Class 14 (Week 7, T)
Learning Models I: non-numerical learning algorithms for constraint grammars

To do
- ☐ Read **Jarosz (submitted)** for Thursday (Nov. 12)
  - ▪ presenters, if you e-mail me your handout as a PDF by noon Thurs., I can print
- ☐ Prepare at least one **question or point for discussion** on the reading
- ☐ Computing **homework** using MaxEnt Grammar Tool is due Thursday (Nov. 2). Turn in write-up on paper; I enabled file upload on CCLE for your 4 text files.

Overview: We know how a grammar works once we have a ranking or set of weights, but how does the learner arrive at that ranking/set of weights? Today we focus on ranking (no numbers) and hidden structure, if we get that far, and Thursday we look at learning numerical weights. (Or we might push the numerical stuff till next Tuesday—we'll see how it goes.)

1. Seeing what algorithm(s) you've already internalized

- o In the following tableaux, you have the winners indicated, and the constraint violations given, but the constraints are not in the right ranking. Figure out the constraint ranking and draw a Hasse diagram of it. Then consider/discuss: how did you figure it out? Are there some simple moves that a computer could have done? (Samoan, from Zuraw, Yu & Orfitelli 2014)

/maile/ 'dog'	*AÍ	DON'T LENGTHEN	DON'T SHORTEN	EDGEMOST-R	FOOT BINARITY	PARSE-σ
a ma(íle)	*					*
☞ *b* (mái)le				*		*
c (mà:)(íle)		*				
d (máile)					*	
/maela/ 'holoow'						
☞ *e* ma(éla)						*
f (máe)la				*		*
g (mà:)(éla)		*				
h (máela)					*	
/tu:si/ 'point'						
☞ *i* (túsi)			*			
j (tú:)si				*		*
k (tù:)(sí:)		*				
l (tú:si)					*	

2. Error-Driven Constraint Demotion (Tesar & Smolensky 2000)

- o Apply the following algorithm (p. 52) to the Samoan data above

- Start with some or no ranking of the constraints—you can use alphabetical order, as above, or you could start with all the constraints in one "stratum" {A, B, C, D, ...}
- For each tableau
    - see what candidate the current ranking says should win
        - if you have any strata with multiple constraints in them, it may be unclear what the winner is supposed to be
        - in that case, T&S propose that you just merge all the constraints in the stratum into a single constraint, adding together the violations for each candidate
        - note that they are **not** proposing that this happens in an actual grammar, and their learning algorithm doesn't work if real grammars are allowed to do this (I think?)
            - instead, this is a special move for learning
    - if that's the correct output, move on to the next tableau
    - if that is not the correct output, compare the spurious winner to the correct output as follows
        - cancel out any violation marks that both candidates have ("mark cancellation")
        - find the **highest-ranked** constraint that still assigns a mark to the spurious winner; call it $C_{spuriousHater}$
        - for each constraint $C_{outputHater}$ that still assigns a mark to the correct output
            - if it's not already the case that $C_{spuriousHater}$ >> $C_{outputHater}$, then demote $C_{outputHater}$ to the stratum right below $C_{spuriousHater}$ (pp. 36-37)
            - For example, if you had A >> B >> C, and you want to demote A to the stratum right below B, now you have B >> {A, C}
- At the end, if you still have multiple constraints in the same stratum, it should be the case that the ranking of those constraints doesn't matter—no matter how you rank them, you get the same winner in every tableau.
    - Maybe the human learner chooses some linear ranking of them, but we have no way of knowing.

(A bit of a problem, unless I've always misunderstood this: I think you often get into situations where you have to loop back through all the tableaux again, because one of the tableaux caused a demotion that destroyed information gained from a previous tableau. So you might need to **loop back through all the tableaux** repeatedly, until they all produce the correct winner.)

- o How does the resulting ranking compare to our Hasse diagram?

3. Recursive constraint demotion (Tesar & Smolensky 2000)

o Apply the following algorithm (pp. 106-108) to the Samoan data above

- There is no starting ranking, just a list of constraints needing to be ranked.
- For each winner (correct output)-loser pair in each tableau
    - list all the winner's uncancelled "marks", and all of the loser's uncancelled "marks"
    - if there are any constraints that assign no marks to any winners, put ("install") those constraints in the top stratum, and remove them from the need-to-be-ranked list
    - if there are any losers that get marks from that top stratum, cross them off the list of candidates (we don't have to worry about ruling them out any more)
- Repeat with the remaining list of constraints and the remaining candidates
    - except now, instead of installing in the top stratum, install constraints in a stratum right below the previous stratum

(There's also a "biased" version of this, Hayes 2004; Prince & Tesar 2004, which seeks to install markedness constraints as high as possible, as a way to address the subset problem: how does a child learning, say, Hawaiian, know that NoCoda is ranked high? Recall Tessier & Jesney's 2014 use of this.)

- OTSoft implements something like this (but also like constraint demotion), including biased versions.

4. Robust Interpretive parsing (Tesar 1999)

- Let's make things a lot harder for the learner now,
  - Above, we assumed that the learner knows what the winning candidate is: [(mái)le, ma(éla), (máe)a, (àː)(í.ŋa)]
  - But what the learner actually hears is more like [máile, maéla, máea, àːíŋa]
  - This makes it really hard to rank certain constraints (which ones?)

- Simple stress in Samoan—the grammar we want to end up with (for some of the rankings, we still haven't seen the data that would justify):

/ iŋoa / 'name'	FOOT BINARITY	RHYTHMTYPE= TROCHEE	EDGEMOST-R	PARSE-σ	ALIGN (PrWd,L;Ft,L)	RHYTHMTYPE= IAMB
☞ *a* i(ŋóa)				*	*	*
i (ì)(ŋóa)	*!					*
b i(ŋoá)		*!		*	*	
c iŋo(á)	*!			**	*	
d (íŋo)a			*!	*		*
/ sikalamu / 'scrum'						
☞ *e* (sìka)(lámu)						**
/ temokalasi / 'democracy'						
☞ *g* (tèmo)ka(lási)				*		**
h te(mòka)(lási)				*	*!	**

- Apply Tesar's proposal, below. We should probably start together and then you can break into groups.

- Start with some ranking (just not the one above, or it won't be much fun)
- For an observed form, like [iŋóa]
  - apply the current grammar to the input, and see what output you get ("production-directed parsing")
    - Be careful: there may be important candidates not shown above. Think through what each constraint wants, rather than manipulating the candidates shown.
  - compare all the parses of the observed form, [i(ŋó)a, (iŋó)a, i(ŋóa)], and determine which one the current grammar likes best ("robust interpretive parsing")
  - if those two forms are different, do constraint demotion based on that pair, as above, with the robust interpretive parsing form as the correct output and the production-directed parsing form as the spurious winner
  - repeat the three steps above until the two forms chosen are identical (then go on to the next tableau)

(This procedure isn't guaranteed to work! We'll see how it does for Samoan. By the way, Praat includes Robust Interpretive Parsing functionality.)

5. Coming up

- The most hidden structure of all: the input!
- The Gradual Learning Algorithm for Noisy Harmonic Grammar
- The Gradual Learning Algorithm for Stochastic OT (including, what is Stochastic OT?)
- How MaxEnt weights are learned (including, what is gradient descent?)

References

Hayes, Bruce. 2004. Phonological acquisition in Optimality Theory: The early stages. In René Kager, Joe Pater & Wim Zonneveld (eds.), *Constraints in Phonological Acquisition*. Cambridge: Cambridge University Press.

Jarosz, Gaja. submitted. Expectation driven learning of phonology. UMass, ms.

Prince, Alan & Bruce Tesar. 2004. Learning phonotactic distributions. In René Kager, Joe Pater & Wim Zonneveld (eds.), *Constraints in Phonological Acquisition*, 245–291. Cambridge: Cambridge University Press.

Tesar, Bruce. 1999. Robust interpretive parsing in metrical stress theory. In Kimary N Shahin, Susan J Blake & Eun-Sook Kim (eds.), *The Proceedings of the West Coast Conference on Formal Linguistics 17*. Stanford, CA: CSLI Publications.

Tesar, Bruce & Paul Smolensky. 2000. *Learnability in Optimality Theory*. Cambridge, Mass.: MIT Press.

Tessier, Anne-Michelle & Karen Jesney. 2014. Learning in Harmonic Serialism and the necessity of a richer base. *Phonology* 31(01). 155–178.

Zuraw, Kie, Kristine Mak Yu & Robyn Orfitelli. 2014. Word-level prosody in Samoan. *Phonology* 31(2). 271–327.