Learning exception-ful grammars: within- and between-word variation[*]

Aleksei Nazarov, Harvard University
anazarov@fas.harvard.edu
Phonology Reading Group, University of Connecticut
April 21, 2017

0. Overview

- In OT, exceptionality (between-word variation) represented as different constraint rankings for different lexical items (Kraska-Szlenk 1995, Pater 2000)

- Modern Hebrew (Temkin-Martínez 2010): example of within-word variation in addition to between-word variation
  - Moreover, between-word variation orthogonal to within-word variation
    - Words that have variants are not exceptional, and *vice versa*

- Existing learners for phonological exceptions are categorical (=non-probabilistic):
  - Induce exceptional constraint when two words have inconsistent ranking conditions (Becker 2009, Coetzee 2009, Pater 2010)
  - Inconsistent ranking conditions occur both for within- and between-word variation: cannot handle Hebrew!

- Proposal: probabilistic inconsistency in Expectation Driven Learning (EDL; Jarosz 2015)
  - EDL learns distributions over pairwise rankings:
    - P(A>>B), P(A>>C), P(B>>C)
  - Allows to define "soft inconsistency"
    - Assume exceptionality if P(A>>B) has opposite values for lexicon and a specific word

- Tested on simplified version of Hebrew dataset:
  - Learner distinguishes between-word from within-word variation:
    - Words that have within-word variation are recognized as being non-exceptions
    - Words that do not have within-word variation are recognized as outliers, and assigned to indexed constraints
    - Indexed constraints are given a high enough ranking to be able to account for the exceptional patterns

[*] I would like to thank Adam Albright, Paul Boersma, Silke Hamann, Gaja Jarosz, Claire Moore-Cantwell, Erin Olson, Marc van Oostendorp, Joe Pater, Juliet Stanton, Kristine Yu, and participants of the PhonoLAM at the University of Amsterdam and the Phon Circle at MIT for very helpful discussion of this project at various stages. All errors remain my own.

1. Exceptions and indexed constraints

1.1 Definition

- Between-word variation (exceptionality):
  - Phonological process applies to proper subset of the lexicon (i.e., not all words)
  - Undergoers vs. non-undergoers not determined by symbols of purely phonological representation:
    - Not by underlying representation (e.g., lexical stress in Russian, cf. Melvold 1989, Revithiadou 1999; see Moore-Cantwell 2017 on learning of underlying representations of various strengths)

      /múk-u/ → [ˈmuku] ‘torment (acc.)’
      /muk´-u/ → [muˈku] ‘flour (acc.)’

    - Not by intermediate representation (e.g., opacity in Canadian English, Joos 1942, Chomsky 1964:73-74)

      /ɹaɪt-ɚ/ → ɹʌɪtɚ → [ɹʌɪɾɚ] “writer”
      /ɹaɪd-ɚ/ → ɹaɪdɚ → [ɹaɪɾɚ] “rider”

    - Not by surface representation (e.g., phonotactic blocking or triggering – as in English, Bakovic 2011)

      /reɪz-z/ → reɪzɨz, *reɪzz ‘raises’
      /reɪv-z/ → *reɪvɨz, reɪvz ‘raves’

1.2 English secondary stress

- Example: English secondary stress (Pater 2000)
  - In majority of lexicon, single heavy syllable before main stress receives secondary stress

(1)
a. *Single light pretonic syllable*
ba.ná.na
ce.rá.mic
go.rí.lla

b. *Single heavy pretonic syllable*
àd.méa.sure
bàn.dá.na
Òc.tó.ber
cì.ta.tion

- Minority pattern: unstressed single heavy pretonic syllables.

(2)
ăd.ván.tage
cŏn.fér
ĕ[k.s]cúr.sion

- Cannot be attributed to any phonological representation
  - Underlying: Only a local "lack of stress" mark on *ad-*, *con-*, *ex-* could generate exceptional pattern; generally not assumed to exist
  - Intermediate: No known opaque interactions
  - Surface: No surface factors (segmental, metrical) can predict this contrast

- Side note: Kim and Pulleyblank (2009) and others suggest that all exceptionality can be reanalyzed as faithfulness to underlying forms
  - English secondary stress (Pater 2000 – see above), Guahibo stress (Mullin 2011), and possibly Ukrainian stress (Osadcha 2014) provide counterexamples

1.3 Exceptionality in OT: lexical indexation

- In OT: if exceptional words cannot be represented with different underlying representations, they must be subject to a different ranking than non-exceptional words[1]

(3)
a. *Ranking for typical English secondary stress (Pater 2000)*

bandana	Parse-syll	Weight-to-Stress	*Clash-Head
b[n̩] (dá.na)	*!		
☞ (bàn) (dá.na)			*
Alexander			
☞ (À.le[g]) ([z]án) der	*	*	
Ă (lè[g]) ([z]án) der	**!		*
Timbuctoo			
(Tìm.bŭc) (tóo)		*!	
☞ (Tìm) (bùc) (tóo)			*

b. *Ranking for exceptional lack of stress on heavy pretonics*

advantage	*Clash-Head	Parse-syll	Weight-to-Stress
☞ ăd (ván) tage		**	**
(àd) (ván) tage	*!	*	*

[1] Besides lexically indexed constraint theory, discussed below, theories that allows for separate rankings/weightings per word include Cophonology Theory (Itô and Mester 1995, Orgun 1996, Inkelas 1998, Inkelas and Orgun 1995, 1998, and Inkelas and Zoll 2007) and Sublexical Phonology (Linzen, Kasyanenko, and Gouskova 2013, Gouskova, Newlin-Łukowicz, and Kasyanenko 2015, Becker and Gouskova 2016).

- One theory that allows for multiple rankings: lexically indexed constraint theory (Kraska-Szlenk 1995, Pater 2000)
    - There is one grammar
    - However, some constraints have indexed variants
        - *Clash-Head is violated for any surface form that has clash with main stress
        - *Clash-Head*i* is violated for any surface form that has clash with main stress <u>if it derives from an input with index *i*</u>

(4) *Violations for lexicon-wide and indexed versions of *Clash-Head*

/advantage *i* /	*Clash-Head	*Clash-Head*i*
ăd (ván.tage)		
(àd) (ván.tage)	*	*
/admeasure/		
ăd (méa.sure)		
(àd) (méa.sure)	*	

- By ranking indexed version higher than lexicon-wide version, words with index *i* may start behaving exceptionally
    - In English case, ranking *Clash-Head*<u>i</u>* above Parse-syll gets the exception effect

(5)
a. *Exceptional* advantage: *no pretonic stress*

/advantage *i* /	*Clash-Head*i*	Parse-syll	Weight-to-Stress	*Clash-Head
☞ ăd (ván.tage)		*	*	
(àd) (ván.tage)	*!			*

b. *Non-exceptional words: default pattern*

/admeasure/	*Clash-Head*i*	Parse-syll	W-to-Stress	*Clash-Head
ăd (méa.sure)		*!	*	
☞ (àd) (méa.sure)				*
Alexander				
☞ (À.le[g]) ([z]án) der		*	*	
Ă (lè[g]) ([z]án) der		**!		*
Timbuctoo				
(Tìm.bŭc) (tóo)			*!	
☞ (Tìm) (bùc) (tóo)				*

- Usually, an indexed constraint is ranked above its lexicon-wide variant (e.g., *Clash-Head*i* >> *Clash-Head):
    - *Clash-Head*i* is violated in a subset of all cases in which *Clash-Head is violated
    - Because of stringency, *Clash-Head >> *Clash-Head*i* has little effect
    - (although see Prince 1997 for cases in which rankings of the type General >> Specific can have an effect)

2. Within- and between-word variation in Modern Hebrew

- In addition to between-word variation, phonological systems can also contain within-word variation (Coetzee and Pater 2011)
- Within-word and between-word variation both observed in Modern Hebrew spirantization (Adam 2002, Temkin-Martínez 2010)

2.1 Spirantization

- Spirantization process, formalized in (6), exemplified in (7):
  - Stops /p, b, k/ realized as [f], [v], [χ] in word-medial post-vocalic position
    - [p], [b], [k] word-initially or in word-medial post-consonantal position
  - Coronals and word-final consonants are protected from alternation
  - *[ɣ] is disallowed

(6) *Spirantization rule*

$$\begin{bmatrix} -son \\ -cont \\ -cor \\ \begin{Bmatrix} -dors \\ +dors, -voi \end{Bmatrix} \end{bmatrix} \rightarrow [+cont] \;/\; \text{V}\_\ldots\#$$

  - Alternation is seen when the same consonantal root occurs with different vowel patterns.

(7) *Spirantization in postvocalic position (Temkin-Martínez 2010:23, 196)*

		Non-PostV	PostV
/p/	/**p**rs/	[**p**aras] ‘spread, 3sg past’	[li**f**ros] ‘spread, inf.’
/b/	/s**b**l/	[lis**b**ol] ‘suffer, inf.’	[sa**v**al] ‘suffer, 3sg past’
/k/	/**k**tv/	[**k**atav], *[**k**asav] ‘write, 3sg past’	[li**χ**tov] ‘write, inf.’
/g/	/p**g**ʃ/	[lif**g**oʃ] ‘to meet, inf.’	[pa**g**aʃ], *[pa**ɣ**aʃ] ‘meet, 3sg past’

2.2 Within-word variation

- Spirantization may apply variably within the same word:
  - Spirantization may optionallty overapply in non-postV position, as in (8)
    - Non-spirantization still preferred
  - Spirantization may optionally underapply in postV position, as in (9)
    - Spirantization still preferred

(8) *Within-word variation in non-postV position (Temkin-Martínez 2010:33)*

/**p**zr/ [**p**izer ≻ **f**izer] ‘scattered’
/k**b**r/ [jik**b**or ≻ jik**v**or] ‘will bury’
/**k**sh/ [**k**isa ≻ **χ**isa] ‘covered’

(9) *Variation in postV position (Temkin-Martínez 2010:33)*
/**k**bs/ [jeχabes ≻ je**k**abes] 'will launder'
/**k**sh/ [jeχase ≻ je**k**ase] 'will cover'

2.3 Between-word variation

- Individual words can deviate from the pattern described in 2.2[2]:
  - Some words categorically disallow underapplication in postV position, as in (10)
  - Some words categorically disallow spirantization, as in (11)

- Words that share the same root may or may not all be exceptional:
  - Compare (10a) with (10b), and (11ab) with (11dce)

(10) *Words that block underapplication*

	Non-PostV	PostV	
a. /m**k**r/	[lim**k**or ≻ limχor]	[maχar], *[ma**k**ar]	*no underapplication in /makar/*
b. /**p**tr/	[**p**iter], *[**f**iter]	[me**f**ater], *[me**p**ater]	*no underapplication in /mefater/*

(11) *Words that block spirantization*

	Non-PostV	PostV	
a. /**k**rʔ/	[**k**ara], *[χara]	[li**k**ro], *[liχro]	
b. /d**k**r/	[lid**k**or], *[lidχor]	[da**k**ar], *[daχar]	
c. /ʃ**k**ʔ/	[liʃ**k**oa ≻ liʃχoa]	[ʃa**k**a], *[ʃaχa]	*no spirantization in /ʃaka/*
d. /**k**tv/	[**k**atav], *[χatav]	[liχtov ≻ li**k**tov]	*no spirantization in /katav/*
e. /**p**tr/	[**p**iter], *[**f**iter]	[me**f**ater], *[me**p**ater]	*no spirantization in /piter/*

- In addition, there are words in which only one consonant behaves exceptionally
  - Temkin-Martínez proposes indexation of constraints to individual segments

(12) *Words in which spirantization is blocked only for /k/*

/**bk**r/	[**b**i**k**er ≻ **v**i**k**er], *χ	[me**v**a**k**er ≻ me**b**a**k**er], *χ

- Essential aspects of data:
  - Variable spirantization process:
    - preference for spirantization in postV position
    - preference for non-spirantization in non-postV position
  - Some words have obligatory spirantization in postV position
  - Some words are barred from undergoing spirantization

[2] Data from Temkin-Martínez 2010:194-8; variants that occur less than 10% of the time are interpreted as ungrammatical, which is in line with Temkin-Martínez' presentation of the facts.

2.4 Why this is exceptionality

- As argued in section 1.1, true between-word variation should not be conditioned by underlying representations
    - Data so far do not exclude analysis where:

      [p] is underlyingly /p/
      [f] is underlyingly /f/, and
      [p ~ f] is underlying underspecified for continuancy: /P/

- However, there are also roots/words, usually represented with underlying fricatives, that follow a different pattern

    - **Underlying fricative pattern 1**: preference for [f, v, χ] over [p, b, k] both in non-postV position

(13) *Preference for fricatives in non-postV position*
/**v**dh/ [**v**ida ≻ **b**ida]
[me**v**ade ≻ me**b**ade]

- - **Underlying fricative pattern 2**: categorical [f, v, χ] in non-postV position

(14) *Categorical requirement of fricatives in non-postV position*
/χjχ/ [**χ**ijeχ], *[**k**ijeχ]
[me**χ**ajeχ], *[me**k**ajeχ]

- If differences in spirantization derived from underlying representations only, we would have the following four patterns of within-word variation to account for:

(15) *Attested preference patterns for fricatives and stops*

Pattern	Example, non-postV	Example, postV	Underspec. account
All stops	[**k**ara], *[**χ**ara]	[ʃa**k**a], *[ʃa**χ**a]	[-continuant]
Preferentially stops	[**k**isa ≻ **χ**isa] [**b**ana ≻ **v**ana]	–	Underspecified?
Preferentially fricatives	[**v**ida ≻ **b**ida]	[li**v**not ≻ li**b**not] [je**χ**ase ≻ je**k**ase]	Underspecified?
All fricatives	[**χ**ijeχ], *[**k**ijeχ]	[ma**χ**ar], *[ma**k**ar]	[+continuant]

- This provides evidence that spirantization undergoes both within-word and between-word variation
    - As can be seen in (15), the "preferentially stops" and "preferentially fricatives" patterns cannot come from the same underlying category, since they both occur in non-postV position
        - Perhaps Moore-Cantwell's (2017) strength of underlying representation approach might be able to salvage a non-exceptional analysis of these data

  - In addition, if we consider that words with the same root should have the same underlying form, the data in (10-11) also shows that variable application of spirantization is not exclusively conditioned by underlying representations

- Simulations in section 5, only words with underlying /p, b, k/ (non-shaded cells in (15))
  - All differences in application of spirantization due to within-word or between-word variation – underlying stop/fricative difference not relevant

3. Learning exception-ful grammars

- Learning indexed constraint grammars in OT has two subproblems:
  - Find ranking (weighting) of indexed and lexicon-wide constraints
    - Not significantly different from other ranking problems
  - Find which indexed constraints and which indices are actually needed
    - Requires inference mechanisms that compare preferences of lexical items

- Existing inference mechanism (Becker 2009, Coetzee 2009, Pater 2010) based on concept of **inconsistency**:
  - Whenever distinct groups of words necessitate contradictory rankings (e.g., *Clash-Head >> Weight-to-Stress* vs. *Weight-to-Stress >> Clash-Head*)
    - Lexically indexed constraint inferred to distinguish both groups

- Problem: inconsistency as defined in Constraint Demotion approach cannot distinguish within-word from between-word variation

3.1 Learning from inconsistency

- Existing indexed constraint learners (Becker 2009, Coetzee 2009, Pater 2010):
  - Rank constraints using Constraint Demotion
  - Whenever inconsistency (see below) is detected, induce an indexed constraint

- Constraint Demotion (Tesar 1995 et seq.) finds a ranking consistent with an entire data set:
  - Start with no constraints in the grammar
  - Find all the winner-loser pairs in the data
    - E.g., àlexánder ~ alèxander
      àlexánder ~ alexánder
      àlexánder ~ aléxander

  - For each constraint and each pair, assess whether this constraint has fewer violations for the winning candidate (W) or for the losing candidate (L)

  - Repeat until all constraints are in the grammar:
    - Find all constraints that have no L marks in them, and add them into the grammar under any constraints that were previously added
    - Remove all winner-loser pairs that had W marks for those constraints

(16) *Comparative tableau for Constraint Demotion (cf. (5b))*

Winner~Woser	Parse-syll	W-to-Stress	*Clash-Head
(àd) (méa.sure) ~ *ăd (méa.sure)	W	W	L
(À.le[g]) ([z]án) der ~ *Ă (lè[g]) ([z]án) der	W	L	W
(Tìm) (bùc) (tóo) ~ *(Tìm.bŭc) (tóo)		W	L

- Inconsistency: there is no single ranking that can account for all data points (winner-loser pairs)
    - All constraints vying for admission to grammar have a Loser mark in them

(17) *Both* àdméasure *and* ădvántage *in data set: inconsistency*

Winner~Loser	Parse-syll	W-to-Stress	*Clash-Head
(àd) (méa.sure) ~ *ăd (méa.sure)	W	W	L
(À.le[g]) ([z]án) der ~ *Ă (lè[g]) ([z]án) der	W	L	W
(Tìm) (bùc) (tóo) ~ *(Tìm.bŭc) (tóo)		W	L
ăd (ván) tage ~ (àd) (ván) tage	L	L	W

- Whenever inconsistency detected, induce indexed constraint to resolve it (see Becker 2009 for detailed proposal)
    - In (18) below, after adding *Clash-Head*i* to the grammar and deleting the last winner-loser pair, we obtain (16), which contains no inconsistency.

(18) *Resolve inconsistency by inserting indexed constraint*

Winner~Loser	Parse-syll	W-to-Stress	*Clash-Head	*Clash-Head*i*
(àd) (méa.sure) ~ *ăd (méa.sure)	W	W	L	
(À.le[g]) ([z]án) der ~ *Ă (lè[g]) ([z]án) der	W	L	W	
(Tìm) (bùc) (tóo) ~ *(Tìm.bŭc) (tóo)		W	L	
ăd (ván) tage *i* ~ (àd) (ván) tage *i*	L	L	W	W

3.2 Problems with categorical learners

- Constraint Demotion is purely categorical: finds a constraint ranking consistent with all data
  - Between-word variation leads to opposite ranking requirements
  - Problem: within-word variation also leads to opposite ranking requirements – (19)

(19) *Inconsistency in the case of variation between* àb.dó.mi.nal *and* ăb.dó.mi.nal

Winner~Loser	*Clash-Head	WSP
àb.dó.mi.nal ~ ăb.dó.mi.nal	L	W
ăb.dó.mi.nal ~ àb.dó.mi.nal	W	L

- If both within-word and between-word variation trigger inconsistency:
  - Indexed constraints will be induced both for words that deviate from the trend, and for words that exhibit within-word variation
  - This is problematic for Hebrew:
    - Non-exceptional words exhibit within-word variation
/**k**sh/ [**k**isa ≻ **χ**isa] (non-postV – preference for no spirantization)
[je**χ**ase ≻ je**k**ase] (postV – preference for spirantization)
    - Should not be connected to indexed constraints
    - Exceptional words do not exhibit within-word variation
/**p**tr/ [**p**iter], *[**f**iter] (no spirantization allowed)
[me**f**ater], *[me**p**ater] (no underapplication allowed)
    - Should be connected to indexed constraints

4. Proposal: "soft inconsistency" from pairwise ranking probabilities

- In order to capture within-word variation, categorical ranking assumption must be relaxed
  - Constraints may be given real-number weights (e.g., Boersma 1998, Goldwater and Johnson 2003) to be able to estimate a probability of each output given an output
  - Constraints may be given a partial (Anttila 2002) or probabilistic (Jarosz 2006, 2015) ranking
- If provided with lexically indexed constraints, such models can capture both within-word and between-word variation
  - Temkin-Martínez (2010) shows that Hebrew data can be captured in a Stochastic OT learned by the GLA (Boersma 1998), given that the following are provided beforehand:

        - An indexed constraint Ident-cont*i*
        - All the items (words and segments) that have index *i*
    - See also Moore-Cantwell and Pater (2016) for the modeling the effects of lexical patterns on productivity via lexically indexed constraints in Maximum Entropy grammar (Goldwater and Johnson 2003 and related literature)

- However, these models are not obviously compatible with the model of exception induction in Becker (2009), Coetzee (2009), and Pater (2010)
    - Inconsistency not defined in numerical ranking learners like the GLA (Boersma 1998) or the Harmonic Grammar family (Goldwater and Johnson 2003, Potts et al. 2010)
        - Taking away inconsistency is what makes within-word variation possible

- Moore-Cantwell (2017): Maximum Entropy model in which strength of underlying representations leads to gradient productivity effects
    - Highly relevant to Hebrew problem, might be able to explain data in section 2.4
    - Here: approach in terms of lexically indexed constraints

- Proposal:
    - Start with Jarosz's (2015) learner that finds probabilities of pairwise rankings

      P(*Weight-to-Stress* >> *Parse-syll*)
      P(**Clash-head* >> *Weight-to-Stress*)
      P(**Clash-head* >> *Parse-syll*)

    - Define **"soft inconsistency"**
        - The procedure finds "soft inconsistency" for a constraint pair {A,B} if the lexicon has P(A >> B) > 0.5, but some words have P(A >> B) < 0.5

    - Define criterion for inferring lexical constraints
        - Induce an indexed constraint whenever inconsistency is detected
            - If lexicon prefers A >> B, induce B*i* for the trend-defying words
        - (Some relaxation of this criterion due to probabilistic nature of learner)

4.1 Pairwise ranking probabilities

- Jarosz (2015) starts with assumption that grammar consists of binomial probabilities over pairwise rankings of OT constraints

(20) *Example of a pairwise ranking grammar*

Ranking	Probability
A >> B	0.7
A >> C	0.5
B >> C	0.2

- Every time the grammar is used, a full categorical ranking is sampled from these probabilities
    - Cannot be done with **independent** coin tosses for each constraint pair;
      e.g., 0.7 x 0.5 x 0.2 = 0.07 chance of picking the incoherent ranking
      A >> **B**
      C >> A
      B >> C
    - Instead: **dependent** coin tosses (see Jarosz 2015 for details)
        - Create <u>random order</u> of constraint pairs
          e.g., < AC, BC, AB >
        - Go through this random order
            - Whenever this constraint pair still has a ranking probability between 0 and 1:
                - Select a pairwise ranking by weighted coin toss
            - Whenever a ranking for another pair is logically entailed by the ranking so far (e.g., C >> A and B >> C entails B >> A):
                - <u>Select entailed rankings without consulting their probability</u>

start:	P(A>>C) = 0.5	P(B>>C) = 0.2	P(A>>B) = 0.7
st. 1	**C >> A**	P(B>>C) = 0.2	P(A>>B) = 0.7
		no entailments	
st. 2	**C >> A**	**B >> C**	P(A>>B) = 0.7
		entailment:	**B >> A**

resulting ranking: B >> C >> A

- This is a useful approximation to storing probabilities over entire rankings (Jarosz 2006):
  e.g., P(A >> B >> C) = 0.05
  P(A >> C >> B) = 0.01
  P(B >> A >> C) = ...
  etc.

- Advantage of pairwise rankings: manageable space of hypotheses
    - If 10 constraints:
        - $10! = 3{,}628{,}800$ full-ranking hypotheses
        - $\binom{10}{2} \times 2 = 90$ pairwise-ranking hypotheses

4.2 Learning pairwise probabilities

- Expectation Driven Learning (EDL, Jarosz 2015): special Expectation Maximization (EM) algorithm that iteratively estimates P(A >> B) for each constraint pair
    - Starts with an initial hypothesis as to pairwise ranking probabilities
        - Here, initial hypothesis is P(A >> B) = 0.5 for all constraint pairs

  - Takes samples from these probabilities
    - Computes new P(A >> B) based on expectations from old P(A >> B)
  - As an EM algorithm, guaranteed to find a (local) maximum of data likelihood (Wu 1983)
    - Stopped when data likelihood has reached 0.95 ("correct convergence"), or after the maximum number of iterations

- Computing new P(A>>B):
  - For each /input/:
    - Take *r* samples from current grammar such that A >> B is always selected
      - Number of times attested candidate for /input/ wins = $\text{Success}_{A >> B}$
    - Take *r* samples from current grammar such that B >> A is always selected
      - Number of times attested candidate for /input/ wins = $\text{Success}_{B >> A}$

    - Compute expected success counts

      $E[\text{Success}_{A >> B}] = \text{Success}_{A >> B} \cdot P(A >> B)$
      $E[\text{Success}_{B >> A}] = \text{Success}_{B >> A} \cdot [1 - P(A >> B)]$

    - Formula for new P(A>>B)

$$P(A \gg B)_{dataset} = \frac{\sum_{/inputs/} E[Success_{A\gg B}]}{\sum_{/inputs/}(E[Success_{A\gg B}] + E[Success_{B\gg A}])}$$

- This computation is done separately for all constraint pairs
  - Creates new probabilities for every constraint pair
  - These new probabilities are then used for sampling and computing expected success counts

4.3 Pairwise probabilities and inconsistency

- Batch EDL algorithm:
  - Expected success counts summed for all inputs

- However, P(A>>B) is also defined for separate inputs:

$$P(A \gg B)_{/input/} = \frac{E[Success_{A\gg B}]}{E[Success_{A\gg B}] + E[Success_{B\gg A}]}$$

- This suggests a definition of "soft inconsistency" between a particular input and the entire dataset

(21) *Soft inconsistency*
If the lexicon yields $P(A >> B)_{dataset} \geq 0.5 + threshold$, then assign inconsistency to all words for which $P(A >> B)_{input} \leq 0.5 - threshold$

- Threshold was set to 0.1

- E.g., if the lexicon yields P(*Clash-Head >> Parse-syll) = 0.6
  - Each word that yields P(*Clash-Head >> Parse-syll) = 0.4 or lower is inconsistent with the lexicon for that constraint pair

- Simplest model:
  - At each iteration, induce or update an indexed constraint for every constraint pair that has inconsistent lexical items
  - Assign all inconsistent lexical items to indexed variant

    e.g., Parse-syll >> *Clash-Head has inconsistent words *advantage*, *confer*
    Parse-syll >> Weight-to-Stress has the inconsistent word *advantage*
    Induce:
    *Clash-Head*i* with *i* = {*advantage, confer*}
    Weight-to-Stress*j* with *j* = {*advantage*}

- Simplest model is problematic:
  - P(A >> B) is computed given imperfect guesses
  - Dependencies between constraints that become apparent at later stages may not be clear when there is uncertainty about lexicon-wide constraint ranking

  - Therefore, especially at the beginning, there will be many "false alarms":
    - Trend-defying lexical items that will later be taken care of by higher-ranked constraints

- Adjusted model – against "false alarms":
  - At each iteration, create or update an indexed version for the constraint whose inconsistent words have the greatest summed difference from the lexicon-wide trend

- Note on inducing indices:
  - To make the model conservative with the number of indices it induces, indices are made recursively
    - All exceptional forms with regard to Ident(cont) go to Ident(cont)*i*
    - If among the exceptional forms in Ident(cont)*i* there are forms that behave yet differently, induce an indexed variant of Ident(cont)*i* → [Ident(cont)*i*] *j*

5. Test on Hebrew spirantization

- Will this learner be able to distinguish within-word from between-word variation in Modern Hebrew?
  - Simplified and reduced version of Temkin-Martínez' (2010) corpus
  - Learner manages to successfully find and rank indexed constraints for postV exceptional forms
    - Underlearning for non-postV exceptional forms
      - Rooted in simplification of simulation

5.1 Data for simulations

- Temkin-Martínez' (2010): corpus of 90 forms (46 roots in total) with probabilistic judgments on spirantization and anti-spirantization (pooled across speakers)

- I picked all 29 forms without multiple stops or anti-spirantization (/f, v, x/ → [p, b, k])

- Transformation of frequency data:
  - Current form of the learner not set up to match relative probabilities of candidates
  - Because learner sees either categorical distribution or 50/50 distribution:
    - For words with > 0.7 observed probability on one candidate, that candidate was marked as the only attested one

      /bnh/ [bana] 0.783 [vana] 0.217
      ⇒ /bana/ [bana]

    - For all other words, candidates with p > 0.1 were given as attested

      /bkh/ [libkot] 0.333 [livkot] 0.614 [libχot] 0.035 [livχot] 0.018
      ⇒ /libkot/ [libkot, livkot]

- With these transformations of the original data, the following frequency pattern with respect to variable and "categorical" application came out:

(22) *Frequencies of variable and "categorical" spirantization in transformed data*

Position of underlying stop /p/, /b/, /k/	Spirantization	Frequency in corpus	Frequency / 4
PostV	yes	12	3
	variable	9	2
	no	4	1
Non-PostV	yes	0	0
	variable	8	2
	no	16	4

- Learning data followed the relative frequencies in the data (divided by four):

(23) *Learning data for Hebrew spirantization (abridged transformed corpus)*

Position of underlying stop /p/, /b/, /k/	Spirantization	Word
PostV	yes	/mebarer/ → [mevarer] /mebatel/ → [mevatel] /gaba/ → [gava]
	variable	/ʃabar/ → [ʃabar, ʃavar] /mekase/ → [mekase, meχase]
	no	/dakar/ → [dakar]
Non-PostV	yes	–
	variable	/lilboʃ/ → [lilboʃ, lilvoʃ] /linpoʃ/ → [linpoʃ, linfoʃ]
	no	/lisbol/ → [lisbol] /liʃkoa/ → [liʃkoa] /liʃpoχ/ → [liʃpoχ] /lizkot/ → [lizkot]

5.2 OT setup for simulations

- Constraints used in analysis (after Temkin-Martínez 2010):
    - *Stop
    - *Fric
    - *PostV.Stop
    - Ident(cont)
    - Max

- Exceptional spirantization (differs from Temkin-Martínez' account)
  *Stop*i* >> *Fric >> *Stop
- Exceptional non-spirantization
  *Fric*j* >> *Stop >> *Fric
  OR Ident-cont*j* >> *Stop >> *Fric, Ident-cont

- Candidates for each word:
    - Apply spirantization to stop and/or delete segment preceding the stop

(24) *Sample tableau: postV spirantization*

/mekase/	Max	*PostV.Stop	*Fric	*Stop	Ident(cont)
mekase		*!		*	
meχase			*		*
mkase	*!			*	
mχase	*!				*

5.3 Results

- Simulations run as for English secondary stress:
  - 20 runs, 80 iterations, inconsistency threshold = 0.1, sample size (*r*) = 50

(25) *Results for Hebrew spirantization*

Correct convergence (≥95% data likelihood)	20/20 runs
Average time till convergence	13.95 iterations (range: 11–20)

- Mostly uniform behavior between runs:
  - Words with exceptional postV patterns always indexed
    - Exceptional spirantization: exceptional *Stop
    - Exceptional non-spirantization: exceptional *Fric
  - Words with exceptional non-postV pattern generally not indexed
    - Exceptional non-spirantization: no indication except 1/4 words in 1/20 runs

(26) *Patterns of indexed constraints for Hebrew spirantization*

	Word	Winner(s)	Indexed to
postV	mebarer	v, *b	*Stop$i$ (20 runs)
	mebatel	v, *b	*Stop$i$ (20 runs)
	gaba	v, *b	*Stop$i$ (20 runs)
	ʃabar	b ~ v	
	mekase	k ~ χ	
	dakar	k, *χ	*Fric$j$ (19 runs) Ident-cont$k$ (1 run)
non-postV	lilboʃ	b ~ v	
	linpoʃ	p ~ f	
	lisbol	b, *v	
	liʃkoa	k, *x	
	liʃpoχ	p, *f	*Fric$j$ (1 run)
	lizkot	k, *χ	

- Why did the learner not note exceptional lack of spirantization in non-postV position?
  - Learner was not set up to match relative probability of output candidates
  - At sampling stage: whenever <u>one of the attested candidates</u> is matched, success is recorded

    /lilboʃ/ → [lilboʃ, lilvoʃ] will yield an equal number of matches for
    *Stop >> Ident (matches [lilvoʃ]) and for
    Ident >> *Stop (matches [lilboʃ])

    /lilboʃ/ → [lilboʃ, lilvoʃ] will have 100% likelihood when grammar always produces /lilboʃ/ → [lilboʃ] (since one of the attested candidates is matched)

    - The majority of non-postV words provide evidence for Ident >> *Stop only
        - No words that provide evidence for *Stop >> Ident only
        - Thus, no inconsistency, and no motivation to infer exceptionality

- This should be remedied when the learner is set up to match probabilities of outputs
    - Work in progress

5. Concluding remarks

- Existing models of lexically indexed constraint induction:
    - Not built to handle within- and between-word variation
- Modern Hebrew spirantization:
    - Orthogonal within- and between-word variation
        - Items that have within-word variation are non-exceptional
        - Items that have no within-word variation are exceptional
- Current proposal:
    - "Soft inconsistency" when certain words prefer opposite pairwise ranking compared to lexicon-wise trend
    - Induce exceptional (lexically indexed) constraint for constraint pair with strongest "soft inconsistency"
- Tested on Hebrew:
    - Exceptionality in postV position learned excellently
    - Exceptionality in non-postV position underlearned because learner does not match output probabilities
- Future work:
    - Set up learner to match probabilities
    - Look at fuller dataset
    - Look at other cases of variation
    - Look at interaction between underlying form learning (Moore-Cantwell 2017) and exceptionality

References

Allen, Blake, and Michael Becker. under revision. "Learning Alternations from Surface Forms with Sublexical Phonology"

Anttila, Arto. 2002. "Morphologically Conditioned Phonological Alternations." *Natural Language & Linguistic Theory* 20 (1): 1–42.

Becker, Michael. 2009. "Phonological Trends in the Lexicon: The Role of Constraints." PhD dissertation, University of Massachusetts, Amherst.

Becker, Michael, and Maria Gouskova. 2016. "Source-Oriented Generalizations as Grammar Inference in Russian Vowel Deletion." Linguistic Inquiry 47 (3).

Boersma, Paul. 1998. "Functional Phonology: Formalizing the Interactions between Articulatory and Perceptual Drives." PhD dissertation, University of Amsterdam.

Chomsky, Noam. 1964. Current Issues in Linguistic Theory. The Hague: Mouton.

Chomsky, Noam, and Morris Halle. 1968. The Sound Pattern of English. New York, Evanston, and London: Harper and Row.

Coetzee, Andries W. 2009. "Learning Lexical Indexation." Phonology 26 (1): 109–45.

Coetzee, Andries W, and Joe Pater. 2011. “The Place of Variation in Phonological Theory.” In Handbook of Phonological Theory, edited by John A. Goldsmith, Jason Riggle, and Alan C. Yu, 2nd ed., 401–34. Wiley-Blackwell.

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. “Maximum Likelihood from Incomplete Data via the EM Algorithm.” Journal of the Royal Statistical Society. Series B (Methodological 39 (1): 1–38.

Goldwater, Sharon, and Mark Johnson. 2003. “Learning OT Constraint Rankings Using a Maximum Entropy Model.” In Proceedings of the Stockholm Workshop on Variation within Optimality Theory, edited by Jennifer Spenader, Anders Eriksson, and Osten Dahl, 111–20.

Gouskova, Maria, Luiza Newlin-Łukowicz, and Sofya Kasyanenko. 2015. “Selectional Restrictions as Phonotactics over Sublexicons.” Lingua 167: 41–81.

Inkelas, Sharon. 1998. “The Theoretical Status of Morphologically Conditioned Phonology: A Case Study from Dominance.” Yearbook of Morphology 1997: 121–55.

Inkelas, Sharon, and C. Orhan Orgun. 1995. “Level Ordering and Economy in the Lexical Phonology of Turkish.” Language 71: 763–93.

Inkelas, Sharon, and C. Orhan Orgun. 1998. “Level (Non)ordering in Recursive Morphology: Evidence from Trukish.” In Morphology and Its Relation to Phonology and Syntax, edited by Steven Lapointe, Diane Brentari, and Patrick Farrell, 360–92. Stanford, CA: CSLI Publications.

Inkelas, Sharon, and Cheryl Zoll. 2007. “Is Grammar Dependence Real? A Comparison Between Cophonological and Indexed Constraint Approaches to Morphologically Conditioned Phonology.” Linguistics 45 (1): 133–71.

Itô, Junko, and Armin Mester. 1995. “The Core-Periphery Structure of the Lexicon and Constraints on Reranking.” In University of Massachusetts Occasional Papers in Linguistics, edited by Jill Beckman, Suzanne Urbanczyk, and Laura Walsh, Vol. 18: Papers in Optimality Theory:181–209. Amherst, MA: GLSA.

Itô, Junko, and Armin Mester. 1999. “The Structure of the Phonological Lexicon.” In The Handbook of Japanese Linguistics, edited by Natsuko Tsujimura, 62–100. Oxford: Blackwell.

Jarosz, Gaja. 2006. “Rich Lexicons and Restrictive Grammars: Maximum Likelihood Learning in Optimality Theory.” PhD dissertation, Johns Hopkins University.

Jarosz, Gaja. 2015. “Expectation Driven Learning of Phonology” Ms., University of Massachusetts Amherst.

Joos, Martin. 1942. “A Phonological Dilemma in Canadian English.” Language, 141–44.

Kim, Heejin, and Jennifer Cole. 2005. “The Stress Foot as a Unit of Planned Timing: Evidence from Shortening in the Prosodic Phrase.” In INTERSPEECH-2005, 2365–68.

Kraska-Szlenk, Iwona. 1995. “The Phonology of Stress in Polish.” PhD dissertation, University of Illinois, Urbana-Champaign.

Melvold, Janis. 1989. “Structure and Stress in the Phonology of Russian.” PhD dissertation, Massachusetts Institute of Technology.

Moore-Cantwell, Claire. 2017. “Concurrent learning of the lexicon and phonology.” Talk given at the LSA 2017 Annual Meeting.

Moore-Cantwell, Claire, and Joe Pater. 2016. “Gradient exceptionality in Maximum Entropy grammar with lexically specific constraints.” *Catalan Journal of Linguistics* 15: 53-66.

Mullin, Kevin. 2011. “The Necessity of Diacritics for Descriptive Adequacy.” Talk given at

RUMMIT 2011.
Nazarov, Aleksei. 2016. “Extending Hidden Structure Learning: Features, Opacity, and Exceptions.” PhD dissertation, UMass Amherst.
Osadcha, Iryna. 2014. On Nominal Stress in Ukrainian. Ms. (Generals Paper), University of Toronto.
Pater, Joe. 2000. “Non-Uniformity in English Secondary Stress: The Role of Ranked and Lexically Specific Constraints.” Phonology 17 (2): 237–74.
Pater, Joe. 2010. “Morpheme-Specific Phonology: Constraint Indexation and Inconsistency Resolution.” In Phonological Argumentation: Essays on Evidence and Motivation, edited by Steve Parker, 123–54. London: Equinox Press.
Potts, Christopher, Joe Pater, Karen Jesney, Rajesh Bhatt, and Michael Becker. 2010. “Harmonic Grammar with Linear Programming: From Linear Systems to Linguistic Typology.” Phonology 27 (1): 1–41.
Prince, Alan S. 1997. “Stringency and Anti-Paninian Hierarchies.” presented at the LSA Institute. http://ruccs.rutgers.edu/images/personal-alanprince/gamma/talks/insthdt2.pdf.
Temkin-Martínez, Michal. 2010. Sources of Non-Conformity in Phonology: Variation and exceptionality in Modern Hebrew Spirantization. PhD dissertation, USC.
Tesar, Bruce B. 1995. “Computational Optimality Theory.” University of Colorado.
Wu, C.F. Jeff. 1983. On the Convergence Properties of the EM algorithm. *Annals of Statistics* 11(1): 95-103.