extensive tests for NPL
  - EDPL massively outperforms NPL (96.0% success vs. 4.3% success)
- We argue that conclusions about necessity of domain-specific mechanisms are premature

The Learner

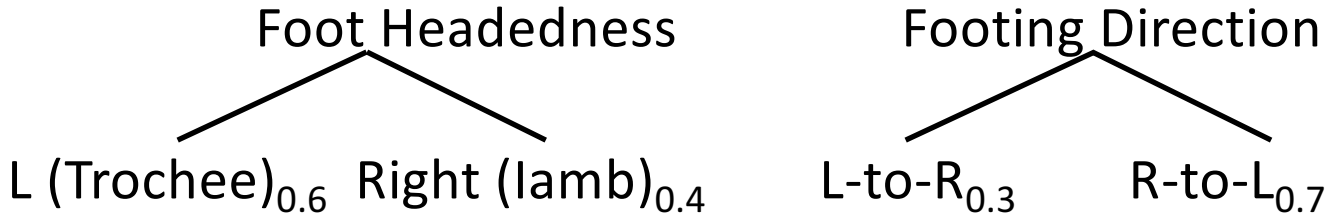
NPL & EDPL overview

NPL	EDPL
Stochastic parameter grammar	
Grammar incrementally updated by Linear Reward-Penalty Scheme (Bush and Mosteller 1951) after each data point	

• NPL samples parameter settings once and uses them to generate output
  • Match → reward all parameters equally
  • Mismatch → penalize all parameters equally
	• EDPL computes individual updates for each parameter
	  • Based on Jarosz' (submitted) Expectation Driven Learning
	  • Requires more computation, but still linear in the number of parameters

Grammar

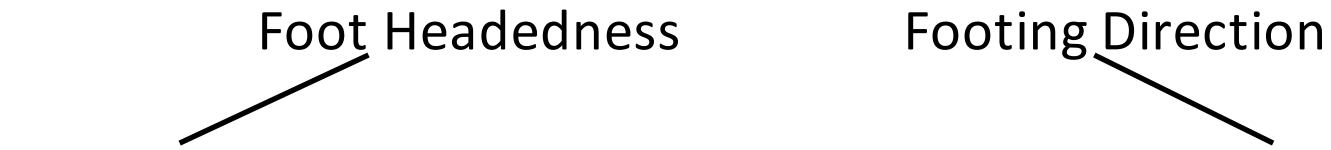
- Stochastic parameter grammar
  - Probability distribution over each parameter's possible settings



- Each time grammar generates output: one setting categorically chosen for each parameter (weighted coin flip)

Grammar

- Stochastic parameter grammar
  - Probability distribution over each parameter's possible settings

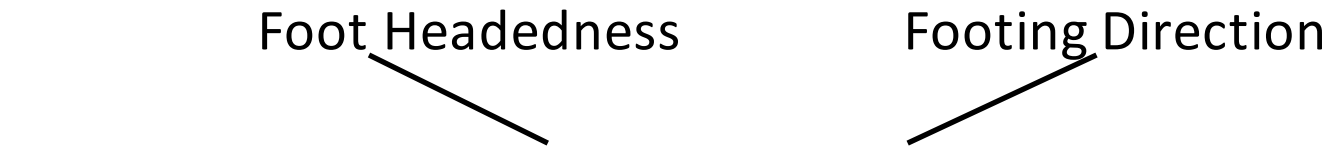


(kà$_L$)**(màa$_H$.**ka$_L$)(tà$_L$)**(mák$_H$.**ma$_L$) (p = 0.6 x 0.7 = 0.42)

- Each time grammar generates output: one setting categorically chosen for each parameter (weighted coin flip)

Grammar

- Stochastic parameter grammar
  - Probability distribution over each parameter's possible settings

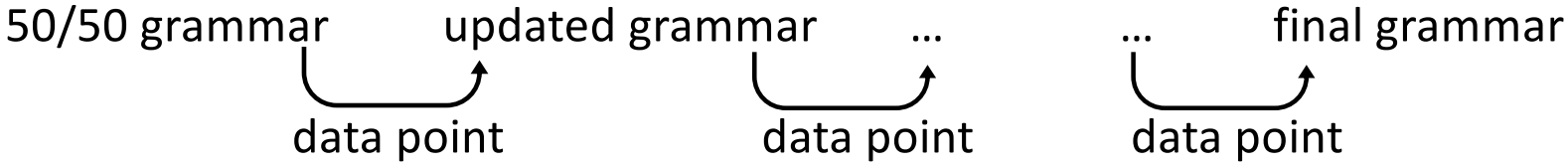


(ka$_L$.**màa$_H$**)(ka$_L$.**tà$_L$**)(màk$_H$)(má$_L$) (p = 0.4 x 0.3 = 0.12)

- Each time grammar generates output: one setting categorically chosen for each parameter (weighted coin flip)

Gradualness

- Gradual, incremental updating of grammar
  - Start from a uniform distribution over parameter settings
  - Data points presented to learner one-by-one
  - For each parameter setting: new probability is weighted average of old probability and R_d (reward value – between 1 and 0) (**Linear Reward-Penalty Scheme**)



Update rule

- Linear Reward-Penalty Scheme (Bush and Mosteller 1951):
  - For each parameter setting ψ_i (FootHead = L, FootHead = R, etc.):

$$p(\psi_i)_{new} = R_d(\psi_i) * \lambda + p(\psi_i)_{old} * (1 - \lambda)$$

  - $p(\psi_i)_{old}$ is the parameter setting's probability in the grammar before the update (e.g. $p(\text{FootHead=L})_{old} = 0.6$)
  - $R_d(\psi_i)$ is the reward value, between 1 and 0 (see next slide)
  - λ is the learning rate, between 1 and 0; here, $\lambda = 0.1$ was chosen

Reward computation (NPL)

- Reward value (R_d) is at least 0 and at most 1
- **NPL**: reward always 0 or 1, based on one single attempt to generate a data point given old parameter setting probabilities
- $R_d = 1$ for:
  - **all** parameter settings chosen in successful attempt to generate current data point and
  - **all** parameter settings not chosen in unsuccessful attempt to generate current data point
- $R_d = 0$ elsewhere

NPL: Uniform reward values

- NPL has no way to determine when parameters matter (or not)
- Yang (2002; p42): “The NPL model may reward wrong parameter values as hitchhikers, and punish correct parameter values as accomplices. The hope is that, in the long run, the correct parameter values will prevail.”
  - Essential parameters can be ‘blamed’ for others’ failures
  - Irrelevant parameters can be credited/blamed
- This weakens learner’s ability to deal with the Credit-Blame problem (Dresher and Kaye 1990)

NPL: Uniform reward values

- E.g., attempt to generate ***pátakana****:*
  - MainStress=L
    - **essential** to getting *pátakana*
  - Ext.m.Edge = L
    - **incompatible** with generating *pátakana*
  - **both penalized** for generating incorrect stress pattern (MainStress = L penalized as “accomplice”)

Reward computation (EDPL)

- **EDPL**: a parameter setting's reward (R_d) is the probability of that parameter setting given the data point at hand: **$p(\psi_i|$data point$)$**
- Can be factored into terms we can easily estimate (Jarosz submitted):

$$p(\psi_i|data\ point) = \frac{p(data\ point|\psi_i) * \ p(\psi_i)}{p(data\ point)}$$

Reward computation (EDPL)

- **EDPL**: a parameter setting's reward (R_d) is the probability of that parameter setting given the data point at hand: **$p(\psi_i|$data point$)$**
- Can be factored into terms we can easily estimate (Jarosz submitted):

Estimated by sampling

Taken from current grammar

$$p(\psi_i|data\ point) = \frac{p(data\ point|\psi_i) * \ p(\psi_i)}{p(data\ point)}$$

Weighted sum of p(data point$|\psi$) for all settings of the current parameter

EDPL: Non-uniform reward values

- Estimation of p(data point|ψ_i):
  - Temporarily set ψ_i (e.g FootHead=L) to 1 and generate current data point *r* times
  - Compute proportion correct results out of *r* attempts (we chose *r* = 50)
- Weighted sum of p(data point|ψ) for all settings of the parameter yields p(data point)
- Note that these computations are all linear in the number of parameters

EDPL: Non-uniform reward values

- Parameter settings rewarded based on relevance to data point
- **Non-essential** parameter settings will have $p(\psi_i|\text{data point}) \approx p(\boldsymbol{\psi_i})$, meaning virtually no change in grammar
  - Data point will be generated correctly equally often under either setting of the parameter
- **Essential** parameter settings will have $R_d > p(\boldsymbol{\psi_i})$: increase in $p(\boldsymbol{\psi_i})$
  - Desired parameter setting will generate data point correctly more often than opposite setting of the same parameter
  - The more evidence the learner can find, the larger the increase in $p(\boldsymbol{\psi_i})$

Simulations and Results

Dresher and Kaye's parameters

- 10 binary parameters on foot placement, quantity sensitivity, and other aspects of metrical phonology
  - Foot placement (4 parameters):
    - Foot Headedness (L/R), Foot boundedness (Y/N), Foot heads have secondary stress (Y/N), Foot heads (may/may not) be light syllables
  - Quantity Sensitivity (Y/N)
  - Extrametricality: Presence (Y/N) and Edge (L/R)
  - Syllable representation (Are CVC syllables Heavy?)
  - Footing direction (L-to-R/R-to-L)
  - Main/secondary stress placement (L/R)

Stress Typology Test Set

- 23 Languages in Dresher and Kaye's system (1990)
  - Focus on 7 out of 10 parameters:
    - All parameters on foot properties, quantity-sensitivity, and extrametricality
  - Constructed languages
    - However, many patterns correspond to real languages (16 out of 23)
- Forms on which stress patterns are presented:
  - All possible 3 to 6-syllable combinations of [ta], [taa], and [tan]

Learning Set-up

- All 23 languages given to NPL and EDPL
- 10 runs for every language
- Simulations run for 1,000,000 iterations or till convergence
  - Convergence: every word's stress generated correctly 99 out of 100 times (checked every 100 iterations)
- Evaluation
  - Did learner converge on a correct grammar?
  - How quickly?

NPL vs. EDPL

- NPL failed to converge for all but one language
  - Overall success rate: **4.3%**
  - within **89,370** iterations on average
- EDPL showed convergence for all languages
  - Overall success rate: **96.0%**
  - within **200** iterations on average

NPL vs. EDPL

- NPL failed to converge for all but one language
  - Overall success rate: **4.3%**
    - One language: initial stress, no secondary stress (cf. Hungarian)
      - Compatible with more ψ combinations than all other patterns
  - within **89,370** iterations on average
- EDPL showed convergence for all languages
  - Overall success rate: **96.0%**
  - within **200** iterations on average

NPL vs. EDPL

- NPL failed to converge for all but one language
  - Overall success rate: **4.3%**
  - within **89,370** iterations on average
- EDPL showed convergence for all languages
  - Overall success rate: **96.0%**
    - All but one language (1/10 runs): particularly ambiguous (see appendix)
    - Faster for languages that have ”signature” (disambiguating) forms
      - Similar to “cues” approach (Dresher and Kaye 1990)
  - within **200** iterations on average

Discussion and conclusion

Summary: NPL vs. EDPL

- EDPL retains advantages of NPL:
  - Gradual update of stochastic grammar; computed in linear time
- At the same time, EDPL has higher power of inference than NPL:
  - Able to distinguish relevant from irrelevant parameters given a data point
  - Leads to success on representative typology

Summary: NPL vs. EDPL

- EDPL retains advantages of NPL:
  - Gradual update of stochastic grammar; computed in linear time
- At the same time, EDPL has higher power of inference than NPL:
  - Able to distinguish relevant from irrelevant parameters given a data point
  - Leads to success on representative typology
- EDPL provides mechanism for identifying unambiguous data (by gauging individual parameter settings' success on a data point)
  - Takes over the function of Dresher & Kaye's cues

Summary: NPL vs. EDPL

- EDPL retains advantages of NPL:
  - Gradual update of stochastic grammar; computed in linear time
- At the same time, EDPL has higher power of inference than NPL:
  - Able to distinguish relevant from irrelevant parameters given a data point
  - Leads to success on representative typology
- EDPL also indirectly provides mechanism for parameter ordering:
  - Some ψs (extrametricality, QS) can be found when the rest of structure (foot headedness, foot boundaries) is not yet (completely) established

Implications for domain-specificity

- Pearl (2007) argues: statistical P & P learning has to be supplemented by domain-specific mechanisms because it fails
- We argue: NPL's failure is not representative of statistical learning in general
  - We have introduced a model, EDPL, that performs well on stress parameter setting
- Therefore: the necessity of domain-specific mechanisms for stress parameters is under question
- Future work: understanding to what extent EDPL duplicates the effect of domain-specific mechanisms (cues and their ordering)

Thank you!

Acknowledgments

- We would like to thank Joe Pater, Kristine Yu, and the members of the UMass Amherst Sound Workshop for their comments and for very useful discussion

References

Berwick, Robert C., and Partha Niyogi. 1996. Learning from Triggers. *Linguistic Inquiry* 27(4): 605-622.

Bush, Robert, and Frederick Mosteller. 1951. A mathematical model for simple learning. *Psychological Review* 58, 313–323.

Chomsky, Noam. 1981. *Lectures on government and binding*. Dordrecht: Foris.

Dresher, B. Elan, and Jonathan D. Kaye. 1990. A computational learning model for metrical phonology. *Cognition* 34: 137-195.

Fodor, Janet D. 1998. Parsing to Learn. *Journal of Psycholinguistic Research* 27(3), 339-374.

Gibson, Edward, and Kenneth Wexler. 1994. Triggers. *Linguistic Inquiry* 25(3): 407-454.

Jarosz, Gaja. Submitted. *Expectation Driven Learning of Phonology*.

Lightfoot, D. 1999. *The Development of Language: Acquisition, Change, and Evolution*. Oxford: Blackwell.

Pearl, Lisa. 2007. *Necessary Bias in Natural Langauge Learning*. Doctoral dissertation, University of Maryland.

Pearl, Lisa. 2011. When Unbiased Probabilistic Learning is Not Enough: Acquiring a Parametric System of Metrical Phonology. *Language Acquisition* 18(2): 87-120.

Sakas, William G., and Fodor, Janet D. 2001. The structural triggers learner. In Stefano Bertolo (ed.) *Language Acquisition and Learnability*, Cambridge University Press, Cambridge, UK.

Yang, Charles. 2002. *Knowledge and Learning in Natural Language*. Oxford: Oxford University Press.