Linguistics 201A   Winter 2019
Phonological Theory I   B. Hayes

Class 7, 1/31/2023:
Token Variation in Phonology; MaxEnt Analysis

1. Current assignments

- Hebrew homeworks due Thursday
- Hand in half-page summaries of Labov chapter.
- For Thurs. 2/2: read:
    - ➢ Bruce Hayes (2022) Deriving the Wug-shaped curve: A criterion for assessing formal theories of linguistic variation. *Annual Review of Linguistics* 8:474-494.
    - ➢ Download from course website.
    - ➢ No summary required

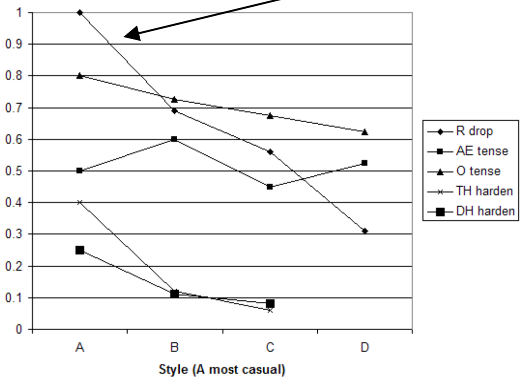
REVIEW OF TOKEN VARIATION

2. Labovian method and "lockstep" phenomena

- Elicit a range of styles
- Look at a controlled set of particular phenomena
- Plot them together, revealing systematicity

3. Lockstep is probably not perfect: variation in the speech of Doris

- Doris is 39, homemaker, African-American.
- She doesn't have perfect lockstep
- Labov thinks that for Doris, and others, r-dropping is more sensitive to style than other processes.



4. What does the language learner learn?

- Is there a magic formula (P-map based?) that tells them the slope of each line?

- Or are the slopes all learned separately?

5. A paper on (frequency) knobs

- Coetzee, Andries W., and Shigeto Kawahara (2013) Frequency biases in phonological variation. *Natural Language & Linguistic Theory* 31:47-89.

SKEPTICISM ABOUT VARIATION

6. Some currently active variation-skeptics

- Karthik Durvasula, Bruce Tesar, Huteng Dai, Paul Smolensky, Kristin Hanson, others
- None of the views below should necessarily be attributed to any particular person on this list.

7. “Different speakers speak different dialects”

- That’s why it’s good to spend a lot of time with one individual, as Labov did.

8. Trying to detect free variation in individual people in an experimental context

- from Hayes and Londe’s wug test; *Phonology* (2009)
- Hungarian stems whose last vowels are [+back], then [eː] go both ways with harmony.
- We wug-tested [ha̲ː̲de̲ː̲l] and [ko̲le̲ː̲n]
  - Options for dative: [haːdeːl-nɔk, haːdeːl-nɛk], [koleːn-nɔk, koleːn-nɛk]
- “In a series of chi-square tests, we found that consultants who gave [haːdeːl-nɔk] were no more likely to give [koleːnnɔk] than consultants who gave [haːdeːl-nɛk]. We obtained similar results for all other pairs where enough data were available for testing.”

9. “Variation is actually rapid switching between multiple (internally invariant) grammars”

- Actually, I think this really does happen:
  - polydialectalism
  - code-switching
- But to do real-life Labovian variation, it would take too many grammars; and miss the orderly relation between the patterns.

10. Grammars are invariant, variation is due to performance

- Is this claim helpful in the absence of a substantive account of performance?
- Many scholars have emphasized: the constraints that modulate frequency in Language A can be inviolable in Language B — variation is dumped into a separate component at a cost in generality.

WEIGHTED-CONSTRAINT-BASED FRAMEWORKS
AND HOW THEY DESCRIBE VARIATION

11. A useful starting point: harmonic grammar

- It's older than OT.
- Similar to an OT grammar, except that the constraints have numerical **weights**, not rankings.
- Basic operation of harmonic grammar:
  - For each candidate, **multiply** each constraint weight by the violation count for that constraints = **harmony**. This is like a "weighted sum".
  - Winner is the candidate with the lowest harmony. (Think of it as a penalty.)

12. References for harmonic grammar

- Legendre, Géraldine; Miyata, Yoshiro; & Smolensky, Paul. (1990). Harmonic Grammar: A formal multi-level connectionist theory of linguistic well-formedness: Theoretical foundations. Report CU-CS-465-90. Computer Science Department, University of Colorado at Boulder.
- Smolensky, Paul, and Geraldine Legendre. 2006. *The Harmonic Mind*. Cambridge: MIT Press. (summarizing work of two decades, including early Harmonic Grammar)
- Pater, Joe (2009) Weighted constraints in generative linguistics. *Cognitive Science* 33: 999-1035.
- Potts, C., J. Pater, K. Jesney, R. Bhatt, and M. Becker (2009) Harmonic Grammar with Linear Programming: From linear systems to linguistic typology. *Phonology* 27:77-118.
  - The math needed is from the 1940s — also very useful in steel production!

13. Harmonic grammar's most salient prediction

- **Ganging**: two constraints together, but separately, are stronger than a third constraint
  - (Or: two violations of one constraint, but not just one, are stronger than a second constraint).
  - Key reference is Jäger and Rosenbach (2008).[1]
- In Harmonic Grammar, constraint ganging is *always on*, not just when we set up a specific conjoined constraint.

14. A real-life example of ganging: Japanese consonant voicing

- Take a look at these forms and discuss when Japanese devoices voiced obstruents in foreign loans.
- Hint: the famous "Lyman's Law" of Japanese states that you cannot have two voiced obstruents in the same stem.
- Source: Shigeto Kawahara (2006) A faithfulness ranking projected from a perceptibility scale: the case of [+voice] in Japanese. *Language* 82:536-574.

[1] Jäger, Gerhard & Anette Rosenbach. 2006. The winner takes it all – almost: cumulativity in grammatical variation. *Linguistics* 44.937–971.

No devoicing

webbu	‘web’
sunobbu	‘snob’
habburu	‘Hubble’
kiddo	‘kid’
reddo	‘red’
heddo	‘head’
suraggaa	‘slugger’
eggu	‘egg’
furaggu	‘flag’

bagii	‘buggy’	bogii	‘bogey’
bobu	‘Bob’	bagu	‘bug’
dagu	‘Doug’	daibu	‘dive’
daijamondo	‘diamond’	doguma	‘dogma’
giga	‘giga- (prefix)’	gaburieru	‘Gabriel’
gibu	‘give’	gaidansu	‘guidance’

Devoicing

gepperusu	‘Göbbels’
gutto	‘good’
betto	‘bed’
doretto	‘dreadlocks’
dettobooru	‘dead ball (baseball term)’
batto	‘bad’
deibitto	‘David’
dokku	‘dog’
bakku	‘bag’
dorakku	‘drug’
bikku	‘big’

15. Harmonic Grammar analysis

- Socrates: let us haul out Excel, and try finding weights for
  - IDENT(voice)
  - LYMAN'S LAW[2] = *[−sonorant,+voice] … [−sonorant,+voice]
  - *VOICED GEMINATE

  that will derive Kawahara's pattern.
- Inputs and candidates:
  - /bobu/ → ✓bobu, *bopu, *pobu, *popu
  - /webbu/ → ✓webbu, *weppu
  - /doggu/ → ✓dokku, *doggu, *dokku, *tokku

- For our spreadsheet, we will want to amplify an OTSoft file with:
  - a row of weights
  - a column to compute Harmony — the SUMPRODUCT() function works great.

16. How common is ganging?

- Potts et al. (*Phonology* 2010) give an elaborate case from Lango, which works very cleanly for them with (nonstochastic) Harmonic Grammar
- Hayes and Wilson (*LI* 2008) point out possible ganging effects in English phonotactics—sounds (e.g. rare [ð] is both a voiced fricative and a dental fricative)
- The OT literature is replete with conjoined constraints.
- *Stochastic* ganging is ubiquitous (readings)

[2] It's really Lyman's Constraint, but nobody calls it that…

17. Returning to variation: varieties of harmonic grammar

- **Classical harmonic grammar**
  - ties only possible when the harmony of two candidates is identical.
- **Noisy harmonic grammar**
  - Let the weights be "perturbed" at each "evaluation time", as in Boersma's (1998) Stochastic OT.
  - Reference: Boersma, Paul, and Joe Pater. 2016. Convergence properties of a gradual learning algorithm for Harmonic Grammar. In John McCarthy and Joe Pater, eds. *Harmonic Grammar and Harmonic Serialism*
  - Noisy Harmonic Grammar is implemented in OTSoft 2.6 (current version).
  - There are many different ways to set up the noisy system — see Hayes (2017), Hayes and Kaplan (in press) for attempts to explore them.[3]
- **Maximum Entropy (maxent) grammar**
  - next

MAXENT GRAMMARS

18. The basic strategy

- Use a classical formula (< 19th century physics) to convert harmony to probability
- As it works out, each candidate gets a positive probability, but this can be so close to zero that most people are willing to treat it as implying impossibility.
- Refs:
  - Goldwater, Sharon, and Mark Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. Proceedings of the Stockholm Workshop on Variation within Optimality Theory, ed. by Jennifer Spenader; Anders Eriksson, and Osten Dahl, 111–120. Stockholm: Stockholm University Department of Linguistics.
  - Wilson, Colin (2006) Learning phonology with substantive bias: an experimental and computational study of velar palatalization. *Cognitive Science* 30:945-82.
  - Hayes, Bruce and Colin Wilson, (2008) A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry* 39: 379-440.

19. The maxent formula

- Found everywhere, but I usually cite the rendering in Goldwater and Johnson (2003):

$$\mathrm{Pr}\,(x) = \frac{\exp(-\Sigma_i\, w_i \mathrm{f}_i\,(x))}{Z}, \text{ where } Z = \Sigma_j\ \exp(-\Sigma_i\, w_i \mathrm{f}_i\,(x_j))$$

[3] (2017) Bruce Hayes, Varieties of Noisy Harmonic Grammar. Proceedings of the 2016 Annual Meeting in Phonology. Bruce Hayes and Aaron Kaplan (in press) Zero-weighted constraints in Noisy Harmonic Grammar, to appear in *Linguistic Inquiry* squibs.

20. The MaxEnt calculations for a given candidate x

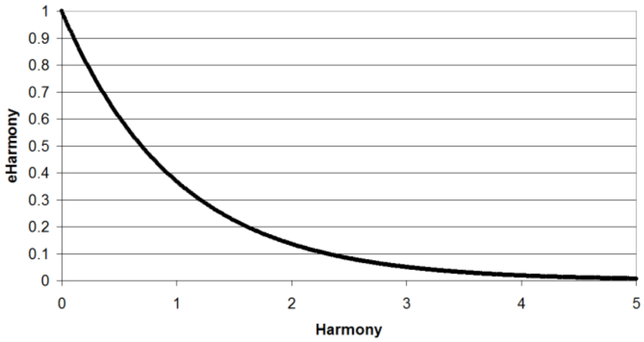
- A theme here is to try to claim that the calculations are commonsensical.
- Imagine that constraint violations are *evidence* for making a *decision.*

Compute this	*Name of what is computed*	*How and why it is computed*
1. $\Sigma_i\, w_i \mathrm{f}_i\,(x)$	Harmony (Smolensky 1986)	Multiply *x*'s violation counts for each constraint (designated $\mathbf{f}_{\boldsymbol{i}}\,\mathbf{(}\boldsymbol{x_j}\mathbf{)}$) by the weight of the constraint ($\boldsymbol{w_i}$), then add up the results across all constraints ($\boldsymbol{\Sigma_i}$). *All available evidence (i.e. constraint violations) bearing on a candidate is considered, in proportion to the constraint weights.*[4]
2. $\exp(-\Sigma_i\, w_i \mathrm{f}_i\,(x))$	eHarmony (Wilson 2014)[5]	Negate the harmony of *x,* then compute the function **exp( )** on the result, where exp(*x*) is a typographic convenience for e^x, $e \approx 2.72$. *As we consider a series of candidates with ever greater harmony penalties, their probabilities should descend not in linear fashion, but instead asymptote to zero (negative exponential curve) — certainty is evidentially expensive.*
3. $\Sigma_j \exp(-\Sigma_i\, w_i \mathrm{f}_i\,(x_j))$	Z, the "normalizing constant"	Compute the eHarmony of every candidate derived from the same input as *x* (*x* included), and sum these values.
4. $\frac{\exp(-\Sigma_i\, w_i \mathrm{f}_i\,(x))}{Z}$	Probability of *x*	Divide the eHarmony of *x* by Z (and similarly for all other candidates). *The probability of a candidate depends inversely on the probability of the candidates with which it competes. (Probability of all candidates must sum to one.)*

[4] This is *not* so for Optimality Theory, where decisions between candidates are made by the highest-ranking constraint that that distinguishes them, and all the evidence from other constraints is ignored.

[5] Wilson was joking in inventing this name (which also denotes a dating web site), but we feel it is quite helpful as a mnemonic.

21. Reference: graph plotting eHarmony against Harmony (from readings)



22. Back to the Japanese data

- Kawahara followed up his original study with an experiment.
  - Kawahara, Shigeto (2011) Japanese loanword devoicing revisited: A rating study. *Natural Language and Linguistic Theory*.
  - This is a rating study, which confirms the psychological reality of the lexical study.
- Since modeling ratings is tricky, let us simply model the lexical frequencies on which the ratings are (probably, mostly) based.
- Kawahara gives data that could (for pedagogical purposes only) be interpreted as the following percentages of devoicing:

babba-type words:	57.4
pabba-type words:	3.7
baba-type words:	assumed near zero

What sort of grammar could generate these numbers?

23. Setting up a MaxEnt grammar

- We can augment what we had earlier for non-stochastic Harmonic Grammar, adding a frequency column.

			Ident(voice)	Lyman	*bb
			0	0	0
babba	babba	436		1	1
	bappa	574	2		
	pabba	0	1		1
	pappa	0	3		
pabba	pabba	963			1
	pappa	37	2		
baba	baba	1000		1	
	bapa	0	1		
	paba	0	1		
	papa	0	2		

24. Obtaining MaxEnt probabilities in Excel

- My habit is to then add an additional bunch of columns, labeled “H”, “eH”, “Z”, and “p” (two more to come)
    - ➢ calculate **Harmony** with SUMPRODUCT()
    - ➢ calculate **eHarmony** with EXP(− …)
    - ➢ calculate **Z** with SUM( ) for each input, copied into each row
    - ➢ calculate **probability** by dividing eHarmony values by Z.

25. How are we doing? One simple way to check

- Calculate observed frequencies (i.e. normalize to between zero and one)
- Make a scattergram of observed vs. expected.
- Good to stretch it so the axes are of equal length.
- [☞ do this]
- You can also make a column with error size (use ABS() function)

26. Hand weighting

- No one does this in real life but it’s informative to try it [☞ let’s]

27. Software for maxent

- The UCLA “MaxEnt Grammar Tool” (https://linguistics.ucla.edu/people/hayes/MaxentGrammarTool/), which had a vogue, and might still be used for heavier-duty work.
- A really powerful package is on Tim Hunter’s GitHub site.
- However, for everyday analysis, I suggest using the Solver tool in Excel[6] (it is an “add-in”, but available to all for free)
    - ➢ Instructions at http://www.solver.com/excel-solver-how-load-or-start-solver?gclid=CIuGjPqd4NECFU5ufgodIpgJyQ

28. Nice aspects of doing MaxEnt in the Solver

- Your tableau is always on screen and easily edited.
- The software seems fine, works swiftly and precisely.
- No magic: you put in the formulae yourself and every step is visible.

THE BASIS OF WEIGHT-SETTING: SOME COMPUTER SCIENCE-EY CONCEPTS

29. Objective function

- A number
- The better the analysis, the higher (or lower, depending) this number will go.

[6] Thanks for former grad student Jesse Zymet, who discovered the Solver-MaxEnt combo.

- This often permits more accurate and reliable learning than under the alternative: “keep trying to change the grammar in ways that are likely to make it better” (cf. Boersma and Hayes 2001, *LI*, for Stochastic OT)

30. Likelihood

- The analysis assigns a probability to every datum.
- Multiply these probability across all data to get the **likelihood** of the data.
- High likelihood means good analysis: it concentrates probability on things that exist, thus minimizing probability squandered on what does not exist.

31. Log likelihood

- If you have too many data, computing likelihood will crash your software by producing a number that is too small for the software to represent.
- Instead, people use the *logarithm* of the likelihood: the sum of the log likelihood of each datum. This find the same answer.

32. Learning the best weights with Solver

- compute log probabilities with LN( )
- Slow way: make a column multiplying frequency by log probability
- The sum over this column is the log likelihood, our objective function
    - Short cut: SUMPRODUCT( ) over log probabilities and candidate frequencies.
- Ask the Solver to maximize the likelihood by changing the weights.
- Impose appropriate conditions on the Solver as you wish:
    - All weights non-negative (this is the default for the Solver; uncheck the box to remove)
    - Maximum on weights to prevent crashing.
    - Temporarily force a constraint weight to be zero, to see what the constraint is accomplishing.

33. Assess how well your model is performing.

- Form adjacent columns of Predicted and Observed
- Plot Predicted and Observed with a scattergram.
- Compute absolute value of difference (ABS( ) function) to find outliers.
- Later on, we’ll consider statistical testing.

34. Another way to assess a model

- What is its log likelihood, compared to other models?
- This turns out to be the basis for one statistical test, given below.

35. One way *not* to assess a model, I think

- **Correlation coefficient** (*r*)of predicted vs. observed
- It is quite possible for a model's predictions to be correlated with the real data, but not to predict them. [☞ explain]

36. Adapting model evaluation to the needs of linguistics — some qualitative common-sense principles

- If some candidate has frequency 0, and the model gives noticeably above 0, that is quite bad.
  - ➢ Similar to generating an ungrammatical outcome in classical nonstochastic grammar
- This implies that if a candidate has probability 1 (given its input), it would be bad for a stochastic model to fall short of 1.
- Otherwise, rough matching is probably fine for most data sets. (We aren't operating particle accelerators.)

37. Fixing up a defective model

- Look at the outliers on your scattergram.
  - ➢ If overgenerated, is there some constraint you missed that they violate?
- Trimming back unnecessary constraints

STATISTICAL TESTING

38. This is relatively new to linguistics

- The work in the 1970s of researchers like Labov or Peter Ladefoged, top quality in its day, did not use statistical testing to evaluate the quantitative claims.
- Only in this century has the field acquired statistical expertise, which is now widespread.
- Statistics itself is getting better, ANOVA replaced by mixed effects regression, now being replaced by Bayesian methods. Statistical models increasingly resemble theories.
- Here, we cover just a quick test; up to you to become an expert if you so choose.

39. The Likelihood Ratio Test

- Remember that log likelihood is our metric of model goodness.
- We can compare log likelihood of *nested* models; e.g. one is the same as the other with an extra constraint.

40. Procedure

- Record the log likelihood with constraint C included
- Take C out, re-fit the weights, and record the (probably lower) likelihood.
- Compute the difference and double it.

- Do chi-square test =CHIDIST(double-difference, 1) to get probability that improvement is not accidental (random variation in data).
- What to throw out is a matter of scientific outlook and journal reviewers. p < .05 is loosey-goosey at a social-science level, p < .00001 is stricter.

41. Multiple constraints

- You can test two constraints at once if you use =CHIDIST(double-difference, 2)
- Or *n*. This value is known as "degrees of freedom"

42. Strategies for constructing a grammar

- Build up from the bottom: add the constraint that best improves log likelihood. Keep going until no further constraints test as significant.
- Start at the top: keep deleting the least effective constraint until all remaining constraints test as significant.
- Example:
  - ➢ (2012) Bruce Hayes, Colin Wilson, and Anne Shisko, Maxent grammars for the metrics of Shakespeare and Milton. *Language*, Dec. 2012
- Widely cited reference:
  - ➢ Anderson, D., and K. Burnham. *Model selection and multi-model inference*. NY: Springer-Verlag.
- Model exploration is easier in R: try Kie Zuraw and Connor Mayer's new MaxEnt implementation in R (plugs into all the other goodies available in R).

43. A note on constraints with zero weights

- They are of course useless for explaining your data.
- But before you throw them away, try letting them take negative values (uncheck the box in Solver).
- It may be that the best weight is significantly negative — what you naively thought was a constraint is a credit.

TERMINOLOGY FOR MAXENT

44. MaxEnt's alter ego in statistics

- Multinomial logistic regression
- Same math, but meant as an effort to find causes of patterns in data, not as a model of linguistic competence
- Little kids, likewise, are making an effort to find patterns in the parental language data, so appealing to an effective statistical method is perhaps not far off base (see also work of Laurel Perkins, with Bayesian inference)

45. Why is MaxEnt (= "maximum entropy") so called?

- [Because mathematical people tend to be clueless when inventing terminology.]

- The spirit of the thing: grammar should assign equal probability — sheer randomness — for all cases where data do not tell it otherwise.
  - This randomness means that the grammar is neutral in its commitments where data are not available.
  - Example: rerun the Japanese problem with all zero data frequencies; you get all zero weights and equal probabilities for all candidates.

HARMONIC BOUNDING IN MAXENT

46. There is none

Every candidate gets at least some probability (look at formulas for why).

47. Philosophical position

- We can accept numbers like 10^{-50} as the equivalent of zero.
- Cf. speech error rates.

48. Consequences for candidate selection

- You can't just look at "all the plausible repairs for the Markedness violations of the maximally-faithful candidate."
- Try putting in /pa/ → [ba] in Japanese.

49. Views on MaxEnt and Harmonic Bounding

- It's a defect: it permits all sorts of "monsters" to be generated
  - Kaplan, Aaron. 2021. Categorical and gradient ungrammaticality in optional processes. *Language* 97:703–731.
  - Anttila, Arto, and Giorgio Magri. 2018. Does MaxEnt overgenerate? Implicational universals in Maximum Entropy grammar. In *Proceedings of the Annual Meetings on Phonology*, vol. 5.
  - Mai, Anna, and Eric Baković. 2020. Cumulative constraint interaction and the equalizer of OT and HG. *Proceedings of the 2019 Annual Meeting on Phonology.*
  - Magri, Giorgio. 2015. How to keep the HG weights non-negative: The truncated Perceptron reweighing rule. *Journal of Language Modelling* 3.2:345–375.
- Embrace non-harmonic bounding and use it as a crucial ingredient in analysis
  - Hayes, Bruce, and Russell Schuh. 2019. Metrical structure and sung rhythm of the Hausa rajaz. *Language* 95:e253–e299.
  - Hayes, Bruce, and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry* 39:379–440.
  - Kaplan, Aaron. 2011. Variation through Markedness Suppression. *Phonology* 28.3:331–370.