Learning within- and between-word variation in probabilistic OT grammars

Aleksei Nazarov, University of Huddersfield

AMP 2017, New York University a.nazarov@hud.ac.uk http://alekseinazarov.org

Types of variation

Within-word	**Between-word**	**Within- and between-word**
X → Y ~ Z / α_β	X_WORD1 → Y / α_β X_WORD2 → Z / α_β	X_WORD1 → Y ~ Z / α_β X_WORD2 → Z / α_β
English nasal place assimilation [4]	*Dutch primary stress [6]*	*Hebrew spirantization [13]*
/n/ → [n ~ m] / _P	CVCV_WORD1 → CVCV́ / _CV# CVCV_WORD2 → CV́CV / _CV#	/p,/ /b/, /k/ → [p ~ f], [b ~ v], [k ~ χ] / V_ /p,/ /b/, /k/ → [p], [b], [k] / V_ /p,/ /b/, /k/ → [f], [v], [χ] / V_
/gɹin bɑks/ → [gɹ**in** bɑks] ~ [gɹi**m** bɑks] /ɪn bɛd/ → [ɪ**n** bɛd] ~ [ɪ**m** bɛd]	/kasino/ → [**ka.ˈsi**.no] /kimono/ → [ˈ**ki.mo**.no]	/me**k**ase/ → [me**k**ase ~ meχase] /da**k**ar/ → [da**k**ar] *χ /ma**k**ar/ → [maχar] ***k**
<u>Requires probabilities</u>	<u>Requires diacritics in UR</u>	<u>Requires probabilities and diacritics</u>

Question: **How can OT learners infer between-word diacritics in the face of within-word variation?**

Current implementation

- Model of between-word variation: Indexed Constraint Theory [8,9]
  - Constraints may have several instantiations: one general version, and versions with various diacritics (indices)

/akta/	$Ident_i$	*Stop/V_	Ident
[akta]		*!	
[axta]			*
/akpa$_i$/			
[akpa]		*	
[axpa]	*!		*

- Soft Inconsistency with threshold x = 0.1
  - Compares entire lexicon vs. individual words, not each word pair
- At most one indexed constraint inferred after each pass through the data:
  - Find all constraint pairs exhibiting Soft Inconsistency
  - Among these, take the one with the highest absolute divergence of p(Cns1 >> Cns2) between exceptions and lexicon

Previous work: probability vs. inconsistency

Probabilistic models: degrees of match

- Stoch. OT [1], EDL [7], MaxEnt [5], Noisy HG [4], …:
  - One input may yield different outputs at different times
- Learning within-word variation in GLA [1], EDL [5], …:
  - For any datum consistent only with Cns1 >> Cns2: p(Cns1 >> Cns2) goes up
  - If some data points consistent only with Cns1 >> Cns2, other data points only with Cns2 >> Cns1:
    - <u>No inconsistency</u>
    - p(Cns1>> Cns2) is between 1 and 0 (which yields within-word variation)
- **No proposed mechanism for inducing between-word diacritics**

Categorical models: (in)consistency

- Standard OT [12]:
  - One input may only ever yield one output
- Learning diacritics in (R)CD, [14]:
  - If all data consistent with Cns1 >> Cns2: set Cns1 >> Cns2
  - If some data points consistent only with Cns2 >> Cns1, other data points only with Cns2 >> Cns1:
    - <u>Inconsistency detected</u>
  - [2,3,10]: Whenever inconsistency is detected, infer a diacritic for representing between-word variation
- **No proposed mechanism for representing within-word variation**

Test: Hebrew spirantization

- Data: simplified Hebrew spirantization [13] (non-postvocalic within-word variation leveled)

postvocalic underlying stops		**non-postvocalic underlying stops**	
/mekase/	→ [me**k**ase ~ meχase]	/linpoʃ/	→ [lin**p**oʃ]
/ʃabar/	→ [ʃa**b**ar ~ ʃavar]	/lisbol/	→ [lis**b**ol]
/dakar/	→ [da**k**ar]	/liʃkoa/	→ [liʃ**k**oa]
/mebarer/	→ [mevarer]	/liʃpoχ/	→ [liʃ**p**oχ]
/mebatel/	→ [mevatel]	/lizkot/	→ [liz**k**ot]
/gaba/	→ [gava]		

- Constraints: *Stop/V_ || *Stop || *NonSibilantFricative || Ident || Max
- Candidates:

postvocalic	**non-postvocalic**
/mekase/ [mekase, meχase, mkase, mχase]	/linpoʃ/ [linpoʃ, linfoʃ, lipoʃ, lifoʃ]

- 20 runs of up to 80 iterations
  - ≥95% match to dataset at all runs within 9–14 iterations (average: 11.35)
  - Learning performance, averaged between runs:

		Appropriate range of variation predicted in wug test	**Diacritic assigned**
Non-exceptions	**Post-V**	No deletion, variable spirantization: 96%	0%
	Post-C	No variation: 95%	3%
Exceptions	**Post-V**	–	91% (/b/ → [v]: 100% /k/ → [k]: 65%)

- Future work: match frequencies of within-word and between-word variants; consider larger data sets; compare to results obtained with UR stress learning in [9]

Proposal: Soft Inconsistency

- Insights from both types of learners combined:
  - **Inconsistency** (see above):
    p(Cns1 >> Cns2) = 1 for *WORD1* & p(Cns1 >> Cns2) = 0 for *WORD2*
  - **Soft Inconsistency** (= Inconsistency x probability):
    p(Cns1 >> Cns2) > [0.5 + x] for *WORD1* & p(Cns1 >> Cns2) < [0.5 - x] for *WORD2*
- Whenever the learner encounters Soft Inconsistency, it has evidence for a diacritic
- Requirements for using Soft Inconsistency:
  - Access to **pairwise ranking probability estimates per individual word**
  - Access to **information across the lexicon**
- Could be implemented in various frameworks, but the most straightforward is **Expectation Driven Learning** (EDL [7]; see extra information to the right)
  - Computes p(Cns1 >> Cns2) for individual words as well as the entire lexicon
  - Batch learning available: access to information across the lexicon

Expectation Driven Learning

- Grammar: probability distribution over pairwise rankings
  - Every time, different full ranking, e.g. Cns1 >> Cns2 >> Cns3, generated with special sampling procedure (see [7])
- For input *d* and{Cns1, Cns2}, P($match_d$ | Cns1 >> Cns2): within sample *S*, the proportion of cases where some attested output for input *d* wins
- To construct *S:* set p(Cns1 >> Cns2) = 1 in current grammar and generate an output for *d* multiple times (in this case, 50 times)

$$p(Cns1>>Cns2)_{new} = \frac{p(match_d \mid Cns1>>Cns2) * p(Cns1 >> Cns2)_{old}}{p(match_d)}$$

References

[1] Boersma, Paul. 1998. "Functional Phonology: Formalizing the Interactions between Articulatory and Perceptual Drives." PhD dissertation, U of Amsterdam. • [2] Becker, M. 2009. "Phonological Trends in the Lexicon: The Role of Constraints." PhD dissertation, UMass Amherst, Amherst. • [3] Coetzee, Andries W. 2009. "Learning Lexical Indexation." *Phonology* 26 (1): 109–45. • [4] Coetzee, Andries W, and Joe Pater. 2011. "The Place of Variation in Phonological Theory." In: J.A. Goldsmith, J, Riggle, and A.C. Yu (eds.), *Handbook of Phonological Theory*, 2nd ed., 401–34. Wiley-Blackwell. • [5] Goldwater, S., and M. Johnson. 2003. "Learning OT Constraint Rankings Using a Maximum Entropy Model." In: J. Spenader, A. Eriksson, and Ö. Dahl (eds.), Proceedings of the Stockholm Workshop on Variation within Optimality Theory, 111–20. • [6] van der Hulst, H.G. 1984. Syllable structure and stress in Dutch. Dordrecht: Foris. • [7] Jarosz, G. 2015. "Expectation Driven Learning of Phonology" Ms., UMass Amherst. • [8] Kraska–Szlenk, I. 1995. "The Phonology of Stress in Polish." PhD dissertation, U of Illinois, Urbana-Champaign. • [9] Moore-Cantwell, C. 2017. "Concurrent learning of the lexicon and phonology." Talk given at the LSA 2017 Annual Meeting. • [10] Pater, J. 2000. "Non-Uniformity in English Secondary Stress: The Role of Ranked and Lexically Specific Constraints." *Phonology* 17 (2): 237–74. • [11] Pater, J. 2010. "Morpheme-Specific Phonology: Constraint Indexation and Inconsistency Resolution." In: S. Parker (ed.), *Phonological Argumentation: Essays on Evidence and Motivation*, 123–54. London: Equinox Press. • [12] Prince, A.S., and P. Smolensky. 1993/2004. Optimality Theory: Constraint Interaction in Generative Grammar. Malden, MA: Blackwell. • [13] Temkin–Martínez, M. 2010. "Sources of Non-Conformity in Phonology: Variation and exceptionality in Modern Hebrew Spirantization." PhD dissertation, USC. • [14] Tesar, B. 1995. "Computational Optimality Theory." PhD dissertation, U of Colorado.

Acknowledgments

Many thanks to Adam Albright, Paul Boersma, Silke Hamann, Gaja Jarosz, John McCarthy, Marc van Oostendorp, Joe Pater, Robert Staubs, Kristine Yu, and audiences at UMass Amherst, the University of Manchester, the University of Amsterdam, MIT, and the University of Connecticut for very helpful and stimulating discussion of various instantiations of this work.