Class 18 (Week 9, T)
Induction I: constraints

To do
☐ Work on your project! (There's no more homework and no more reading.)
☐ There is no class on Thursday (Thanksgiving)
☐ Next Tuesday we meet **4-6 PM**, in **Campbell 2122** (conference room)

Overview: What if we aren't born with a constraint inventory? We'll take a tour of some proposals.

1. Discuss Hayes & Wilson (2006)

2. Naradowsky, Pater & Smith (2011): Error-driven constraint induction

- Do we really need to consider such a broad range of constraints?
- Main idea:
  - When you hear a form (say, [#n…]), compare it to a randomly sampled, hypothetical similar form (e.g., [#ŋ...])
  - If your current grammar assigns better harmony to the sampled neighbor than to the observed form, the grammar needs updating
    - Can you see a problem with this?

    - Refinement: remember the last *n* forms you heard (here, *n*=3000), and don't worry about a neigbor that's in that set

- How to construct a constraint
  - Make a lattice of all the constraints that could prefer the observed form
    - In the [#n…]/[#ŋ...] example, consider all constraints that penalize [+dorsal]
      - In the feature system used in this paper, [CORONAL] is privative
      - Which means what?

      - Let's draw some of the lattice.

  - Sample a few of these constraints (here, 3) and add them to the grammar
  - Update weights of this new constraint set
    - Use Perceptron rule to update: similar to Gradual Learning Algorithm, except it cares about the number of violations
- More precisely: see over

Algorithm 1 Learning Procedure

```
function LEARN
    model ← []
    while learning do
        o ← draw(D)
        u ← draw(neighborhood(o)
        if model.score(u) > model.score(o) then
            diff ← u.features \o.features            ▷ Constraint Induction
            cspace ← []
            for c ← allconstraints do                 ▷ Filter Space
                if c.contains(diff) then
                    cspace+ = c
                end if
                model.features+ = draw(space)         ▷ Add constraint
            end for
            for j ← 0 until model.size do             ▷ Perceptron Update
                model[j] = f.violations(u) − f.violations(o) ∗ rate
            end for
        end if
    end while
end function

function NEIGHBORHOOD(o)
    plattice ← [o.size]
    for i ← o.size do
        plattice[i] ← o[i].phones                     ▷ (Phone, Distance) pairs
    end for
    return draw(plattice.allPaths)                    ▷ Sum weight in each path
end function
```

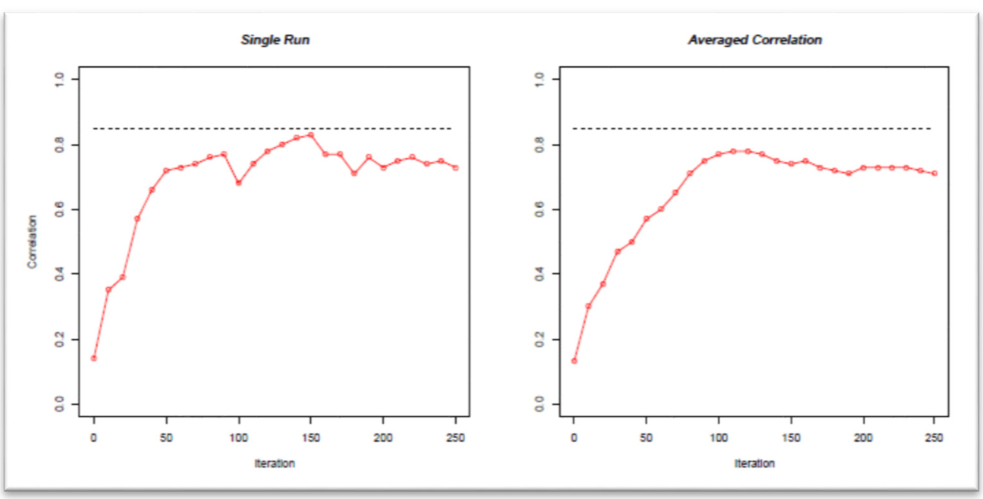
(p. 13)

- Tested on English onsets, as in Hayes & Wilson (2006)
- Resulting constraints (over):
    - [__] means “any segment”, i.e. [+segmental]

Weight	Feature	Description
1.34	[_] [^-strid,+cor]	S T vs. TH S
1.28	[^+cons,+ant]	S vs ZH
1.26	[^-approx,+lab]	M vs. NG
1.26	[_] [^-cons,+cor]	G R vs. P W
1.26	[-approx,-voice] [^-ant,+cor]	G R vs. CH L
1.22	[^-son,-voice] [_]	S K vs. Z P
1.22	[^+strid][-strid,+son]	S L vs TH L
1.20	[^-son,+voice][+lab]	G R vs. P W
1.20	[^+dorsal,+cons][^+approx,-ant]	G R vs. P W
1.20	[^+dorsal][^-strid,-ant]	G R vs. CH L
1.18	[^+cont,-voice]	S vs. ZH
1.18	[+strid,+cor][^-strid]	S T vs. S S
1.18	[-son,-voice][_]	G R vs. P W
1.18	[_][^+approx,+cor]	G R vs. P Y
1.16	[-son,+ant][-ant]	S W vs. S R
1.16	[_][^+high,+approx]	S W vs. S R
1.16	[_][^+high,+son]	S W vs. S R
1.16	[^-voice][_]	T R vs. V R
1.14	[^+strid,-voice][_]	S L vs. TH L
1.14	[-son][^+ant,+lat]	S L vs. S R

(p. 16)

- Correlation to human judgments of English onsets: -0.829 (vs. Hayes & Wilson's -0.859)
- Interesting problem: tendency towards over-fitting as learning continues
    - As assessed by worsening correlation to human judgments



(p. 18)

  - See the paper for some ways to improve the model

- Discuss: Both this model and the Hayes/Wilson model are for phonotactic learning. How could we extend either of these to learning alternations?

---

3. Flack (2007): inducing a constraint from perceptual experience

We got this far.

- There are languages that prohibit [p] specifically in word-initial position: *#P
  - Initial [p] has particularly short VOT, and it's more variable than initial [b]'s
  - Difference in maximum burst intensity for initial [p] and [b] is smaller than for other voiceless-voiced pairs (p. 122)

- To produce an instance of a category ([p], [b], [t], etc.) in a context, speaker samples values for various phonetic dimensions from stored distributions centered on prototype
- In perception, listener must guess the category
  - Some noise is added: perception is imperfect
  - Rather than Bayes' rule as in Kirby (2013), finds the closest prototype
- Listener gets feedback on accuracy
  - Allows listener to update prototypes
  - Listener also **stores accuracy rate** for each category, perhaps over a moving window of the past *n* tokens (here, *n*=400)
    - Specifically, **hit rate** and **false alarm rate**
    - Does anyone know these terms?

- Hit rates for each consonant as learning proceeds over time:

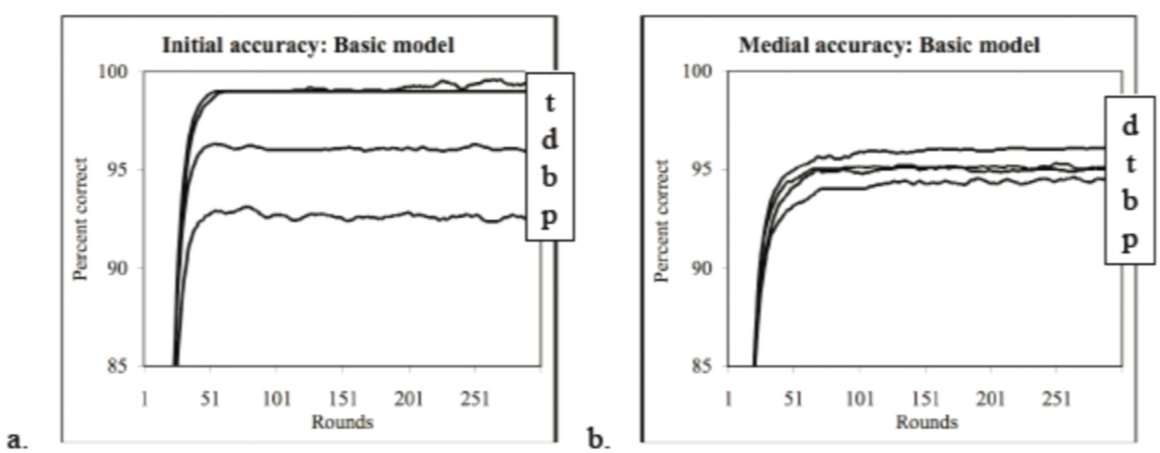


Figure 7. Model accuracy for each initial (a) and medial (b) consonant, averaged across 20,000 simulations of 300 rounds each. The lines represent moving averages across 15-round windows.

(p. 141)

- Important point about phonologization
  - Once *#P has been promoted high enough, the learner gets no experience of initial [p]!
  - But they do still have one important piece of information listed above—can you guess what?

- The learner's rule for inducing a constraint:

(117) Some segment *x* is perceptually difficult in some context $Context_Z$ if either:

a. Accuracy($x/Context_Z$) < *threshold*

and

Accuracy($x/Context_Z$) < Accuracy($y/Context_Z$) ➔ Constraint *$x/Context_Z$

This difference must be significant ($\alpha = 0.01$).

b. Accuracy($x/Context_Z$) < FalseAlarm($x/Context_Z$) ➔ Constraint *$x/Context_Z$

(p. 160)

  - ▪ Where "accuracy" means hit rate
  - ▪ If there is no hit rate, because the sound never actually occurs in that context, treat it as 0.
  - ○ So how would this work for a language with no initial [p]? Let's draw a possible confusion matrix.

- Results: both a learner of simplified French (has initial [p], but it is perceptually difficult) and a learner of simplified Cajonos Zapotec (no initial [p]) learn *#P in nearly all runs

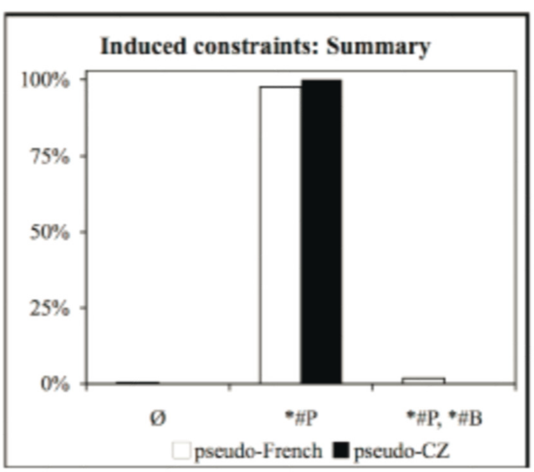


Figure 23. Constraints induced in each of 250 simulations of 40,000 rounds each.

(p. 173)

4. A selection of other approaches that we won't have time to cover in depth

- Hayes (1999): as we saw last time, generate lots of features according to a set of templates, and then select the ones that match the phonetic map well, with a bias favoring simpler constraints

- Boersma & Pater (2007): in Harmonic Grammar, construct *positive* constraints for every property that the observed form has (as well as some other constraints, including negative ones)
  - e.g., on observing Canadian English [ʔʌɪs] 'ice', construct these, among many others:

(17) *Observed structure* *Constructed constraint*

a. D
[-low] RAISED DIPHTHONG: A diphthong must be [-low]
(Assign a reward of 1 to each diphthong that is [-low])

b. D C
[-low][-vce] RAISED/VOICELESS: A diphthong preceding a voiceless consonant must be raised
(Assign a reward of 1 to each diphthong that is [-low] that precedes a voiceless consonant)

(p. 4)

  - Pater (2014) proposes something similar for the same case, but now without positive constraints

- Moreton (2010): explore infinite space of possible constraints with evolutionary algorithm
  - Every subpart of every possible representation is a constraint
  - Start with a random set of constraints
  - Error-driven: if current grammar selects a candidate that doesn't match the observed true winner…
    - constraints that favor observed forms (correct winners) are allowed to breed.
    - breeding = combine two constraints to produce a new, offspring constraint with aspects of each parent. Offspring can also mutate.

- Pizzo (2013): Inducing constraints to handle alternations (Turkish vowel harmony and devoicing)
  - On making an error, create a constraint at random
  - According to certain templates
  - Can be faithfulness or markedness
  - Must penalize some way in which the spurious winner differs from the observed winner
  - The researcher can set parameters for how much stem-faithfulness and tier-markedness constraints show be allowed/favored

- Alderete, Tupper & Frisch (2013): Connectionst model of OCP-Place in Arabic roots

5. Coming up next Tuesday (there is no class this Thursday)

- Inducing feature and natural classes

References

Alderete, John, Paul Tupper & Stefan A. Frisch. 2013. Phonological constraint induction in a connectionist network: learning OCP-Place constraints from data. *Language Sciences* 37. 52–69.

Boersma, Paul & Joe Pater. 2007. Constructing constraints from language data: the case of Canadian English diphthongs (handout). Paper presented at the 38th meeting of the North East Linguistic Society.

Flack, Kathryn Gilbert. 2007. The Sources of Phonological Markedness. University of Massachusetts, Amherst PhD dissertation.

Hayes, Bruce. 1999. Phonetically driven phonology: the role of Optimality Theory and inductive grounding. In Michael Darnell, Frederick J Newmeyer, Michael Noonan, Edith Moravcsik & Kathleen Wheatley (eds.), *Functionalism and Formalism in Linguistics, Volume I: General Papers*, 243–285. Amsterdam: John Benjamins.

Hayes, Bruce & Colin Wilson. 2006. A Maximum Entropy Model of Phonotactics and Phonotactic Learning.

Kirby, James. 2013. The role of probabilistic enhancement in phonologization. In Alan C. L. Yu (ed.), *Origins of sound change*, 228–246. Oxford: Oxford University Press.

Moreton, Elliott. 2010. Constraint induction and simplicity bias in phonological learning (handout). Workshop on Grammar Induction. Cornell University, ms.

Naradowsky, Jason, Joe Pater & David Smith. 2011. Feature induction for online constraint-based phonology acquisition. *The Northeast Computational Phonology Workshop (NECPhon)*.

Pater, Joe. 2014. Canadian raising with language-specific weighted constraints. *Language* 90(1). 230–240..

Pizzo, Presley. 2013. An online model of constraint induction for learning phonological alternations. Instanbul.