

Reasoning about rankings

24.964—Fall 2004
Modeling phonological learning

Class 8 (28 Oct, 2004)

Review of previous weeks

The *subset problem*

- Overt data are often (always?) compatible with more than one possible analysis, include more and less permissive grammars
- The learner must refrain from selecting a grammar that is too permissive, solely on the basis of ambiguous evidence
- Knowing how much to allow can be tricky

Another example

The *azba* language

pa	ap	sa	*za	apsa	aspa
ba	ab	as	*az	*abza	azba

- Phonemic voicing only among stops
- Regressive voicing assimilation

The *azba* language

Relevant constraints:

- \mathcal{M} : AGREE(voi)
- \mathcal{M} : *b (no voiced stops)
- \mathcal{M} : *z (no voiced frics)
- \mathcal{F} : IDENT(stop voi)_{Onset}
- \mathcal{F} : IDENT(stop voi)
- \mathcal{F} : IDENT(fric voi)_{Onset}
- \mathcal{F} : IDENT(fric voi)

The *azba* language

pa	ap	sa	*za	apsa	aspa
ba	ab	as	*az	*abza	azba

The problem: [azba] requires something to outrank *z, but what?

- Faith(fricative voicing) would allow [z] everywhere
- Faith(stop voi)/Ons + Agree(voi) is the right solution

The *azba* language

RCD, as usual, is hopeless

- Stratum 1: Agree, and all \mathcal{F}
- Stratum 2: *b, *z

The *azba* language

BCD also fails on this language:

- Starts by putting AGREE up top (only inviolable \mathcal{M} constraint)
- Then must choose among \mathcal{F} constraints
- \mathcal{F} : IDENT(stop voi)_{Onset}: *frees up 1*
- \mathcal{F} : IDENT(stop voi): *frees up 2*
- \mathcal{F} : IDENT(fric voi)_{Onset}: *inactive*
- \mathcal{F} : IDENT(fric voi): *frees up 1*

The *azba* language

Resulting ranking from BCD:

- Stratum 1: AGREE(voi)
- Stratum 2: IDENT(stop voi)
- Stratum 3: *b, *z
- Stratum 4: remaining \mathcal{F}

(What is wrong with this ranking?)

Brief aside: OTSoft

An extremely useful tool for playing with these algorithms

- Hayes, Tesar, and Zuraw (2003) OTSoft 2.1
- Available from: <http://www.linguistics.ucla.edu/people/hayes/otsoft/>
 - (Unfortunately, Windows only)
- Format of input files is (roughly) what we have been using so far
- Can output ranking arguments, pretty tableaux, Hasse diagrams, factorial typologies, etc.

The *azba* language

My implementation of LFCD from last time doesn't seem to work. Gets to:

- Stratum 1: AGREE(voi)
- Stratum 2: IDENT(stop voi)_{Onset}
- Stratum 3: *z

...and then says that there is apparently no ranking.

- This is due to a bug in how "stalemates" are detected
- Fixed LFCD.pl so that it only gives up when no constraints were rankable

The *azba* language

This now works:

- Stratum 1: AGREE(voi)
- Stratum 2: IDENT(stop voi)_{Onset}
- Stratum 3: *z
- Stratum 4: IDENT(stop voi)
- Stratum 5: *b
- Stratum 6: IDENT(fric voi), IDENT(fric voi)_{Onset}

(See Prince & Tesar, p. 26)

The *azba* language

Prince & Tesar's intuition: need to favor rankings that have the fewest additional consequences

- Here, $\text{IDENT}(\text{stop voi})_{\text{Onset}}$ allows fewer (unseen) possibilities than $\text{IDENT}(\text{fric voi})$, even if the two are not in a specificity relation
- But how can we know what the additional consequences will be, if we haven't seen the data that tests it?
 - (Worse: we will NEVER see the data that tests it)

The *azba* language

One possibility:

- The incorrect grammar that the BCD chooses allows both *azba* and *abza* (because AGREE and IDENT(stop voi) collude to create progressive voicing assimilation)
- Negative evidence: we're not getting *abza* as often as expected
- Indirect positive evidence: we're getting everything else more often than expected

Reasoning about observed and expected probabilities might be able to steer you away from an overly permissive ranking

Bayes' Theorem

$$P(A,B) = P(A) \times P(B|A) = P(B) \times P(A|B)$$

Bayes' Theorem

$$P(A,B) = P(A) \times P(B|A) = P(B) \times P(A|B)$$

$$P(B|A) = \frac{P(B) \times P(A|B)}{P(A)}$$

Bayes' Theorem

Where this is most useful:

- A is a datum (an observed event)
- B is a hypothesis about what caused or contributed to the event

A baseball example

Your friend: The Red Sox won the game last night!

You: Hmmmm, I wonder if it was a home or an away game?

A baseball example

Red Sox season stats (up until last night)

- Overall: 97-64 (won 60.2%)
- Home: 55-26 (won 67.9%)
- Away: 43-38 (won 53.1%)

81 home games, 81 away games

A baseball example

- $P(\text{home}|\text{won}) = P(\text{won}|\text{home}) \times P(\text{home}) / P(\text{won})$

$$P(\text{home}|\text{won}) = .679 \times .5 / .602$$

$$P(\text{home}|\text{won}) = .564$$

- What if the number of home and away games was not equal?

The cookies example

- There are two cookie jars: a white one and a black one
- The white one has 30 oatmeal cookies and 10 snickerdoodles
- The black one has 20 oatmeal cookies and 20 snickerdoodles

George takes a cookie from one of the jars and hands it to you. It's a snickerdoodle. Which jar did it probably come from?

The cookies example

- There are two cookie jars: a white one and a black one
- The white one has 30 oatmeal cookies and 10 snickerdoodles
- The black one has 20 oatmeal cookies and 20 snickerdoodles

George takes a cookie from one of the jars and hands it to you. It's a snickerdoodle. Which jar did it probably come from?

- $P(\text{black}|\text{snickerdoodle}) = \frac{P(\text{snickerdoodle}|\text{black}) \times P(\text{black})}{P(\text{snickerdoodles})}$

$$P(\text{black}|\text{snickerdoodle}) = \frac{.5 \times .5}{.375}$$

$$P(\text{black}|\text{snickerdoodle}) = .666666667$$

What can Bayes' Theorem do for us?

Brewer, Butler, Tucker & Hammond (2004) *SIMPLE? (SWOT presentation)

- Participants asked to judge acceptability of words with C, CC, and CCC onsets
- Example items:

C	CC	CCC
zɪlm	twɪlm	skɪlm
gɑɪθ	stɑɪθ	stɪɑɪθ
vɔɪk	floɪk	stɪoɪk
vɪʃ	kwɪʃ	skɪʃ
ḍzæntʃ	spæntʃ	stɪæntʃ
basp	plasp	stɪasp
ɪɑɪv	klɑɪv	stɪɑɪv

Brewer et al (2004)

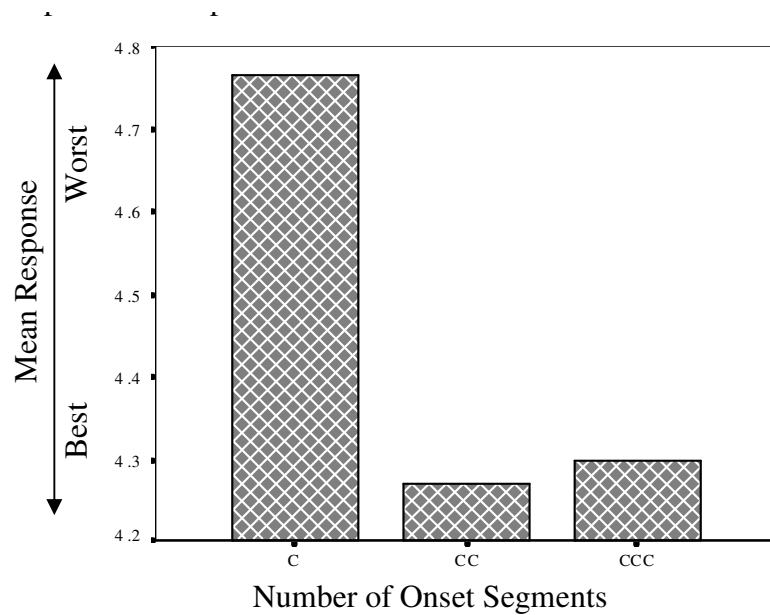
Procedure:

- Participants heard words over headphones
- Asked: how possible is this as an English word?
- Experiment 1: rated from 1 (possible) to 7 (not possible) (??)
- Experiment 2: yes/no (forced choice)

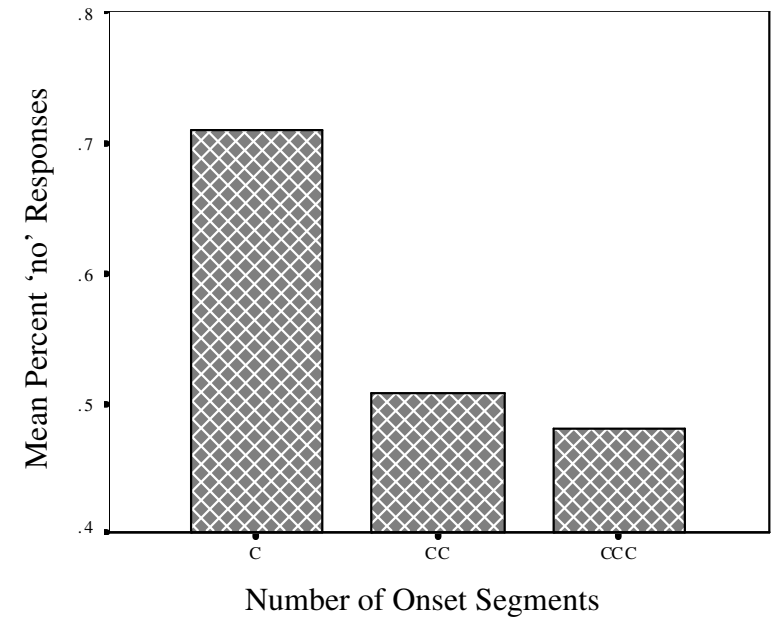
Brewer et al (2004)

Results:

Ratings:



Yes/no forced choice:



Brewer et al (2004)

The puzzle: why do people rate CC onsets as more likely English words?

- CC less frequent than C. Rough counts from monosyllables:

∅	173
C	2952
CC	1378

- CC more marked than C
- Can't just throw in a *SIMPLE constraint (makes bad typological predictions)

Brewer et al (2004)

Brewer et al's intuition:

- CC onset is more “unambiguously English” than C
- CC can only arise in a grammar where $\mathcal{F} \gg * \text{COMPLEX}$ (like English), whereas C can arise in any grammar

Brewer et al (2004)

A way of formalizing this intuition:

$$P(\text{English}|\text{C}) = \frac{P(\text{C}|\text{English}) \times P(\text{English})}{P(\text{C})}$$

$$P(\text{English}|\text{CC}) = \frac{P(\text{CC}|\text{English}) \times P(\text{English})}{P(\text{CC})}$$

Brewer et al (2004)

Let's assume some arbitrary numbers:

	English	Total
∅	.05	.05
C	.65	.80
CC	.30	.15

Brewer et al (2004)

A way of formalizing this intuition:

$$P(\text{English}|\text{C}) = \frac{P(\text{C}|\text{English}) \times P(\text{English})}{P(\text{C})}$$

$$P(\text{English}|\text{C}) = \frac{.65 \times ??}{.80}$$

$$P(\text{English}|\text{C}) = .8125 \times \textit{English bias}$$

$$P(\text{English}|\text{CC}) = \frac{P(\text{CC}|\text{English}) \times P(\text{English})}{P(\text{CC})}$$

$$P(\text{English}|\text{CC}) = \frac{.30 \times ??}{.15}$$

$$P(\text{English}|\text{CC}) = 2 \times \textit{English bias}$$

What can Bayes' Theorem do for us?

Zuraw (2000) UCLA dissertation:

- Tagalog nasal substitution

p ighatíʔ	‘grief’
pa-mim m ighatíʔ	‘being in grief’

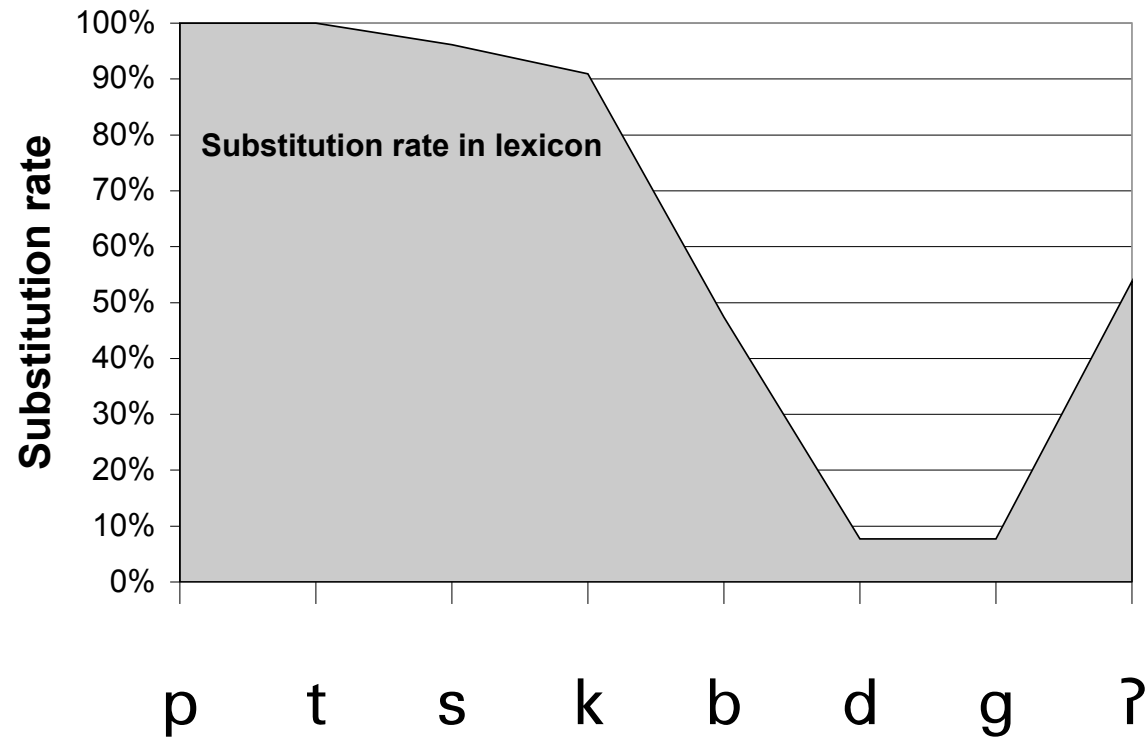
vs.

p oʔók	‘district’
pam- p oʔók	‘local’

(See Zuraw, chapter 2, ex. 7 for more examples)

Zuraw (2000)

Exceptions are not distributed evenly for all consonants



Zuraw (2000)

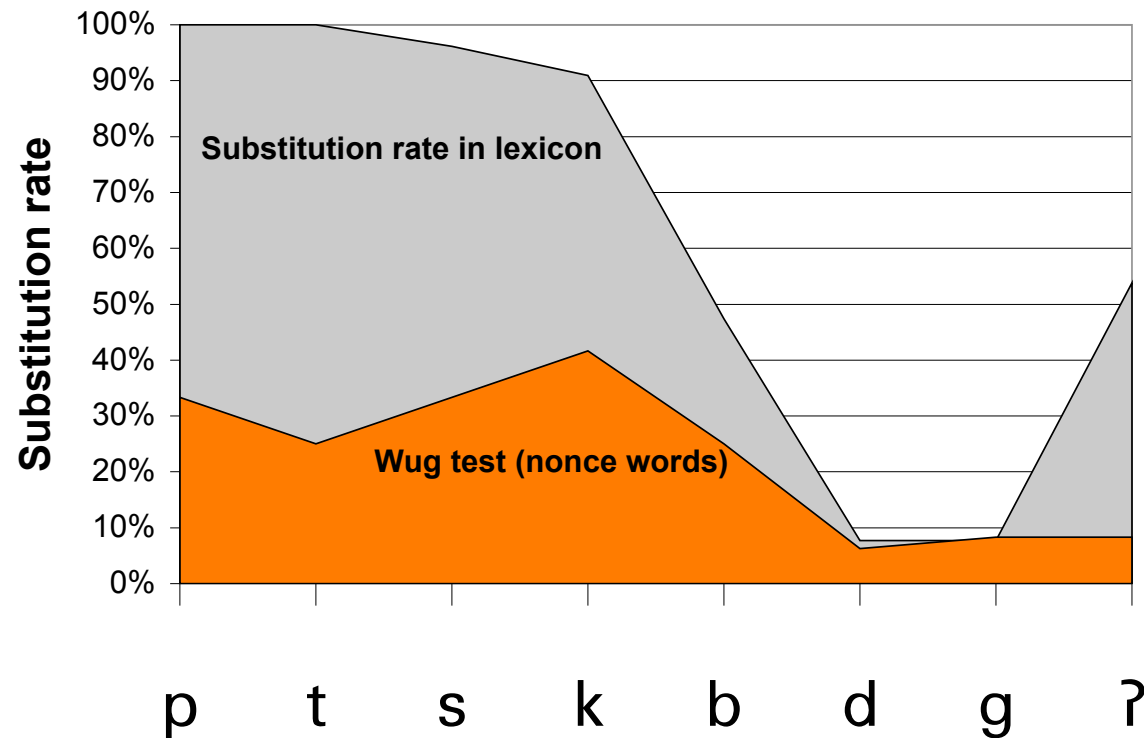
An experiment to test the productivity of nasal substitution

Pagbubugnat ang trabaho niya. Siya ay _____
bugnat-ing TOP job POSS he/she

His/her job is to *bugnat*. He/she is a _____

Zuraw (2000)

Results: productivity across consonants generally mirrors rate of substitution for existing words



- Overall lower rate of application, however (nonce words tend to alternate less)

Zuraw (2000)

The challenge:

- Nasal substitution has plenty of exceptions.
- In order for exceptions to surface faithfully, we need $\mathcal{F} \gg \mathcal{M}$
- Yet we need the grammar to do more than this; needs to be able to apply nasal substitution to a certain extent for novel forms, depending on the C involved
 - Existing (established) words are produced faithfully, however they are in the input data
 - For novel words, \mathcal{F} is irrelevant, so decision falls to lower ranked \mathcal{M} constraints
 - This means that speakers must somehow learn those rankings, even though for existing words, they are “hidden” by \mathcal{F}

Zuraw (2000)

The proposal:

- Learners can learn rankings of \mathcal{M} constraints if they don't treat all data as listed forms
- That is, if they assume their grammar needs to produce it productively, they will need to rank constraints to get the right answer

Zuraw (2000)

Reasoning about this: (see p. 87)

Given that the listener has heard [mamumuntol], we can calculate...

- $P(/man/ + /R_{CV}/ + /puntol/ \mid [mamumuntol])$
 = probability that speaker produced form compositionally, and grammar yielded [mamumuntol]
- $P(/mamumuntol/ \mid [mamumuntol])$
 = probability that speaker had this as a listed form from /muntol/
- $P(/mampupuntol/ \mid [mamumuntol])$
 = probability that speaker had listed form from /puntol/, but grammar applied nasal substitution

Zuraw (2000)

Using Bayesian inversion to calculate these:

- $P(/man/ + /R_{CV}/ + /punto/ \mid [mamumunto])$
$$= \frac{P([mamumunto] \mid /man + R_{CV} + punto/) \times P(/man + R_{CV} + punto/)}{P([mamumunto])}$$

- $P(/mamumunto/ \mid [mamumunto])$
$$= \frac{P([mamumunto] \mid /mamumunto/) \times P(/mamumunto/)}{P([mamumunto])}$$

- $P(/mampupunto/ \mid [mamumunto])$
$$= \frac{P([mamumunto] \mid /mampupunto/) \times P(/mampupunto/)}{P([mamumunto])}$$

(Luckily, $P([mamumunto])$ is constant across all three; comparison comes down to likelihood of nasal sub. in various contexts, and relative proportion of /p/ vs /m/ roots in lexicon)

Zuraw (2000)

Where this leads:

- Uncertainty by the learner about whether to simply list a form, or rerank the grammar so that it is productively derivable, cause some reranking to happen
- The result is a grammar with “subterranean” $\mathcal{M} \gg \mathcal{F}$ rankings, even though in general, they are hidden by higher-ranked \mathcal{F} constraints

(The details are quite complex, because the process is gradient and requires some heavy duty math; we’ll look some more at this next week)

What can Bayes' Theorem do for us?

Constraint ranking:

- Is [ta] evidence for $\text{MAX}(\text{C}) \gg * \text{CODA}$?

- $P(\text{MAX}(\text{C}) \gg * \text{Coda} \mid [\text{ta}]) = \frac{P([\text{ta} \mid \text{MAX}(\text{C}) \gg * \text{Coda}] \times P(\text{MAX}(\text{C}) \gg * \text{Coda})}{P([\text{ta}])}$

- $P(* \text{Coda} \gg \text{MAX}(\text{C}) \mid [\text{ta}]) = \frac{P([\text{ta} \mid * \text{Coda} \gg * \text{MAX}(\text{C})]) \times P(* \text{Coda} \gg \text{MAX}(\text{C}))}{P([\text{ta}])}$

Albro (2000)

PEDPRA: A probabilistic error-driven phonotactic ranking algorithm

- Idea: ranking arguments are determined probabilistically, using Bayesian inference

- $$P(\text{ranking}|\text{datum}) = \frac{P(\text{datum}|\text{ranking})P(\text{ranking})}{P(\text{datum})}$$

How do we know these probabilities?

Albro (2000)

$$P(\text{ranking}|\text{datum}) = \frac{P(\text{datum}|\text{ranking})P(\text{ranking})}{P(\text{datum})}$$

- $P(\text{ranking})$: by the tenets of OT, constraints are freely rerankable. Thus, $P(C1 \gg C2) = P(C2 \gg C1) = .5$
- $P(\text{datum}|\text{ranking})$: ??? (how likely this ranking is to produce this datum)
- $P(\text{datum})$: ??? (how likely is a datum in the world?)

Albro (2000)

The overall approach:

- Initial estimation phase in which the learner explores the predictions of different rankings
 - “Learn how to learn phonotactics (find out the likelihood of getting evidence for various ranking pairs, given randomized input grammars)”
- Then look at actual data, seeing what rankings they seem to imply
- Finally, evaluate the evidence for these rankings, compensating for overall differences in probability that particular rankings are needed to explain particular data

Albro (2000)

Parameter estimation (estimating the Bayesian priors)

1. Pick an input at random
2. Pick a ranking at random
3. See what the grammar generates for that input/ranking
4. Then think what you would do if you had actually heard that as an input datum
 - Treat output of (3) as input (IN=OUT), and find a ranking that would produce it

- This increases the probability that this ranking could be a necessary one for actual data in the world
- Often, the rankings needed to derive the form are *not* the ones you started out with by random picking

Albro (2000)

Learning phase: similar, but with real words

- For each datum, treat as input, and see what rankings are necessary
- Make a note of each ranking that seems necessary while learning

Albro (2000)

Bayesian inference:

- Calculate probabilities of pairwise ranking arguments, based on their necessity in the training data, but moderated by their overall probability
 - If a ranking often seemed necessary in the input data, and it's not a probable ranking in the world at large, that's good evidence that it's REALLY needed
 - If you sometimes thought you needed a particular ranking, but that ranking is one that pops up all the time, it's not such good evidence

Is this what Albro is, in fact, suggesting? (see p. 5) Is it the right way to think about things?

For next time

- Download OTSoft, and run your *azba* file through it, trying RCD, BCD, and LFCD
 - Augment your *azba* file to include some test forms (hypothetical inputs and possible candidates, but no form marked as winner) to test the performance of the grammar on novel words

- Readings:
 - Boersma and Hayes (2001) Empirical tests of the Gradual Learning Algorithm
 - ◇ Try the GLA on *azba*, too (using OTSoft)
 - Zuraw (2000) Patterned Exceptions in Phonology chapters 2-3