Class 2, 6/24/13: Stochastic OT/Modeling
Hungarian Vowel Harmony
1. **Course website is moved!**
   
   - http://www.linguistics.ucla.edu/people/hayes/LSA2013/
   - If you don’t remember Google my name (and ignore the folk singer and the swimmer).
2. How is software/exercise going?

- Windows
- Mac
3. **Handing in last time’s exercise**

- Email it to me, or (better) print it out if you have a printer.
4. Exercise for next time

- At end of these slides
5. Reading for next time

- Download from: http://www.linguistics.ucla.edu/people/hayes/HungarianVH/HayesLondeHungarian2006.pdf
6. Where we are

- Virtues of computation
  - Long term theoretical goal: the Simulated Child
  - Ability to deal with full-scale analysis
- A simple ranking algorithm – Recursive Constraint Demotion – applied to Turkish
- Variation
  - Token/the research literature in sociolinguistics
  - Type/The Law of Frequency Matching
7. Brief followup on the Law of Frequency Matching

- Frequency-matching is known to be a common ability in animals (Gallistel 1990, ch. 11)\(^1\); and in humans for nonlinguistic tasks (Hasher and Zacks 1984).\(^2\)
- The Story of the Ducks and the Fish (Gallistel)

---

\(^1\) Gallistel, Charles (1990) The Organization of Learning, MIT Press.
8. Theoretical and computational resources we need to cover variation effectively

- We want a **grammar formalism** that can generate variation
- We want an effective **learning algorithm** that can learn such grammars from data;
  - … just as Recursive Constraint Demotion does for single-output phenomena
- We want our system to handle **large amounts of data**
  - scaling up to real-life scale
  - data will come from searching **electronic corpora** — ideally approximating the language learner’s experience
9. **Example we will work with today**

- Hungarian vowel harmony
- We will model simple corpus and experimental data from Hungarian with various formal frameworks
  - today, Stochastic OT
  - next time, some versions of Harmonic Grammar
STOCHASTIC OT AND THE GRADUAL LEARNING ALGORITHM
10. Basic references

• Invented by Paul Boersma in his 1998 (published) dissertation.
• First applied to phonology in Boersma and Hayes (2001, *Linguistic Inquiry*)
11. Core idea

- **We refrain from knowing the rankings for sure** — we only have a probabilistic knowledge of them.
- So, we can run the “uncertain” grammar over and over, many times
  - Each time is called an **evaluation time**
  - Each time, we guess what the grammar is …
  - Assuming this guess, we see what it generates
- Once we done this enough, we have an approximate description of how the grammar depicts variation.
- … and we can check the result against variable data from people.
12. How you might start out on this problem

- Designing a grammar:
  - Take every pair of constraints A and B, and assign a probability that A dominates B.
- But what would guarantee consistency across all such rankings?
13. The continuous ranking scale (Boersma 1997)

\[ \text{C}_1 \quad \text{C}_2 \]

(high ranked) \quad (low ranked)

→ \( \text{C}_1 \) outranks \( \text{C}_2 \)

- \( \text{C}_1 \) is said to have a high ranking value, \( \text{C}_2 \) a low one.
14. Jiggling

- At each evaluation time, jiggle the positions of C1 and C2, obtain **selection points**.
- Then you sort by selection point and pick a winner following the normal principles of OT.
15. How to jiggle: use the Gaussian distribution

- $\sigma =$ standard deviation
- $\mu =$ mean
16. Intuitive meaning of using the Gaussian

- Small jiggles are frequent; large ones rare; no upper limit in principle
17. Relative ranking probability from overlapping distributions

- Quoting from Boersma and Hayes:

\[(6) \quad \text{Overlapping ranking distributions}\]

In (6), the ranking values for \(C_1\) and \(C_2\) are at the hypothetical values 87.7 and 83.1. Since the evaluation noise is 2.0, the normal distributions assigned to \(C_1\) and \(C_2\) overlap substantially. While the selection points for \(C_1\) and \(C_2\) will most often occur somewhere in the central “hump” of their distributions, they will on occasion be found quite a bit further away. Thus, \(C_1\) will outrank \(C_2\) at evaluation time in most cases, but the opposite ranking will occasionally hold. Simple calculations show that the percentages for these outcomes will tend towards the values 94.8\% (\(C_1 \gg C_2\)) and 5.2\% (\(C_2 \gg C_1\)).
18. Part of a spreadsheet

- [http://www.linguistics.ucla.edu/people/hayes/GLA/RankingValuesToProbabilities.xls](http://www.linguistics.ucla.edu/people/hayes/GLA/RankingValuesToProbabilities.xls)

<table>
<thead>
<tr>
<th>Difference in ranking value</th>
<th>Probability higher outranks lower</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>0.1</td>
<td>0.51</td>
</tr>
<tr>
<td>0.5</td>
<td>0.57</td>
</tr>
<tr>
<td>1</td>
<td>0.64</td>
</tr>
<tr>
<td>5</td>
<td>0.96</td>
</tr>
<tr>
<td>10</td>
<td>0.9998</td>
</tr>
<tr>
<td>50</td>
<td>1.000000000</td>
</tr>
</tbody>
</table>
Ranking Probability Resulting from Differences in Ranking Value

- X-axis: Difference in ranking value
- Y-axis: Ranking probability

Graph shows a curve that increases as the difference in ranking value increases, indicating higher probability with greater difference.
19. Restrictiveness of stochastic OT

- You cannot substantially overlap two constraints that don’t overlap with each other.
- We in fact will see later in the course that this is a serious problem with this theory!
LEARNING STOCHASTIC OT GRAMMARS: THE GRADUAL LEARNING ALGORITHM
20. How could these grammars be learned?

- Boersma (1998) invented a scheme
- He called it the **Gradual Learning Algorithm**.
21. The core of the algorithm

- Live surrounded by the language, with random inputs.
  ➢ Or simulate this situation computationally…
- **Hear a datum** — maybe the only output for this input, perhaps one of a set of free variants.
- Grab the UR (how!?) and **generate your own output** with the grammar you’ve got so far.
- If it matches, do nothing
- If it fails to match, treat the heard datum as the “**winner**”, your own output as the “**loser**”.
- **Nudge** winner-preferrers up a bit, loser-preferrers down a bit.
22. What does this procedure do?

- It *tends* to produce grammars where:
  - Always-winners always win
  - Always-losers always lose
  - Free variation: frequency matching (which is what we want)

- For whether the original promise of this algorithm has been fulfilled, see later on in this course.
23. How much to nudge?

- A little bit
- Useful to start with a *larger* little bit (2) then take it down to refine the outcome (.01).
- The stipulated amount of nudging at any given point is called the *noise*
24. Tiny GLA demo

- Let’s do an abstract case

<table>
<thead>
<tr>
<th>Input</th>
<th>Candidate 1</th>
<th>Candidate 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>667</td>
<td>333</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

- Run it in OTSoft.
HUNGARIAN VOWEL HARMONY
25. Sources

26. Hungarian vowels

Back \([u, u:, o, o:, ɔ, a:]\) abbreviated “B”

Front rounded \([y, y:, ȯ, ȯ:]\) abbreviated “F”

Front unrounded, often called “neutral” \([i, i:, e:, ɛ]\) abbreviated “N”
27. Dative suffix

- Is representative in its behavior
- Allomorphs: back [-nɔk] and front [-nɛk]
28. Closest vowel back: back suffixes

BB  [ɔblɔk-nɔk]  ‘window-dat.’
NB  [biːroː-nɔk]  ‘judge-dat.’
FB  [glykoːz-nɔk]  ‘glucose-dat.’
29. Closest vowel front rounded: front suffixes

F  \( [yʃt\text{-nɛk}] \) ‘cauldron-dat.’

NF \( [sɛmøltʃ\text{-nɛk}] \) ‘wart-dat.’

BF \( [ʃofør\text{-nɛk}] \) ‘chauffeur-dat.’
30. \( F + N^* \): front suffixes

\[
\begin{align*}
\text{FN} & \quad [fy:ser-n\emptyset k] \quad \text{‘spice-dat.’} \\
\text{FNN} & \quad [\emptyset :rize\emptyset -n\emptyset k] \quad \text{‘custody-dat.’}
\end{align*}
\]
31. Zones of Variation

- Individual stems vary in the kind of harmony they take—you must memorize.
- There are also “vacillators”: stems for which either front or back suffixes are acceptable, and occur in various proportions.
- The zones: words ending in BN or BNN, plus [N] and marginally, [NN]
### 32. Examples: lexical arbitrariness of harmony within the zones of variation (BN)

<table>
<thead>
<tr>
<th>Word (([o]+[e:]))</th>
<th>Gloss</th>
<th>Google hits (Sept. 2008)</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>doménnak [dome:n-nök]</td>
<td>‘domain (on Web)-dat.’</td>
<td>5</td>
<td>2.1</td>
</tr>
<tr>
<td>doménnek [dome:n-nék]</td>
<td></td>
<td>234</td>
<td>97.9</td>
</tr>
<tr>
<td>bohémnak [bohe:m-nök]</td>
<td>‘easy-going-dat.’</td>
<td>433</td>
<td>24.4</td>
</tr>
<tr>
<td>bohémnak [bohe:m-nök]</td>
<td></td>
<td>1,340</td>
<td>75.6</td>
</tr>
<tr>
<td>honvédnak [honve:d-nök]</td>
<td>‘Hungarian soldier-dat.’</td>
<td>8,820</td>
<td>74.1</td>
</tr>
<tr>
<td>honvédnak [honve:d-nék]</td>
<td></td>
<td>3,084</td>
<td>25.9</td>
</tr>
<tr>
<td>poénnak [poe:n-nök]</td>
<td>‘punch line-dat.’</td>
<td>56,400</td>
<td>99.9</td>
</tr>
<tr>
<td>poénnek [poe:n-nék]</td>
<td></td>
<td>36</td>
<td>0.1</td>
</tr>
</tbody>
</table>

- The number of forms that have free variation is small — most settle on the ends of the frequency spectrum.
33. Corpus study

- Hayes and Londe (2006) did a Google survey, about 9,000 words, counting both -nɔk and -nɛk
34. Statistical patterns within the zones of variation

- **Height Effect**: the higher the last N vowel in BN, BNN, the more you get front harmony.
- **Count Effect**: more front harmony in BNN than BN.
35. Productivity of Height and Count Effects: Hayes and Londe’s wug test

![Graph showing the productivity of height and count effects with symbols representing different sounds and their proportions back in the corpus and wug test.](image-url)
36. Where we want to go

- Go to a new corpus and verify the basic numbers.
- Use **pivot tables** in Excel to count the numbers efficiently.
- Create an OTSoft input file.
- Learn a stochastic OT model using the GLA
- Run the model on the data.
- Grab the results, interpret them graphically and assess how we are doing.
37. The simple data we will be modeling

- Basically the pattern reported in Hayes/Londe
- But we have a bigger corpus, the Hungarian WebCorpus (Halácsy et al. 2004, Kornai et al. 2006), and searched all inflected forms (and a few derived forms) of the same stems (not just the dative).
- Searching done mainly by Kie Zuraw, based on a Hungarian morphology-generator by me.
- I counted from a digested version prepared by Kie, using pivot tables in Excel.
38. The issue of what to count: tokens or types?

- In experiments that test what speakers have learned from the lexicon, there is a very consistent outcome: what matters is
  - Not how frequent (in usage) are the individual words are that make up the pattern (token frequency)
  - This means, e.g., that very common words have much less influence on intuitions than you might think; [ʃɛɾ] ~ ??[ʃɛd]
• Rather, how well populated with words are the rival patterns (Here, we have a bigger corpus, the Hungarian WebCorpus (Halácsy et al. 2004, Kornai et al. 2006), and searched all inflected forms (and a few derived forms) of the same stems (not just the dative).

• So, each word counts as just one!

39. How to count items in a raw text corpus: a tricky issue

- What if a single word vacillates? Proposed answer:
  - Take a vote. E.g. 60 front, 40 back is counted as “.6 of a word” front; “.4 of a word back”
  - This is **token-weighted type frequency**
  - This is easy to do in Excel, which I did.
  - Take a quick look at the spreadsheet.
40. The Hungarian counts: how many data do we have?

<table>
<thead>
<tr>
<th></th>
<th>e</th>
<th>é</th>
<th>l</th>
<th>Grand Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>BN</td>
<td>1039</td>
<td>575</td>
<td>1459</td>
<td>3073</td>
</tr>
<tr>
<td>BNN</td>
<td>208</td>
<td>73</td>
<td>251</td>
<td>532</td>
</tr>
<tr>
<td>Grand Total</td>
<td>1247</td>
<td>648</td>
<td>1710</td>
<td>3605</td>
</tr>
</tbody>
</table>
41. **What are the percentage back?**

<table>
<thead>
<tr>
<th></th>
<th>e</th>
<th>é</th>
<th>i</th>
<th>Grand Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>BN</td>
<td>0.106</td>
<td>0.876</td>
<td>0.973</td>
<td>0.661</td>
</tr>
<tr>
<td>BNN</td>
<td>0.000</td>
<td>0.247</td>
<td>0.239</td>
<td>0.147</td>
</tr>
<tr>
<td>Grand Total</td>
<td>0.088</td>
<td>0.805</td>
<td>0.865</td>
<td>0.585</td>
</tr>
</tbody>
</table>
42. What are token-weighted front counts?

<table>
<thead>
<tr>
<th></th>
<th>e</th>
<th>é</th>
<th>l</th>
<th>Grand Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>BN</td>
<td>929.1</td>
<td>71.6</td>
<td>40.1</td>
<td>1040.8</td>
</tr>
<tr>
<td>BNN</td>
<td>208.0</td>
<td>55.0</td>
<td>191.0</td>
<td>454.0</td>
</tr>
<tr>
<td>Grand Total</td>
<td>1137.1</td>
<td>126.5</td>
<td>231.1</td>
<td>1494.7</td>
</tr>
</tbody>
</table>
43. And back counts?

<table>
<thead>
<tr>
<th></th>
<th>e</th>
<th>é</th>
<th>I</th>
<th>Grand Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>BN</td>
<td>109.9</td>
<td>503.4</td>
<td>1418.9</td>
<td>2032.2</td>
</tr>
<tr>
<td>BNN</td>
<td>0.0</td>
<td>18.0</td>
<td>60.0</td>
<td>78.0</td>
</tr>
<tr>
<td>Grand Total</td>
<td>109.9</td>
<td>521.5</td>
<td>1478.9</td>
<td>2110.3</td>
</tr>
</tbody>
</table>
44. A labor-saving stipulation

- There are a lot of stems whose last vowel is B
- Every one of them always takes back suffixes (the remainder are probably typos).
- Since the data aren’t at hand, let just make up a number: 4000/4000
ANALYSIS OF HUNGARIAN
45. Constraint system

- Let’s keep it simple
- LOCAL I: don’t disagree in backness with a high front rounded vowel in an adjacent syllable.
- LOCAL e: don’t disagree in backness with e: in an adjacent syllable.
- LOCAL ɛ: don’t disagree in backness with e: in an adjacent syllable.
- LOCAL NN: don’t disagree in backness with an adjacent sequence of two front vowels.³
- LOCAL B: don’t disagree in backness with a back vowel in an adjacent syllable.

• DISTAL B: don’t disagree in backness with a back vowel in any preceding syllable.
## 46. OTSoft file

<table>
<thead>
<tr>
<th></th>
<th>Local B</th>
<th>Distal B</th>
<th>Local I</th>
<th>Local e:</th>
<th>Local E</th>
<th>Local NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>back</td>
<td>4000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>front</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BI</td>
<td>back</td>
<td>1418.9</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>front</td>
<td>40.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Be:</td>
<td>back</td>
<td>503.4</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>front</td>
<td>71.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BE</td>
<td>back</td>
<td>109.9</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>front</td>
<td>929.1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>BNI</td>
<td>back</td>
<td>60.0</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>front</td>
<td>40.1</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Bne:</td>
<td>back</td>
<td>18.0</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>front</td>
<td>73</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>BNE</td>
<td>back</td>
<td>0.0</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>front</td>
<td>208</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>
47. Let’s see how we can do

- Running the input file in OTSoft.
48. Let’s interpret how well we are doing

- As an aid to intuition, we’ll make line graphs and a scattergram in Excel.
- We’ll discuss model evaluation more carefully later on.
49. A case where the model isn’t matching the data very well

- Look at the BNN forms.
- But then, look at the BNN forms in the wug test results above (35)
- As a model of Hungarians, it seems to be doing rather well.
50. Understanding the probabilities

- Converting differences in ranking values to probability.
51. Some humility

- We’re predicting 7 proportional outcomes.
- We’re using 6 numbers (5 really, since the overall level is arbitrary) to do it.
- If we don’t do well we’d better rethink our approach!
THEORY: WHAT ABOUT INDIVIDUAL WORDS?
52. Desiderata

- For particular invariant forms like poénnak, we want Faithfulness to force their use.
- For novel forms (e.g. never heard with suffix, or wug), we want a stochastic grammar to generate frequency-matching behavior.
- Listing cannot in general ride roughshod over grammar, since some possibilities aren’t even listable. Examples: B-stem with -nek, F stem with -nak, datives that change consonants of the stem.
- Say something about the (relatively few) doublet forms, where there is variation within a single stem.
53. Zuraw’s theory: the dual listing/generation model

- Source: Zuraw 2001 dissertation, cited earlier
- Words are memorized—even inflected ones—as they are heard.
- The constraint ranking needed: Faithfulness is quite high, forcing the use of particular memorized forms.
- But a stochastic grammar is nonetheless created from the data — treating them “as if” they were free variation data.
- I.e.: memorize, but be ready to project.
54. Not everything is memorizable?

- If Local B strongly outranks Faithfulness, then nonsense like “ab-nek” cannot surface; it would have to become [ab-nak].
55. What about doublets?

- It would be natural to assign them doublet lexical entries.
- These entries must themselves be somehow probabilistic, to reflect the variation seen above in (32).
56. **Next time**

- We will (politely) trash the GLA: convergence failures, gross errors of ranking
- We will then cover the basics of Harmonic Grammar and discuss a couple algorithms using it that seem to work better.
57. Exercise

- Take this OTSoft file and run it with OTSoft:

<table>
<thead>
<tr>
<th></th>
<th>Local</th>
<th>Distal</th>
<th>Local I</th>
<th>Local E</th>
<th>Local NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>back</td>
<td>4000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>front</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>BI</td>
<td>back</td>
<td>1418.9</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>front</td>
<td>40.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Be:</td>
<td>back</td>
<td>503.4</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>front</td>
<td>71.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BE</td>
<td>back</td>
<td>109.9</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>front</td>
<td>929.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNI</td>
<td>back</td>
<td>0</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>front</td>
<td>100</td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
What did I do to alter the frequencies?
What happens when you run the GLA?
What point from the lecture do you think I was trying to emphasize?