A. Overview

(1) The plan:
   a. HG “double” tradeoff example
   b. how Maxent grammars work
   c. some simple examples
   d. learning constraints
   e. etc.

B. HG

(2)  
<table>
<thead>
<tr>
<th>/tap/</th>
<th>Faith</th>
<th>HaveOns</th>
<th>NoCoda</th>
</tr>
</thead>
<tbody>
<tr>
<td>tap</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>ta&lt;p&gt;</td>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(3)  
<table>
<thead>
<tr>
<th>/a/</th>
<th>Faith</th>
<th>HaveOns</th>
<th>NoCoda</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ta</td>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(4)  
<table>
<thead>
<tr>
<th>/ap/</th>
<th>Faith</th>
<th>HaveOns</th>
<th>NoCoda</th>
</tr>
</thead>
<tbody>
<tr>
<td>ap</td>
<td></td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Taap</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a&lt;p&gt;</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ta&lt;p&gt;</td>
<td>**</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(5)  
\[ H(tap) > H(ta<p>) \]
\[ 0F + 0H - 1N > -1F + 0H + 0N = \]
\[ -1N > -1F = \]
\[ 1F - 1N > 0 \Rightarrow \]
\[ 1F - 1N \geq 1 \]
(6) $H(a) > H(Ta)$
\[
0F - 1H + 0N > -1F + 0H + 0N = \\
-1H > -1F = \\
1F - 1H > 0 \quad \Rightarrow \\
1F - 1H \geq 1
\]

(7) $H(Tap) > H(ap)$
\[
-1F + 0H - 1N > 0F - 1H - 1N = \\
-1F - 1N > -1H - 1N = \\
-1F + 1H > 0 \quad \Rightarrow \\
-1F + 1H \geq 1
\]

(8) $H(Tap) > H(a<p>)$
\[
-1F + 0H - 1N > -1F - 1H + 0N = \\
-1F - 1N > -1F - 1H = \\
1H - 1N > 0 \quad \Rightarrow \\
1H - 1N \geq 1
\]

(9) $H(Tap) > H(Ta<p>)$
\[
-1F + 0H - 1N > -2F + 0H + 0N = \\
-1F - 1N > -2F = \\
1F - 1N > 0 \quad \Rightarrow \\
1F - 1N \geq 1
\]

(10) Minimize $1F + 1H + 1N$
\[
1F - 1N \geq 1 \\
1F - 1H \geq 1 \\
1H - 1N \geq 1 \\
1F \geq 1 \\
1H \geq 1 \\
1N \geq 1
\]

(11) This is implemented in `hg2.R` which produces the following weights:
\[
w(F) = 3 \\
w(H) = 2 \\
w(N) = 1
\]
C. Maxent

(12) The score of a phonological representation $x$, denoted $h(x)$, is

$$h(x) = \sum_{i=1}^{N} w_i C_i(x)$$

(13) For any $C_i$ in $(C_1, C_2, \ldots, C_N)$: $w_i \geq 0$

(14) Given a phonological representation $x$ and its score $h(x)$ under a grammar, the maxent value of $x$, denoted $P^*(x)$, is

$$P^*(x) = \exp(-h(x))$$

(15)

<table>
<thead>
<tr>
<th>$x$</th>
<th>$*#V$</th>
<th>$*#C$</th>
<th>Score $H(x)$</th>
<th>Maxent value $P^*(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV</td>
<td>3·0</td>
<td>2·0</td>
<td>0 + 0 = 0</td>
<td>$\exp(-0) = 1$</td>
</tr>
<tr>
<td>CVC</td>
<td>3·0</td>
<td>2·1</td>
<td>0 + 2 = 2</td>
<td>$\exp(-2) = .14$</td>
</tr>
<tr>
<td>V</td>
<td>3·1</td>
<td>2·0</td>
<td>3 + 0 = 3</td>
<td>$\exp(-3) = .05$</td>
</tr>
</tbody>
</table>

(16) This just changes the scale and orientation:
- `plot(seq(1,10),type='l')`
- `plot(exp(-seq(1,10)),type='l')`

(17) “We suppose an infinite set $\Omega$ consisting of all universally possible phonological surface forms.”

(18) Given a phonological representation $x$ and its maxent value $P^*(x)$, the probability of $x$, denoted $P(x)$, is

$$P(x) = \frac{P^*(x)}{\sum_{y \in \Omega} P^*(y)}$$

(19) Given a maxent grammar and a set $D$ of observed data, the probability of $D$ under the grammar is:

$$P(D) = \prod_{x \in D} P(x)$$
(20) Convert to logs:

\[ \log(P(D)) = \sum_{x \in D} \log(P(x)) \]

(21) Hayes & Wilson 2008’s learning problem: maximize \( \log(P(D)) \).

D. Some simple examples

(22) An example from English:
   a. find a weight for NoCoda
   b. data from newdic.txt

(23) There are 3685 monosyllabic words in the corpus. 3477 of them end in a consonant and 208 do not.

(24) The \texttt{getvalue()} function in the \texttt{maxent1.R} script invoked on the \texttt{mono1} dataframe allows us to set different weights for the NoCoda constraint and see the effect on \( \log(P(D)) \). What is the best weight for NoCoda? What is the best value for \( \log(P(D)) \)?

(25) Another example: onsets and codas in monosyllables:

<table>
<thead>
<tr>
<th>type</th>
<th>count</th>
<th>HaveOns</th>
<th>Nocoda</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVC</td>
<td>3344</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>CV</td>
<td>203</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VC</td>
<td>133</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>V</td>
<td>5</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

The \texttt{mono2} dataframe has these data.

(26) The \texttt{learn()} function will learn weights for a simple constraint system. (Next week we’ll look at the Maxent Grammar Tool, which does the same thing more robustly for more complex systems.)
(27) Running the examples:
   a. download relevant files to your desktop
   b. start R
   c. select desktop as current directory under Misc menu
   d. type: `source('maxent1.R')`
   e. type: `getvalue(mono1,3.1)` to see what the weight 3.1 predicts
   f. type: `getvalue(mono2,c(3.1,2))` to see what the weights 3.1 and 2 predict
   g. type: `learn(mono1)` or `learn(mono2)` to find best weights.

E. Learning constraints

(28) Constraint format

\[
\begin{bmatrix}
\alpha F \\
\beta G \\
\gamma H \\
\delta I \\
\epsilon J \\
\zeta K
\end{bmatrix}
\]

(29) One of the matrices of a constraint can contain the complementation operator; thus `[^\alpha F, \beta G, ...]` means any segment not in the natural class `[\alpha F, \beta G, ...]`.

(30) If the number of features and feature values is finite, and the number of matrices in any one constraint is finite, then the number of possible constraints is finite.

(31) Accuracy
Select constraints that are violated as little as possible by the actual data \((O[C_i])\) as compared with the possible data \((E[C_i])\).

(32) Generality
Prefer constraints with fewer matrices and where matrices define more general classes.
Phonotactic learning algorithm
Input: a set \( \Sigma \) of segments classified by a set \( \mathcal{F} \) of features, a set \( \mathcal{D} \) of surface forms drawn from \( \Sigma^* \), an ascending set \( \mathcal{A} \) of accuracy levels, and a maximum constraint size \( \mathcal{N} \).

1: begin with an empty grammar \( \mathcal{G} \)
2: for each accuracy level \( a \) in \( \mathcal{A} \) do
3: repeat
4: select the most general constraint with accuracy less than \( a \) (if one exists) and add it to \( \mathcal{G} \)
5: train the weights of the constraints in \( \mathcal{G} \)
6: while a constraint is selected in step 4
7: end for

F. Et cetera

Maxent values can be converted to probabilities and/or regressed against well-formedness judgments.

Assumptions:
a. features
b. accuracy and generality
c. projections

References
